**A Project Report on**

**Disease Prediction and Medicine Recommendation System**

submitted in partial fulfillment for the award of

**Bachelor of Technology**

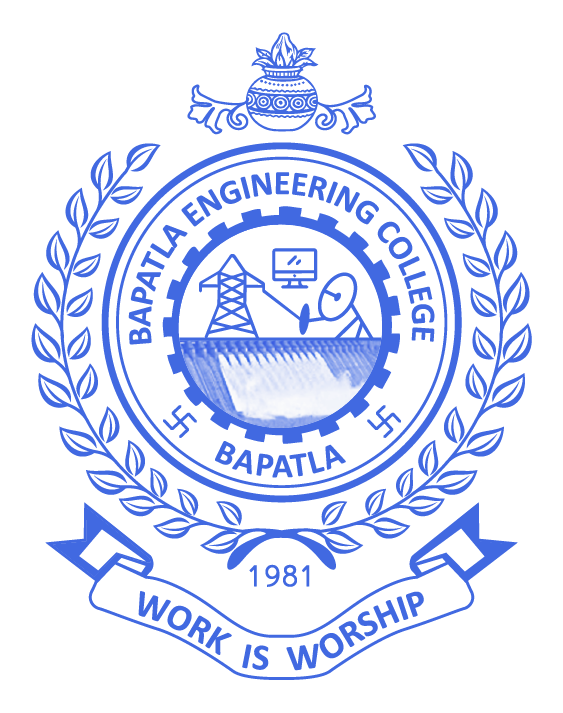
in

**Computer Science and Engineering**

by

**M. Poojitha (Y20ACS499) P. Yaswanth (Y20ACS523)**

**M. Chandrika (Y20ACS492) K. V. S. Vyshnavi(Y20ACS477)**



Under the guidance of

**Dr. K. Mani Deep, Ph.D.,**

Associate Professor

Department of Computer Science and Engineering

**Bapatla Engineering College**

(Autonomous)

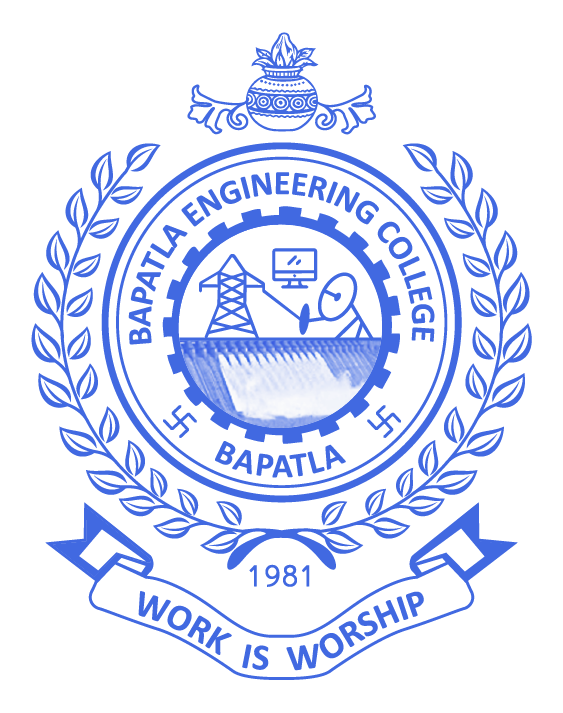
(Affiliated to Acharya Nagarjuna University)

**BAPATLA – 522 102, Andhra Pradesh, INDIA**

**2023-2024**

**Department of**

**Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the project report entitled **Disease Prediction and Medicine Recommendation System** that is being submitted by M. Poojitha (Y20ACS499), P. Yaswanth (Y20ACS523), M. Chandrika (Y20ACS492), K. V. S. Vyshnavi (Y20ACS477) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

**Signature of the Guide Signature of the HOD**

**Dr. K. Mani Deep, Ph.D., Dr. M. Rajesh Babu**

**Associate Professor Associate Professor**

**DECLARATION**

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

M. Poojitha (Y20ACS499)

P. Yaswanth (Y20ACS523)

M. Chandrika (Y20ACS492)

K. V. S. Vyshnavi (Y20ACS477)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **Dr. K. Mani Deep,** Associate Professor, Department of CSE, for his valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. M. Rajesh Babu**, Assoc. Prof. & Head of the Dept., for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal **Dr. Nazeer Shaik** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. N. Sudhakar,** Prof., Dept. of CSE, for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

M. Poojitha (Y20ACS499)

P. Yaswanth (Y20ACS523)

M. Chandrika (Y20ACS492)

K. V. S. Vyshnavi (Y20ACS477)

Table of Contents

[List of Figures viii](#_Toc165030663)

[List of Tables ix](#_Toc165030664)

[**Abstract** x](#_Toc165030665)

[1 Introduction 1](#_Toc165030666)

[1.1 Objective 1](#_Toc165030667)

[1.2 Overview of the project 1](#_Toc165030668)

[1.3 Machine learning 2](#_Toc165030669)

[1.4 Working of Machine Learning 2](#_Toc165030670)

[1.5 Advantages of Machine Learning 3](#_Toc165030671)

[2 Literature Survey 4](#_Toc165030672)

[2.1 Existing System 6](#_Toc165030673)

[2.2 Limitations of Existing methods 6](#_Toc165030674)

[3 Proposed System 7](#_Toc165030675)

[3.1 Architecture 7](#_Toc165030676)

[3.1.1 Data set 7](#_Toc165030677)

[3.1.2 Attribute selection 7](#_Toc165030678)

[3.1.3 Overfitting 8](#_Toc165030679)

[3.1.4 Accuracy 8](#_Toc165030680)

[3.1.5 Training Time 9](#_Toc165030681)

[3.1.6 Data Preprocessing 9](#_Toc165030682)

[3.1.7 Classification Techniques 11](#_Toc165030683)

[3.2 Features of Proposed System 12](#_Toc165030684)

[3.2.1 Feasibility Study 12](#_Toc165030685)

[3.2.2 Technical Feasibility 13](#_Toc165030686)

[3.2.3 Operational Feasibility 13](#_Toc165030687)

[3.2.4 Economic Feasibility 13](#_Toc165030688)

[3.3 Advantages of Proposed System 14](#_Toc165030689)

[4 Software and Hardware Requirements 15](#_Toc165030690)

[4.1 Software Requirements 15](#_Toc165030691)

[4.2 Hardware Requirements 15](#_Toc165030692)

[5 Design 16](#_Toc165030693)

[5.1 Goals 16](#_Toc165030694)

[5.2 Activity Diagram 17](#_Toc165030695)

[5.3 Use Case Diagram 18](#_Toc165030696)

[5.4 Class Diagram 19](#_Toc165030697)

[5.5 Sequence Diagram 20](#_Toc165030698)

[6 Methodology and Algorithms 21](#_Toc165030699)

[6.1 Data collection 21](#_Toc165030700)

[6.1.1 Dataset 21](#_Toc165030701)

[6.2 Naive Bayes 22](#_Toc165030702)

[6.3 Random Forest 23](#_Toc165030703)

[6.4 Decision Tree 23](#_Toc165030704)

[7 Testing 25](#_Toc165030705)

[7.1 Testing levels 25](#_Toc165030706)

[7.1.1 Unit testing 25](#_Toc165030707)

[7.1.2 Integration Testing 26](#_Toc165030708)

[7.1.3 System Testing 26](#_Toc165030709)

[7.1.4 Acceptance Testing 26](#_Toc165030710)

[7.1.5 Performance testing 27](#_Toc165030711)

[7.2 System test cases 27](#_Toc165030712)

[7.3 Testing the level of prediction of disease: 28](#_Toc165030713)

[8 System Implementation 31](#_Toc165030714)

[8.1 Dataset 31](#_Toc165030715)

[8.2 Preprocessing 35](#_Toc165030716)

[8.3 Functionality of the System 35](#_Toc165030717)

[8.3.1 Input Processing Block 35](#_Toc165030718)

[8.3.2 Disease Prediction Block 36](#_Toc165030719)

[8.3.3 Medicine Recommendation Block 36](#_Toc165030720)

[8.4 GitHub Repository 36](#_Toc165030721)

[9 Results 37](#_Toc165030722)

[9.1 Code 37](#_Toc165030723)

[9.2 GUI 37](#_Toc165030724)

[10 Conclusion 42](#_Toc165030725)

[11 Future work 43](#_Toc165030726)

[12 References 44](#_Toc165030727)

List of Figures

Figure 3.1 Proposed System architecture 9

Figure 5.1 Activity Diagram 17

Figure 5.2 Use Case Diagram 18

Figure 5.3 Class Diagram 19

Figure 5.4 sequence Diagram 20

Figure 7.1 Strong Prediction 29

Figure 7.2 Average Prediction 29

Figure 7.3 Weak Prediction 30

Figure 8.1 structured dataset 31

Figure 8.2 features in the dataset 32

Figure 8.3 labels in the dataset 32

Figure 8.4 description dataset 33

Figure 8.5 medication dataset 33

Figure 8.6 diet dataset 34

Figure 8.7 precautions dataset 34

Figure 9.1 code in jupyter notebook 37

Figure 9.2 Output home page 37

Figure 9.3 Tkinter system for predicting disease 38

Figure 9.4 Symptoms in list box 39

Figure 9.5 Random Forest Prediction 39

Figure 9.6 Naïve Bayes prediction 40

Figure 9.7 Decision tree prediction 40

Figure 9.8 Prescribing medications associated with diet and precaution 41

List of Tables

Table 7.1 Test case showing correct symptoms details 27

Table 7.2 Test case showing incorrect symptoms 28

**Abstract**

Disease Prediction and Medicine Recommendation system is based on predictive modeling that forecasts the user’s disease on the basis of the symptoms that user provides as an input to the system and also makes pharmaceutical recommendations. It uses data mining techniques for developing decision support system for diseases prediction through a set of medical data sets. In this project, we propose a new knowledge-based system for disease prediction using Naive Bayes algorithm, Decision Tree algorithm, Random Forest algorithm by having these three algorithms there are some medical benefits like early disease detection, patient care and remedies.

To display symptoms, disease, and medications for user interface we use python GUI interface. Out of all the graphical user interface methods tkinter is most commonly used interface. Tkinter output gives the fastest and easiest way to create GUI applications. The proposed system uses few Machine Learning Algorithms Such as Naive Bayes, Random Forest, Decision Tree. We use Tkinter module to represent graphical user interface and import the NumPy and pandas modules to access and perform operations on data sets.

# 1 Introduction

With the increase in the number of patients and diseases per annum medical system is overloaded and with time became overpriced in many countries. Most of the disease involves a consultation with doctors to urge treatment. With sufficient data prediction of disease by an algorithm are often very easy and cheap. Prediction of disease by watching the symptoms is an integral part of treatment. In our project, we've tried to accurately predict a disease by watching the symptoms of the patient and suggests some medications. Such a system can have a large potential in medical treatment of the longer term.

## Objective

The main objective of this project is to find out the disease by using machine learning technique that is computationally efficient as well as accurate for the prediction of diseases and medicine recommendation. Enhancing the prediction level to the diseases at earlier stage this can save the time of doctor in decision, increase accuracy, and also gives result fast.

## Overview of the project

Here we proposed a disease prediction and medicine recommendation system project data mining techniques are widely used in developing decision support system for diseases prediction through a set of medical data sets. We propose a new knowledge-based system for disease prediction using Naive Bayes algorithm, decision tree algorithm, random forest algorithm by having these algorithms as well as recommending suitable medicines and remedies. There are some medical benefits early disease detection, patient care.

## Machine learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that for themselves. For example, medical diagnosis, image processing, prediction, classification, etc. The intelligent systems built on machine learning algorithms have the capability to learn from experience or historical data.

## Working of Machine Learning

Machine learning algorithm has two tracks: Training, Testing. Prediction of a disease by using patient's symptoms and history machine learning technology is striving from past decades. Machine Learning technology gives an immeasurable platform in the medical field so that healthcare issues can be resolved efficiently. We are applying machine learning to maintained complete hospital data.

Machine learning technology which allows building models to get quickly analyze data and deliver results faster, with the use of machine learning technology doctors can make a good decision for patient diagnoses and treatment options, which leads to improvement of patient healthcare services. Healthcare is the most prime example of how machine learning is used in the medical field.

## Advantages of Machine Learning

1. Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.
2. Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

# Literature Survey

In this section we provide relevant background on previous work on Disease Prediction and Medicine recommendation system. Recently, several methods have been proposed for predicting diseases. Many of these methods are based on Machine Learning models.

In the paper [1] proposed by Nayak et al. (2023) developed an intelligent disease prediction and drug recommendation prototype employing multiple machine learning algorithms. The study focuses on leveraging various approaches within machine learning to enhance disease prediction accuracy and recommend appropriate drugs or treatments. The authors emphasize the importance of integrating diverse machine learning techniques to improve the overall performance of the prediction and recommendation system.

In the paper [2] proposed by Gupta et al. (2021) a computer-based disease prediction and medicine recommendation system utilizing machine learning techniques. The study, published in the International Research Journal of Engineering and Technology, emphasizes the application of machine learning approaches to enhance disease prediction accuracy and recommend appropriate medicines or treatments. The authors highlight the significance of leveraging computational methods for improving healthcare decision-making processes.

In the paper [3] proposed by Goyal et al. (2020) presented a medicine recommendation system in the International Research Journal of Engineering and Technology. The study focuses on developing a system to recommend medicines using computational methods. The authors highlight the importance of such systems in assisting healthcare professionals in making informed decisions regarding patient treatment options.

In the paper [4] proposed by Swarupa et al. (2021) presented a smart disease prediction system based on the Random Forest algorithm. The study was presented at the IEEE International Conference on Intelligent Systems, Smart and Green Technologies in 2021, discusses the application of Random Forest in disease prediction. The authors describe the methodology and implementation of the system, highlighting its potential in accurately predicting diseases. Their findings contribute to the advancement of intelligent systems in healthcare.

In the paper [5] proposed by John & Ilyas (2016), a medication recommendation system based on clinical documents was presented at the IEEE International Conference on Information Science. The study focuses on leveraging clinical documents to recommend medications, highlighting the potential of such systems in improving healthcare decision-making processes.

In the paper [6] proposed by Knahl & Aydin (2019) conducted a literature review titled 'A Literature Review on Medicine Recommender,' published in the International Journal of Advanced Computer Science and Applications in 2019. The review provides a comprehensive overview of existing research on medicine recommendation systems. Through a thorough analysis of relevant literature, the authors discuss various methodologies, algorithms, and approaches employed in medicine recommendation. Their review serves as a valuable resource for understanding the current state of the art in medicine recommender systems.

## Existing System

Since the arrival of advanced computing, doctors still require the technology in various possible ways like surgical representation process and x-ray photography, but the technology perceptually stayed behind. The method still requires the doctor’s knowledge and experience due to alternative factors starting from medical records to weather conditions, atmosphere, blood pressure and numerous alternative factors. The huge numbers of variables are granted as entire variables that are required to understand the complete working process itself, nevertheless, no model has analyzed successfully. To tackle this drawback, medical decision support systems must be used. We are applying machine learning to maintained complete hospital data Machine learning technology which allows building models to get quickly analyze data and deliver results faster, with the use of machine learning algorithms which is Decision tree Classifier to predict the disease based on symptoms.

## Limitations of Existing methods

1. These systems still primarily depend on doctors' skills and expertise to evaluate results and make decisions, despite the use of machine learning algorithms.
2. It may be inaccurate due to the insufficient use of relevant data by existing algorithms, including weather, patient lifestyle, and genetic predispositions.
3. Here there is a chance of missing data and there will be small data sets.
4. It also takes more time to collect users data.
5. And also the output process is slow, as compared to other types of technologies.

# Proposed System

The proposed system uses a few Machine Learning Algorithms Such as Naive Bayes, Random Forest, Decision Tree. We use Tkinter module to represent graphical user interface and import the NumPy and pandas’ modules to access and perform operations on data sets. We take several data sets and train the system in order to predict the disease by taking symptoms as input and gives personal medicines as well as remedies. In this system we try to provide the people with an application which can anticipate the disease of a patient with high accuracy. User gives the required symptoms as input. The model predicts the output as disease, medications and remedies to the Output Screen.

## Architecture

The architecture of the system is designed to facilitate efficient handling and processing of healthcare data for disease prediction and medication recommendation. It encompasses several key components and processes:

### Data set

Dataset is collected from Kaggle, it contains few files such as disease dataset, medication dataset, description dataset, diet dataset, and precaution dataset.

### Attribute selection

Attribute Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Attribute selection and Data cleaning should be the first and most important step of your model designing.

### Overfitting

Reducing redundant data in a dataset is pivotal for mitigating overfitting in machine learning models. By eliminating irrelevant or redundant features, the model focuses on the most informative aspects of the data, decreasing the likelihood of memorizing noise or irrelevant patterns during training. With less redundant data, the model is better equipped to generalize to unseen examples, as it learns more robust and representative relationships within the data. This reduction in redundancy also minimizes the risk of the model making decisions based on noise, thus enhancing its accuracy and reliability in real-world applications.

### Accuracy

Improving accuracy in machine learning models is essential for ensuring reliable predictions and decision-making. One effective strategy to enhance accuracy is by reducing the presence of misleading or erroneous data within the dataset. When the dataset contains fewer misleading data, the model can focus on learning genuine patterns and relationships, leading to more accurate predictions. By filtering out irrelevant or noisy information, the model's training process becomes more focused, allowing it to discern meaningful signals from the data more effectively.

As a result, the model is better equipped to generalize to unseen instances and make accurate predictions in real-world scenarios. Moreover, reducing misleading data minimizes the risk of the model being influenced by outliers or anomalies, further improving its overall accuracy and robustness. In summary, by mitigating the impact of misleading data, machine learning models can achieve higher levels of accuracy, providing more dependable insights and predictions for various applications.

### Training Time

Reducing the number of data points in a dataset decreases algorithm complexity and training time. With fewer data points, computations during training are streamlined, optimization processes are simplified, and the model becomes more lightweight and easier to deploy. This efficiency boost enables faster training without sacrificing performance, making machine learning algorithms more scalable and practical for real-world applications.

### Data Preprocessing

Data mining is the process of analyzing, extracting data and furnishes the data as knowledge which forms the relationship between the available data. Some of the data mining techniques include association, clustering, classification, and prediction. Various data mining tools are compared to analyze the performance of healthcare data for disease prediction and recommend medications.

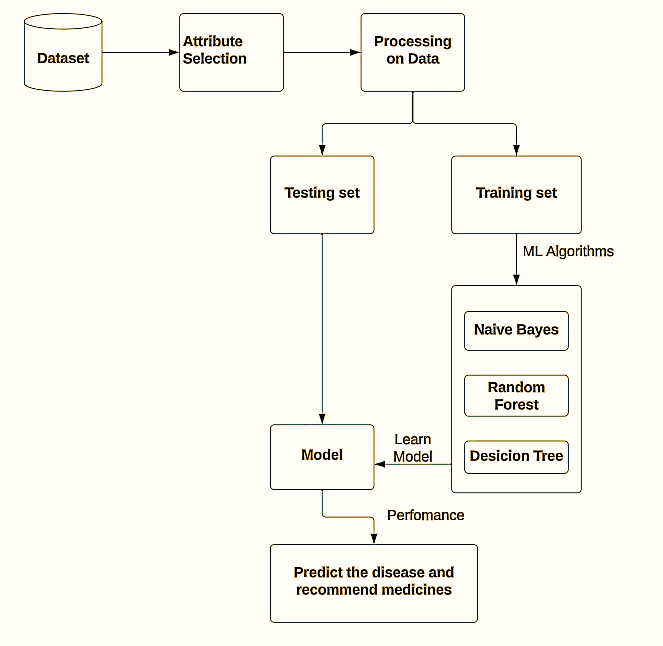
****

Figure 3.1 Proposed System architecture

The diagram shows a simplified overview of a machine learning workflow for medical diagnosis. Here’s a breakdown of the steps:

1. **Data Selection:** Collect a relevant dataset containing information on diagnosed conditions, such as diabetes patients' data for a diabetes diagnosis program.
2. **Data Processing:** Clean, format, and handle missing data to prepare the dataset for machine learning. Ensure patient data is anonymized and properly formatted.
3. **Split Data:** Divide the dataset into training and testing sets to train and evaluate the machine learning model's performance, respectively.
4. **Machine Learning Algorithms:** Choose suitable algorithms like Naive Bayes, Random Forest, or Decision Tree based on the diagnosis problem's nature.
5. **Training the Model:** Feed the training set into the chosen machine learning algorithm to allow it to learn and identify patterns in patient data associated with specific diseases.
6. **Model Performance:** Evaluate the trained model's accuracy using the testing set to assess its diagnostic capabilities.
7. **Prediction and Medication Recommendation:** Utilize the accurate model to predict diseases based on new patient data and recommend suitable medications.

This workflow illustrates the sequential steps involved in developing and deploying a machine learning model for medical diagnosis, emphasizing the importance of data preparation, algorithm selection, and model evaluation in achieving accurate diagnostic outcomes.

### Classification Techniques

Classification techniques are fundamental in machine learning for categorizing data into predefined classes or categories. They are crucial for various applications such as spam detection, sentiment analysis, and medical diagnosis. In this section, we explore three prominent classification techniques:

1. **Naive Bayes:** Naive Bayes is a simple, yet powerful probabilistic classifier based on Bayes' theorem with the "naive" assumption of independence among features. It is particularly effective for text classification tasks such as email spam detection and sentiment analysis. Naive Bayes classifiers are computationally efficient, making them suitable for large datasets. They can handle multiple classes and are robust to irrelevant features.
2. **Decision Tree:** Decision trees are versatile and interpretable models that recursively partition the feature space into segments, forming a tree-like structure. Each node represents a decision based on a feature, leading to leaf nodes that correspond to class labels. Decision trees are advantageous for their ability to handle both numerical and categorical data. They are easy to understand and visualize, making them useful for explaining the decision-making process to stakeholders. However, decision trees are prone to overfitting, especially with complex datasets.
3. **Random Forest:** Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions through voting or averaging. It alleviates the overfitting issue of individual decision trees by aggregating the results of multiple models. Random Forests are highly scalable and can efficiently handle large datasets due to their parallelizable nature. They are robust to noise and outliers, making them suitable for real-world applications. Additionally, Random Forests provide feature importance rankings, aiding in feature selection and understanding the underlying data patterns.

These classification techniques offer a diverse set of tools for solving classification problems across various domains. Their effectiveness and applicability depend on factors such as dataset size, feature space complexity, and interpretability requirements. By understanding the characteristics and strengths of each technique, practitioners can make informed decisions in designing and deploying classification models.

## Features of Proposed System

The system allows users to input their symptoms, leveraging machine learning algorithms to predict potential diseases accurately. Based on the predicted diseases, the system generates personalized medication recommendations tailored to each patient's unique needs. The system utilizes structured data sources exclusively, eliminating the complexity associated with handling unstructured data. By leveraging structured data from diverse healthcare datasets, the system constructs robust disease risk models and enhances prediction accuracy. The system features a user-friendly interface that allows seamless interaction with healthcare providers and patients.

### Feasibility Study

Preliminary investigation examines project feasibility the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

### Technical Feasibility

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the situation. Technical feasibility center on the existing computer system and to what extent it can support the proposed addition. Python and its libraries are technology software which are used to develop Data Analytics. So, there is no need for additional purchase of any software and these are open-source software which are freely available in Internet.

### Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information systems that will meet the user's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the application implementation. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use the system. Also, the system is very friendly and easy to use.

### Economic Feasibility

To decide whether a project is economically feasible, we have to take into consideration various factors as:

1. Cost benefit analysis
2. Long-term returns
3. Maintenance costs

The proposed system is computer based. It requires average computing capabilities which is very basic requirement and can be afforded by an organization; it doesn't incur additional economic overheads, which renders the system economically feasible.

## Advantages of Proposed System

1. It provides higher accuracy.
2. We leverage not only the structured data but also the text data of patients.
3. based on the proposed Naive Bayes, random forest and decision tree algorithms.
4. Personalized medication recommendations can be generated by the system based on analysis of individual patient data, taking into consideration the symptoms of each patient.
5. The system can possibly improve patient outcomes and reduce costs associated with extended illnesses using real-time patient data and predictive modelling to enable early disease detection.

# Software and Hardware Requirements

The following software and hardware requirements are essential for the successful implementation of this project:

## Software Requirements

1. Coding language: Python
2. Work Space: Jupyter Notebook
3. GUI : Tkinter
4. Operating System: Windows 8 required
5. Libraries: Pandas, NumPy, Matplotlib, Sklearn, Tkinter

## Hardware Requirements

1. Processor: 7th Gen Intel core i5 or Ryzen 5 or higher
2. RAM: 8 GB
3. Hard Disk: 256 GB

# Design

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

## Goals

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of Object-Oriented tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

## Activity Diagram

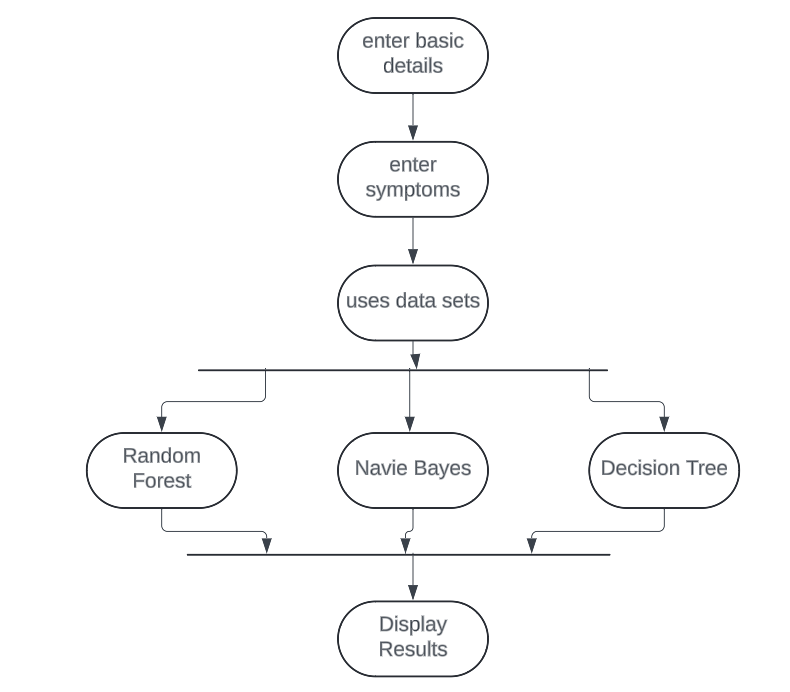
Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

Figure 5.1 Activity Diagram

In the Figure 5.1 Activity Diagram, the user will enter his basic details and then his symptoms and the Algorithms like Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB) will train the model and its respective disease as well as medications will be given on the user interface.

## Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

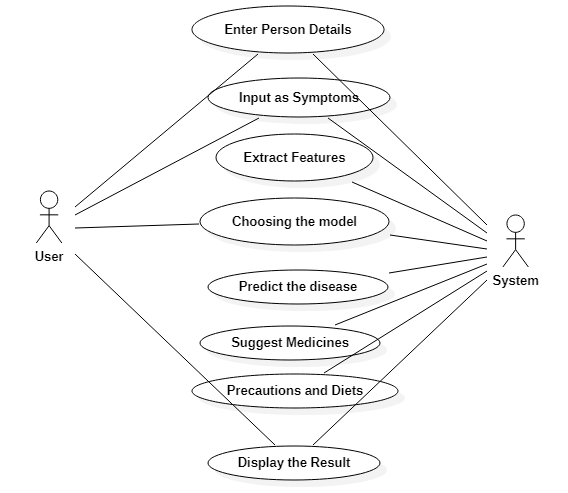


Figure 5.2 Use Case Diagram

In the Figure 5.2 Use Case Diagram, shows that system the takes a user's symptoms as input, extracts features from them, and then uses a model to predict the disease with medications. The system then displays the results to the user.

## Class Diagram

A class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

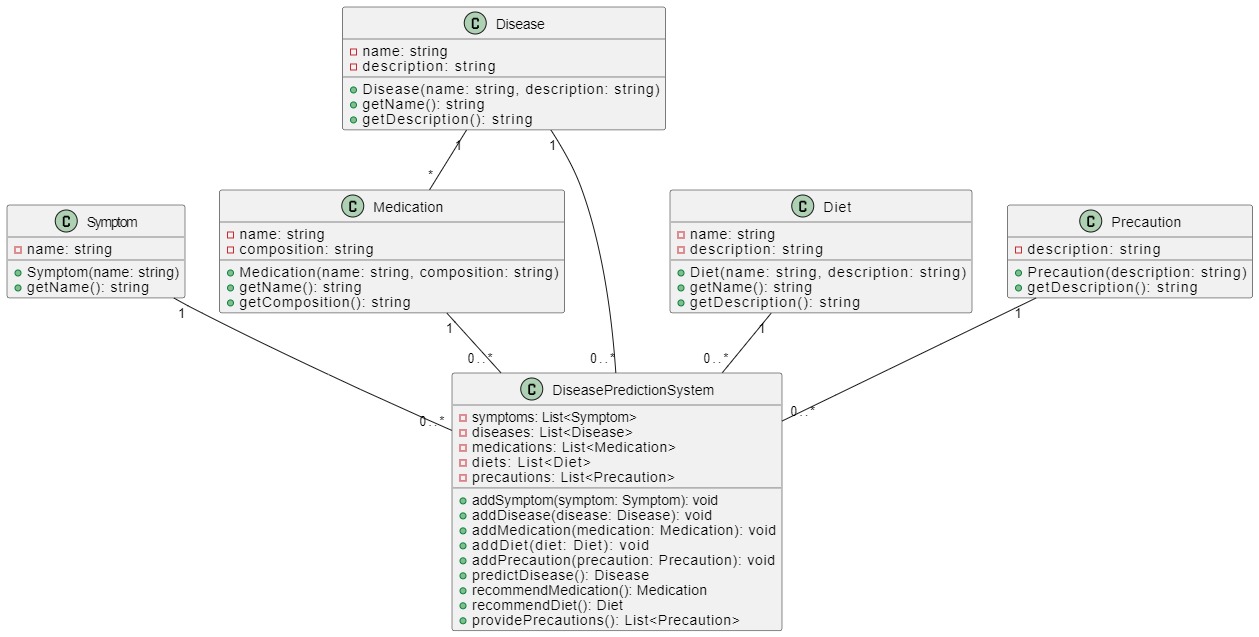


Figure 5.3 Class Diagram

In the Figure 5.3 Class Diagram, shows a disease prediction system with medications. The system contains information on various diseases, symptoms, medications, diets, and precautions. It also has methods to add new information and predict diseases based on a list of symptoms.

## Sequence Diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

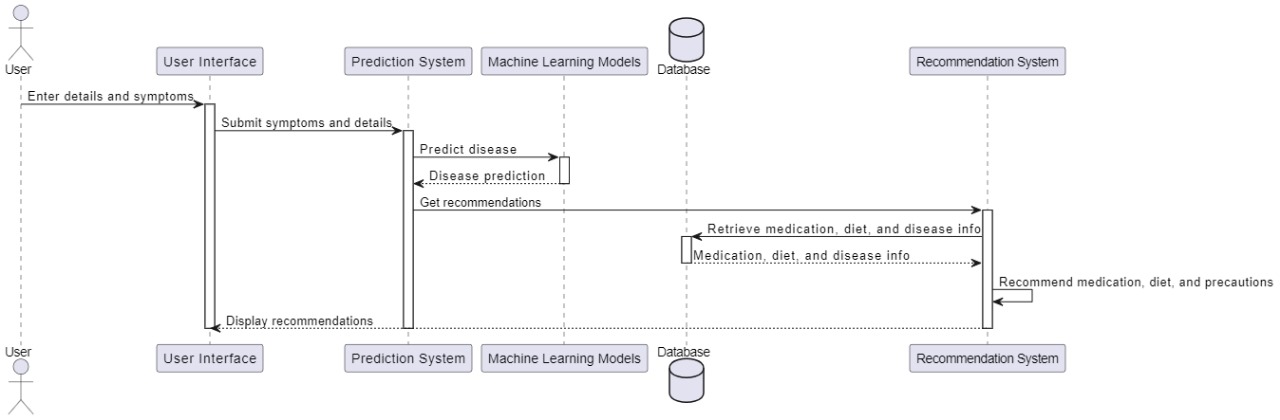


Figure 5.4 sequence Diagram

In the Figure 5.4 sequence Diagram, shows a recommendation system that leverages a user interface, prediction system, machine learning models, and a database to provide users with medication, diet, and precaution recommendations.

# Methodology and Algorithms

The following are the methods and algorithms used to implement our project:

## Data collection

Due to big data progress in biomedical and healthcare communities, accurate study of medical data benefits early disease recognition, patient care and community services. When the quality of medical data is incomplete the exactness of study is reduced. Moreover, different regions exhibit unique appearances of certain regional diseases, which may result in weakening the prediction of disease outbreaks. In this project, it bid a Machine learning Decision tree map, Naive Bayes, Random Forest algorithm by using structured data from hospital.

It also uses Machine learning algorithm for partitioning the data. To the highest of gen, none of the current work attentive on together data types in the zone of remedial big data analytics. Compared to several typical calculating algorithms, the scheming accuracy of our proposed algorithm reaches 95% with an regular speed which is quicker than that of the unimodal disease risk prediction algorithm and produces report.

### Dataset

Datasets are collected from Kaggle, The project leverages four key datasets: symptoms.csv, medication.csv, diet.csv and precautions.csv. The "symptoms.csv" dataset contains anonymized patient data for disease prediction algorithms. "Medication.csv" provides comprehensive information on medications, including dosages and side effects, facilitating medicine recommendation systems. "Diet.csv" offers tailored dietary recommendations for various diseases, while "precautions.csv" outlines specific health precautions and workout routines. Integrating these datasets, the project aims to develop a robust disease prediction and medicine recommendation system complemented by personalized health guidance for enhanced patient care and wellness management.

## Naive Bayes

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P (d1, d2, d3/h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) \* P(d2|H) and so on. This is a very strong assumption that is most unlikely in real data, i.e., that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption does not hold.

1. Representation Used by Naive Bayes Models
2. The representation for naive Bayes is probabilities.
3. A list of probabilities is stored to file for a learned naive Bayes model.

This includes:

1. **Class Probabilities:** The probabilities of each class in the training dataset.
2. **Conditional Probabilities:** The conditional probabilities of each input value given each class value.
3. On applying the naïve bayes on the dataset we achieved accuracy nearly 94%.

## Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. We can understand the working of Random Forest algorithm with the help of following steps.

1. First, start with the selection of random samples from a given dataset.
2. Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
3. In this step, voting will be performed for every predicted result.
4. At last, select the most voted prediction result as the final prediction result. On applying random forest we achieved accuracy 95%.

## Decision Tree

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too. The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data(training data). The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

**Decision Tree Algorithm Pseudocode**

1. Place the best attribute of the dataset at the root of the tree.
2. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
3. Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.
4. On applying decision tree, we achieved accuracy 93%.

# Testing

Software Testing is a process of executing the application with an intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application are working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of the application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

## Testing levels

Testing levels encompass a range of specificity in software testing, tailored to different stages of development and system complexity. These levels include unit testing, which examines individual components in isolation. integration testing, focusing on the interaction between integrated components; system testing, evaluating the system as a whole; and acceptance testing, verifying whether the system meets user requirements. Each level serves a distinct purpose in ensuring the quality and functionality of the software, with testing becoming progressively broader in scope and addressing higher-level functionalities as development progresses.

### Unit testing

Unit testing validates the functionality of isolated code units, typically at the function level, by executing automated test cases. It ensures that individual components produce expected outputs for various inputs, facilitating rapid feedback on code changes and early bug detection. By pinpointing and addressing errors early in the development cycle, unit testing enhances code reliability, maintainability, and overall software quality.

### Integration Testing

Integration testing is any type of software testing that seeks to verify the interfaces between components of a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.

### System Testing

System testing evaluates the entire integrated system to ensure it meets specified requirements and functions as intended. It examines system behavior in real-world scenarios, assessing its performance, reliability, and compliance with user expectations. Through comprehensive testing of the system's functionality and features, system testing aims to validate its readiness for deployment and identify any potential issues or discrepancies.

### Acceptance Testing

Acceptance testing assesses the application's readiness for deployment by confirming its adherence to user requirements and expectations. It involves validating that the software meets stakeholders' criteria and fulfills its intended purpose within the intended environment. Through acceptance testing, stakeholders determine whether the application is fit for deployment and ready to be used in production.

### Performance testing

Performance testing involves assessing the speed and efficiency of a system, application, or device by measuring various parameters like response time, throughput, and scalability. It helps in identifying potential bottlenecks and optimizing performance to ensure smooth operations and user satisfaction.

## System test cases

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance with a specific requirement. I have designed and executed a few test cases to check if the project meets the functional requirements.

In the below table, shows that if the user enters the correct symptoms which can be either two or more symptoms related to the specific disease then the actual disease will be predicted and suggest medications by the system.

Table 7.1 Test case showing correct symptoms details

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition test** | **Input specifications** | **Output**  **specifications** | **Pass/Fail** |
| If the user tries to give correct symptoms of disease the symptoms may be two or more. | Gives correct symptoms the person is suffering. | The person conclude with actual disease and suggest medications. | Pass |

In the below table, shows that if the user enters the incorrect symptoms which can be either two or more symptoms according to his/her wish than the system will classify the wrong disease with medications.

Table 7.2 Test case showing incorrect symptoms

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition test** | **Input specifications** | **Output**  **Specification** | **Pass/Fail** |
| If the user tries to give incorrect symptoms of disease the symptoms may be two or more. | Gives incorrect symptoms the person is suffering. | The person conclude with wrong disease and medications. | Fail |

## Testing the level of prediction of disease:

The predictions made by different machine learning algorithms for disease diagnosis. The algorithms considered are Decision Tree, Random Forest, and Naive Bayes. Each algorithm predicts a disease based on input data, with varying degrees of confidence indicated by the prediction level.

**Strong Prediction:** In the Figure 7.1, shows that all three machine learning models (Decision Tree, Random Forest, Naive Bayes) predict the same disease, the prediction level is strong. In this scenario, the consensus among multiple algorithms reinforces the confidence in the diagnosis of Fungal infection.

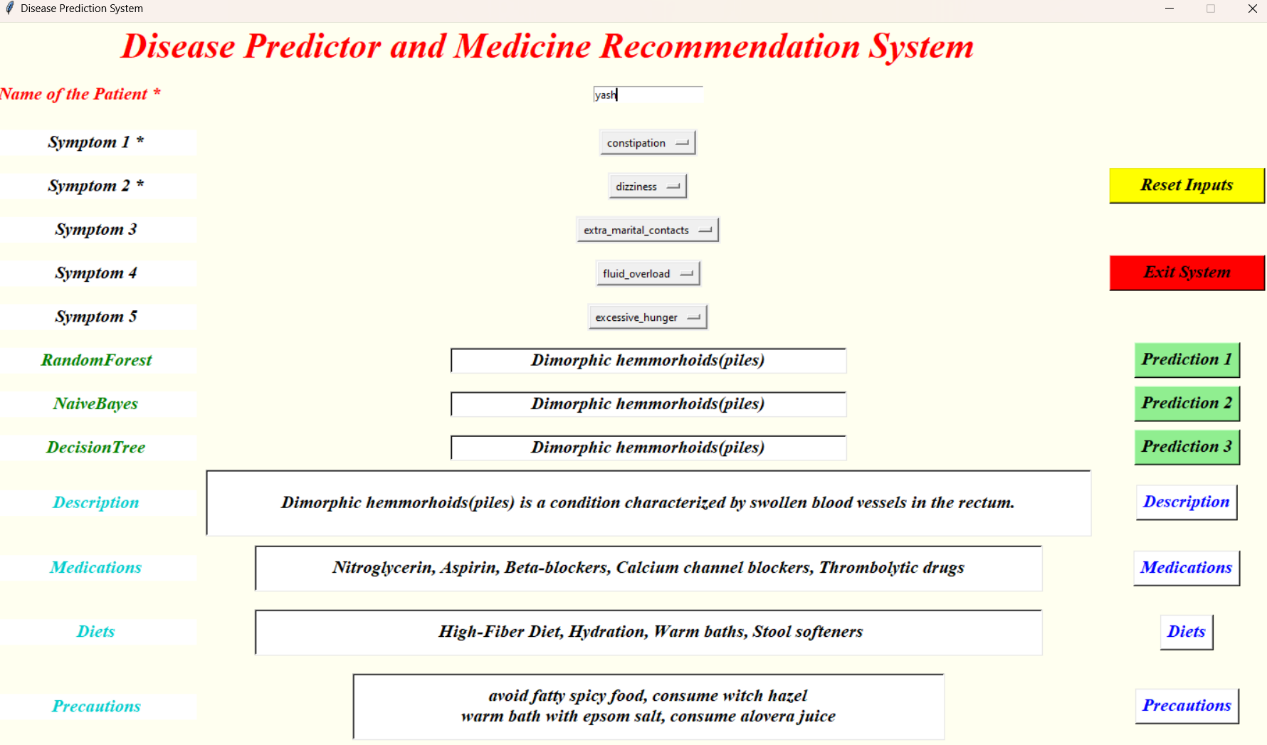


Figure 7.1 Strong Prediction

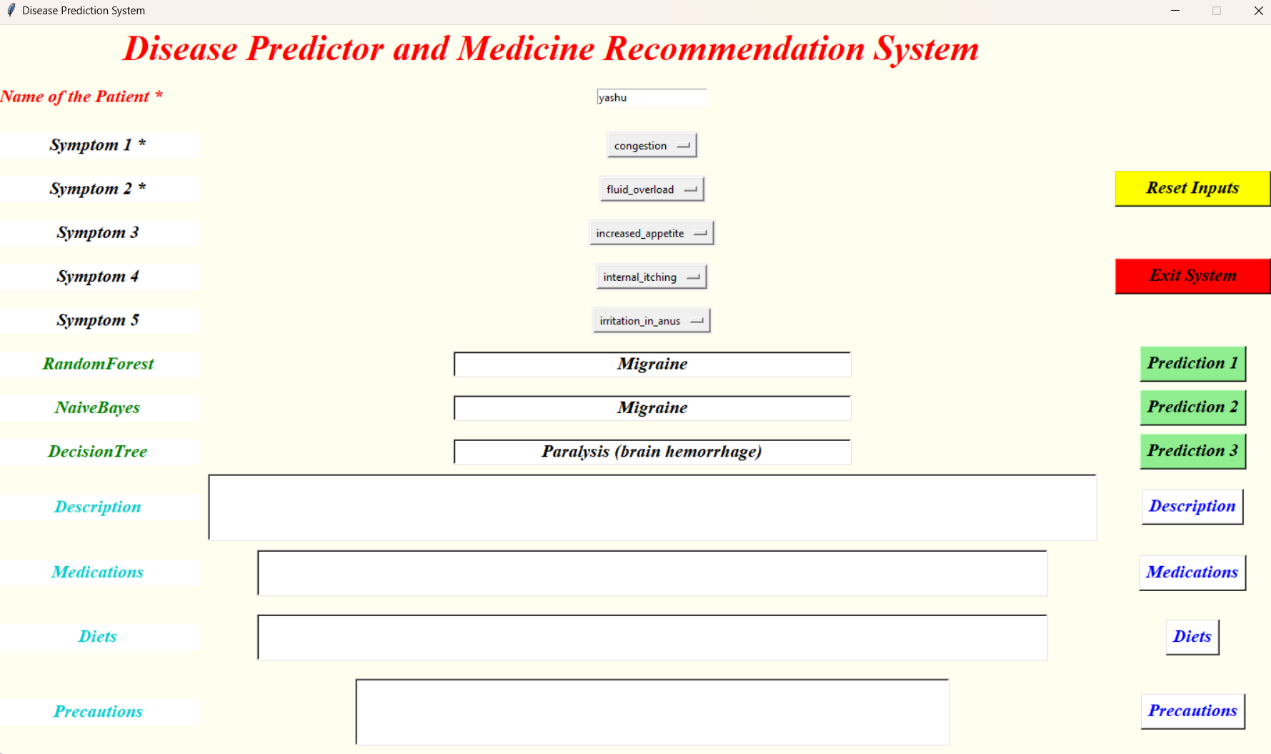


Figure 7.2 Average Prediction

**Average Prediction:** In the Figure 7.2, shows if two out of three models predict the same disease while the third predicts a different disease, the prediction level is average. In this case, there is some level of agreement among the models, indicating a moderate level of confidence in the diagnosis of Migraine.

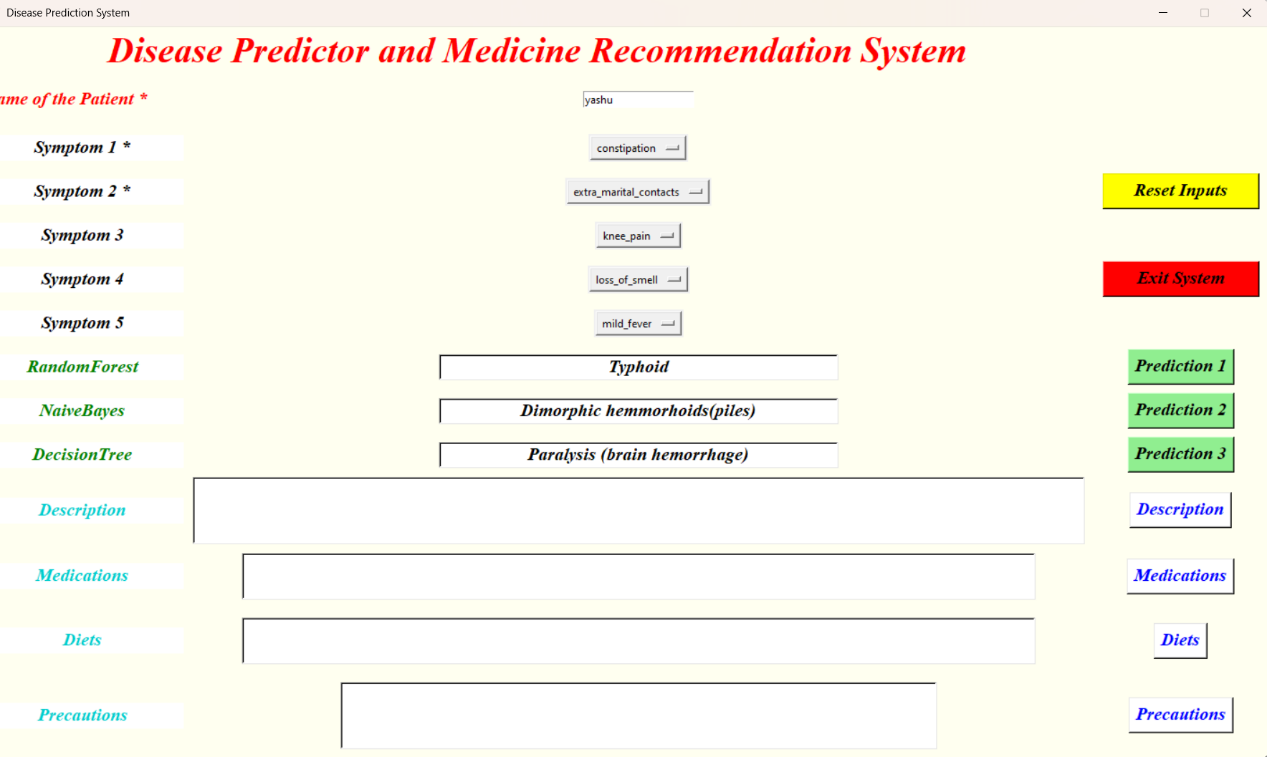


Figure 7.3 Weak Prediction

**Weak Prediction:** In the Figure 7.3, shows that all three models predict different diseases, the prediction level is weak. In the above scenario suggests conflicting results among the models, leading to uncertainty in the diagnosis. Dimorphic hemmorhoids is the final prediction in this case, indicating the need for further evaluation or additional data for a more accurate diagnosis.

# System Implementation

System implementation involves the process of putting the designed system into action, transforming theoretical concepts into practical reality. It encompasses a series of steps aimed at deploying the software and hardware components, configuring them to meet specific requirements

## Dataset

The dataset was taken from Kaggle website where it contains the symptoms of each disease and each disease the symptoms may differ because all the human beings cant get the same symptoms.

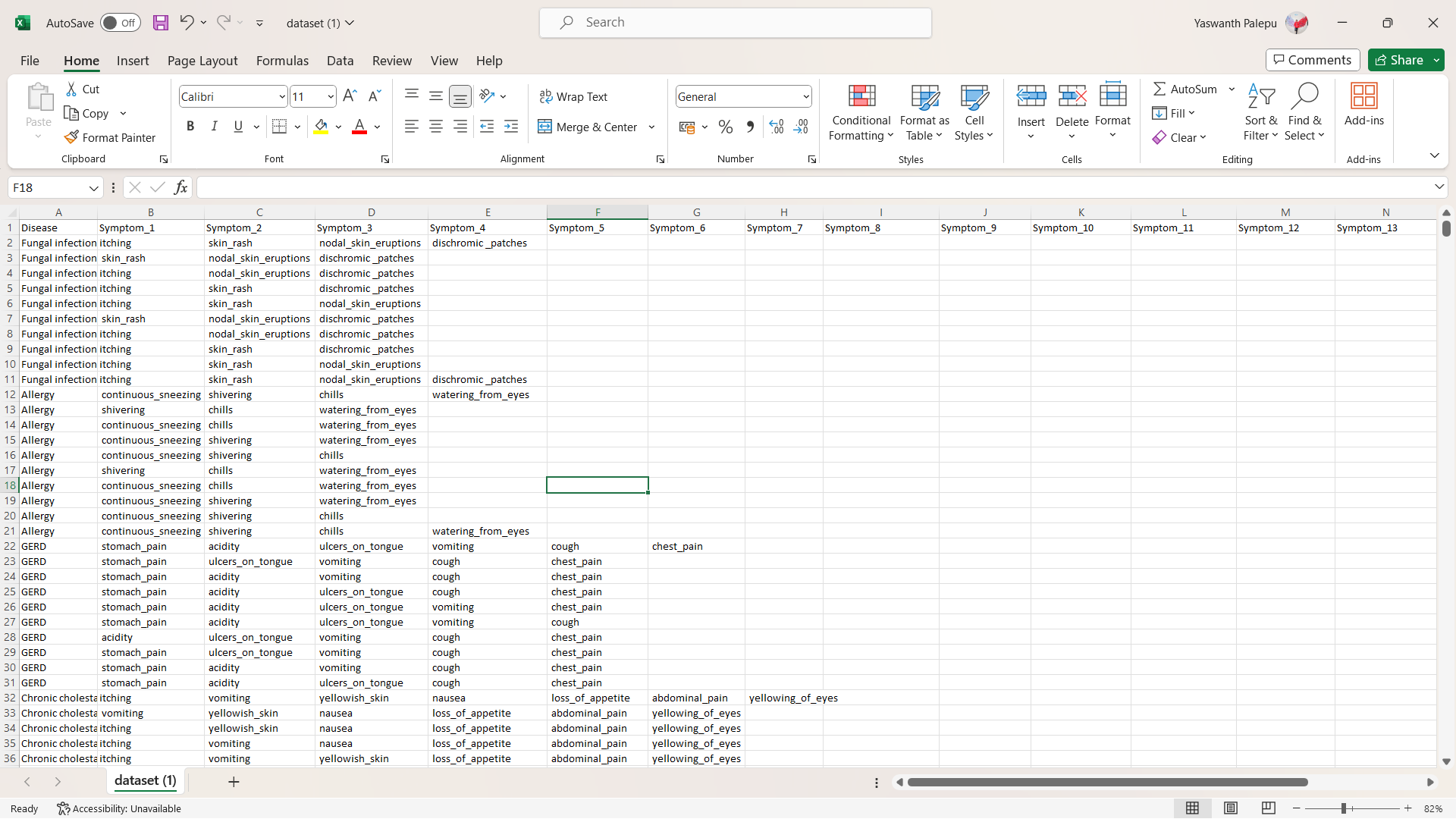


Figure 8.1 structured dataset

In the Figure 8.1 contains raw dataset of 18 columns and approximately 5000 records.

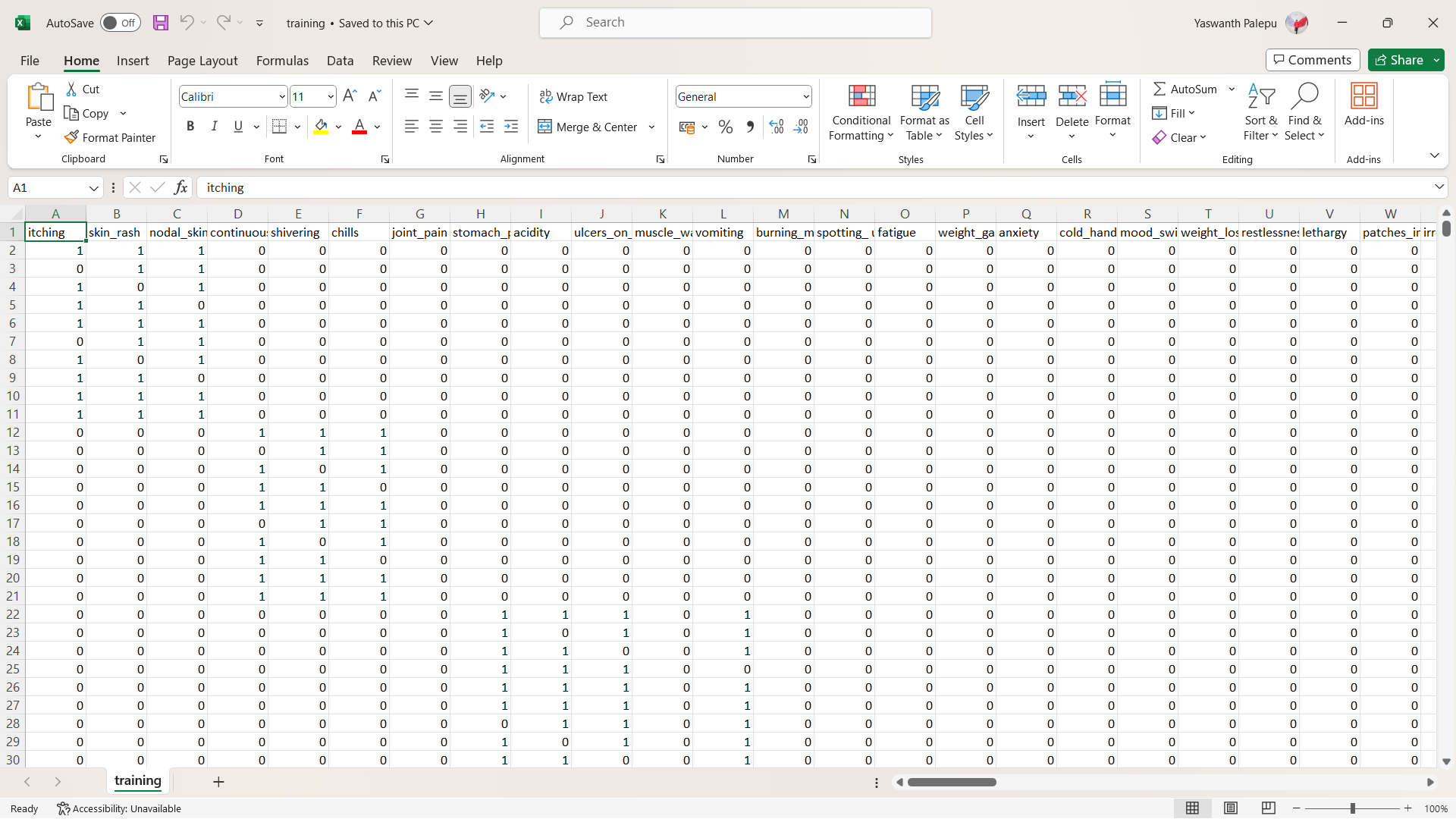


Figure 8.2 features in the dataset

In the Figure 8.2 contains some of the features moreover the dataset contains total of 91 attributes. The dataset has approximately 5000 records.

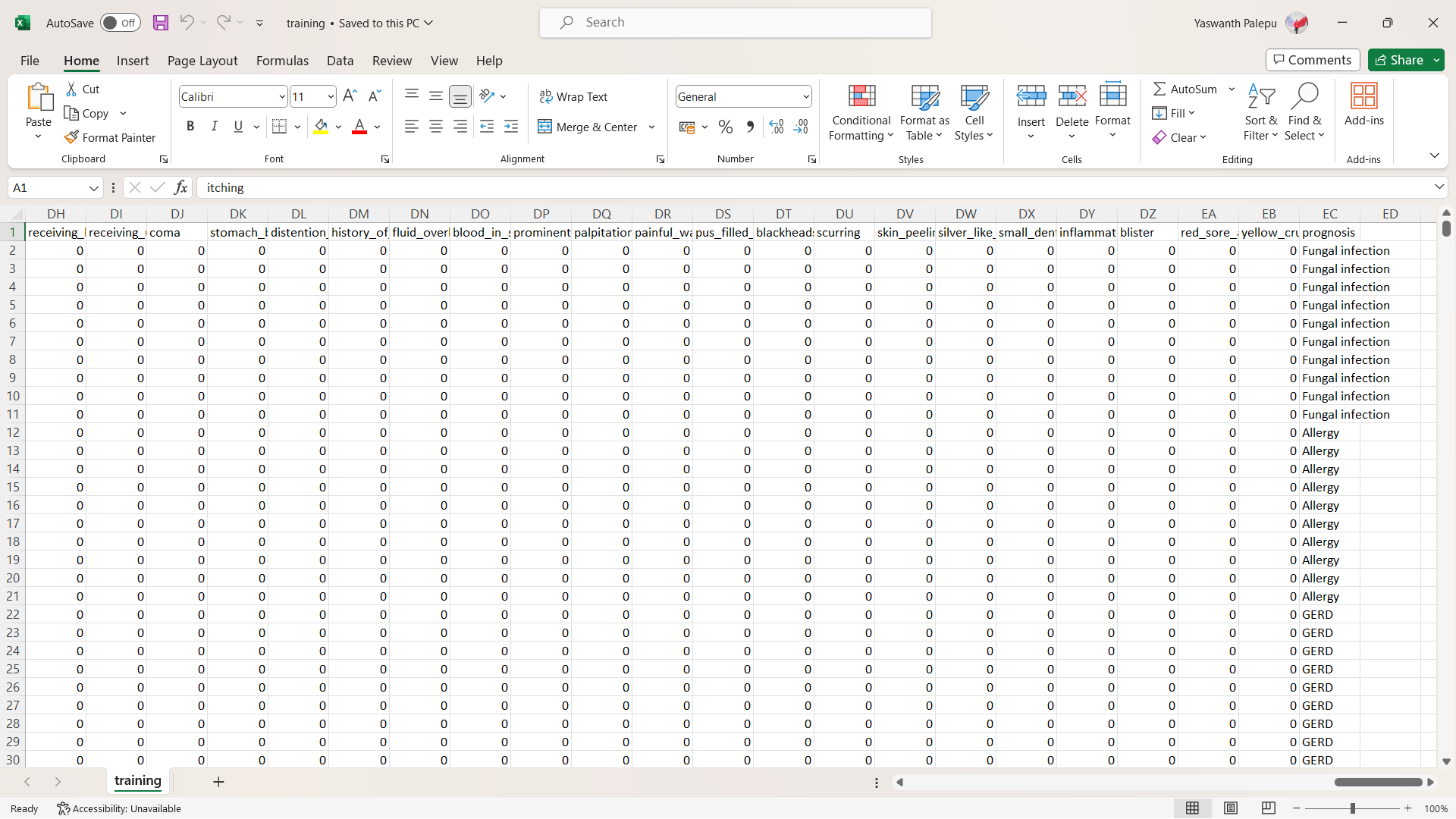


Figure 8.3 labels in the dataset

In the Figure 8.3 contains the target value prognosis contains the disease related to respective symptoms and has a total of 41 different diseases.

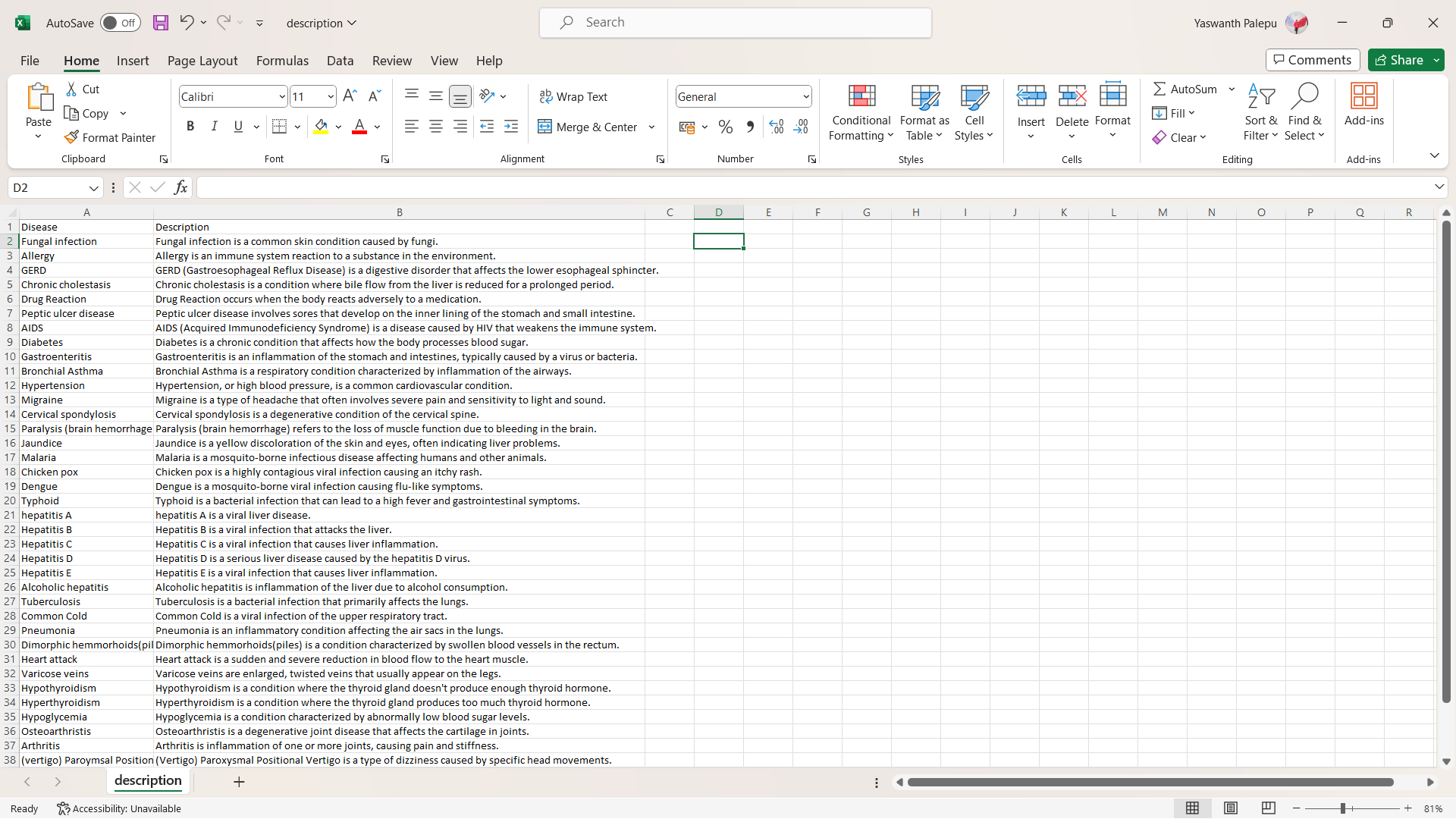
****

Figure 8.4 description dataset

In the Figure 8.4 contains the brief description about the 41 different diseases. In the Figure 8.5 contains the suggestable medication for the disease related to respective symptoms.

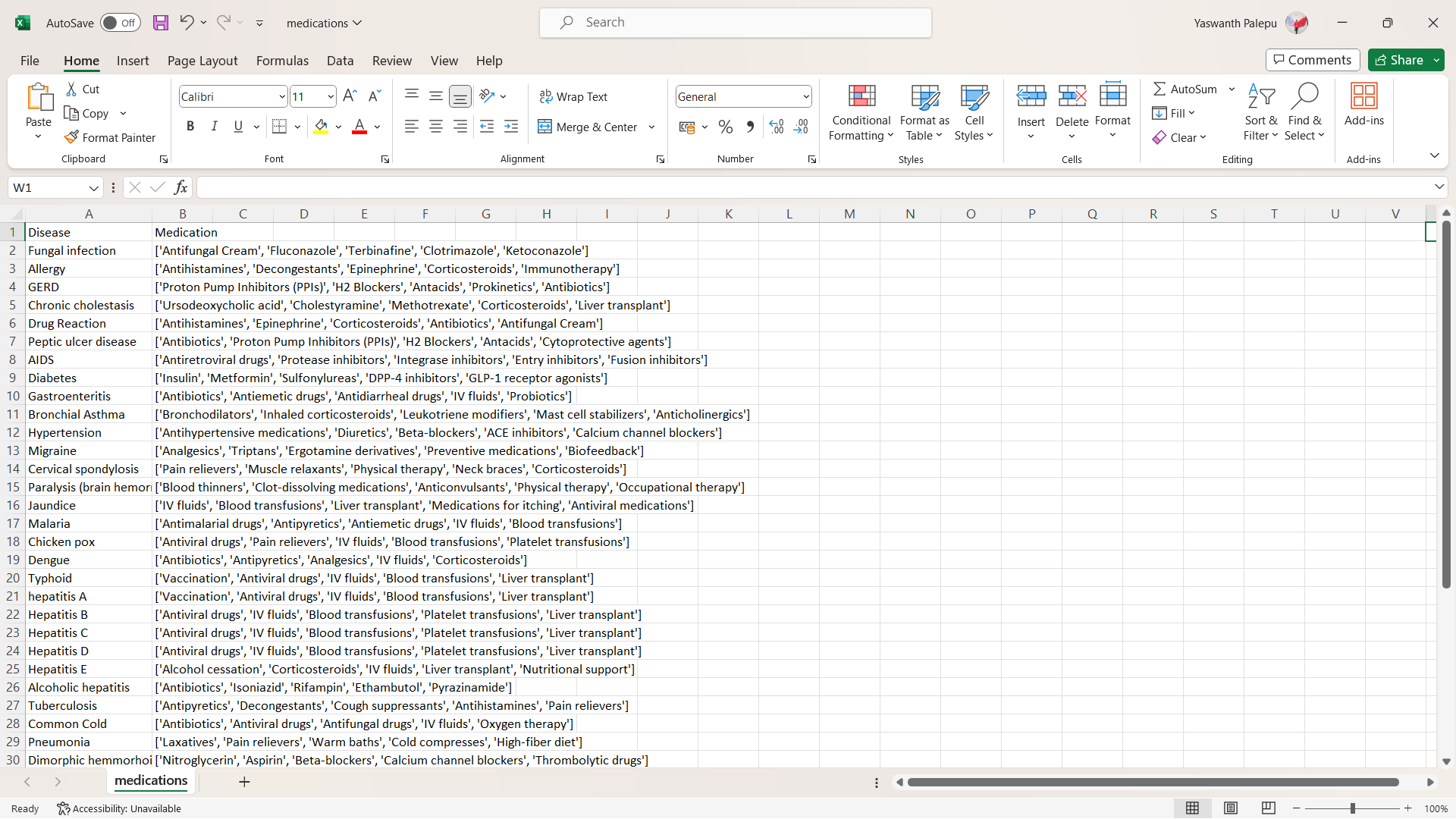


Figure 8.5 medication dataset

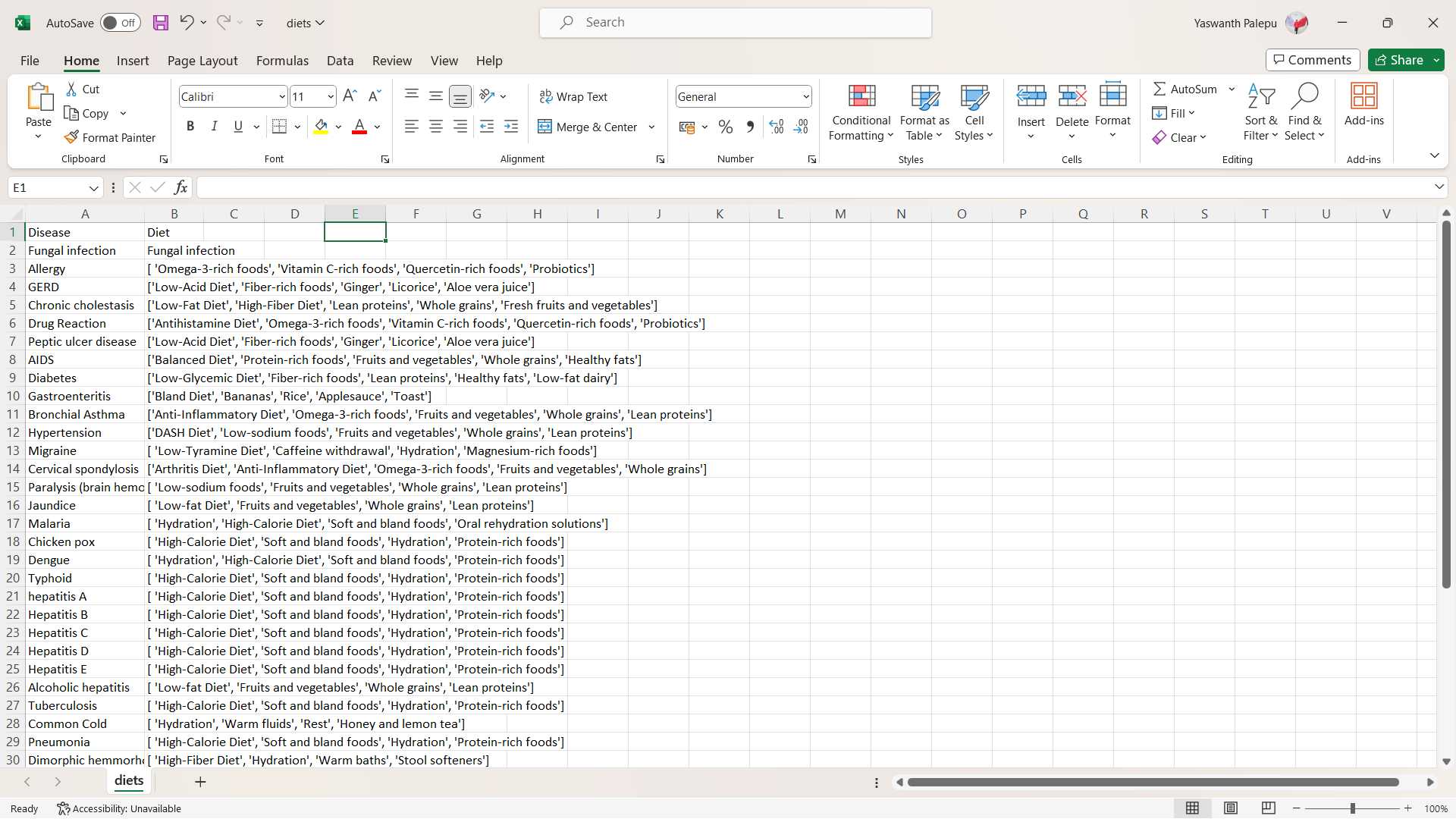
****

Figure 8.6 diet dataset

In the above Figure 8.6 contains the suggestable diet for the disease related to respective symptoms.

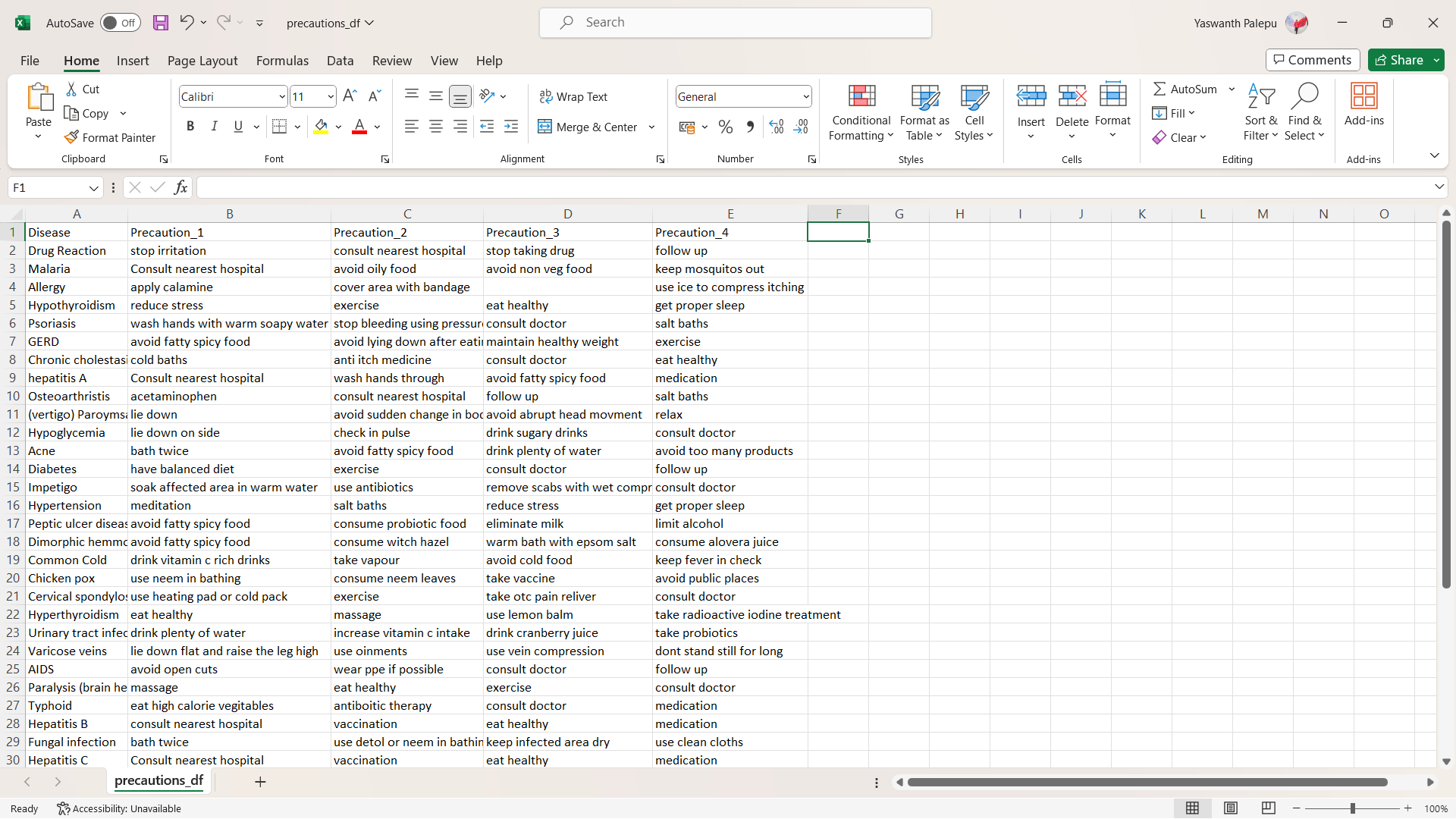


Figure 8.7 precautions dataset

In the above Figure 8.7 contains the suggestable precaution for the disease related to respective symptoms. It has 5 attributes and 41 records.

## Preprocessing

Clean the data by removing any irrelevant or missing values and normalize the features if necessary.

1. **Handling Missing Values:** Check for missing values in the dataset, which can arise due to sensor errors or data collection issues.
2. **Removing Irrelevant Features:** Identify and remove any features that are not relevant to the prediction task. For water quality prediction, features like timestamp or location coordinates may not be useful and can be discarded.
3. **Data Splitting:** Split the pre-processed data into training and testing sets. The training set is used to train the model, while the testing set is used to evaluate its performance.

## Functionality of the System

We will discuss the implementation details of the disease prediction and medicine recommendation system. We provide an overview detailing the functionality of each component at a block level.

### Input Processing Block

The Input Processing Block is responsible for receiving input data from the user, which typically includes symptoms and patient information. This block preprocesses the input data and passes it to the Disease Prediction Block for further analysis.

**Inputs:** User-provided symptoms, patient information.

**Outputs:** Preprocessed input data.

### Disease Prediction Block

The Disease Prediction Block utilizes machine learning algorithms to analyze the input data and predict the likelihood of various diseases based on the provided symptoms. This block employs algorithms such as Naïve Bayes, Decision Tree, and Random Forest to generate accurate predictions.

**Inputs:** Preprocessed input data.

**Outputs:** Predicted disease probabilities for each potential illness.

### Medicine Recommendation Block

The Medicine Recommendation Block takes the predicted disease probabilities from the Disease Prediction Block and recommends suitable medications. This block gives personal recommendations such as medications, description about the disease, diet, and precautions.

**Inputs:** Predicted disease probabilities.

**Outputs:** Recommended medications, diet, and precautions for the identified diseases.

## GitHub Repository

Explore our project's source code, contribute, and stay updated by visiting our GitHub repository. Click [**https://github.com/YaswanthPalepu/Disease-prediction-and-medicine-recommendation-system.git**](https://github.com/YaswanthPalepu/Disease-prediction-and-medicine-recommendation-system.git) to access the repository. We welcome contributions, feedback, and collaborations to further enhance our disease prediction and medicine recommendation system.

# Results

## Code

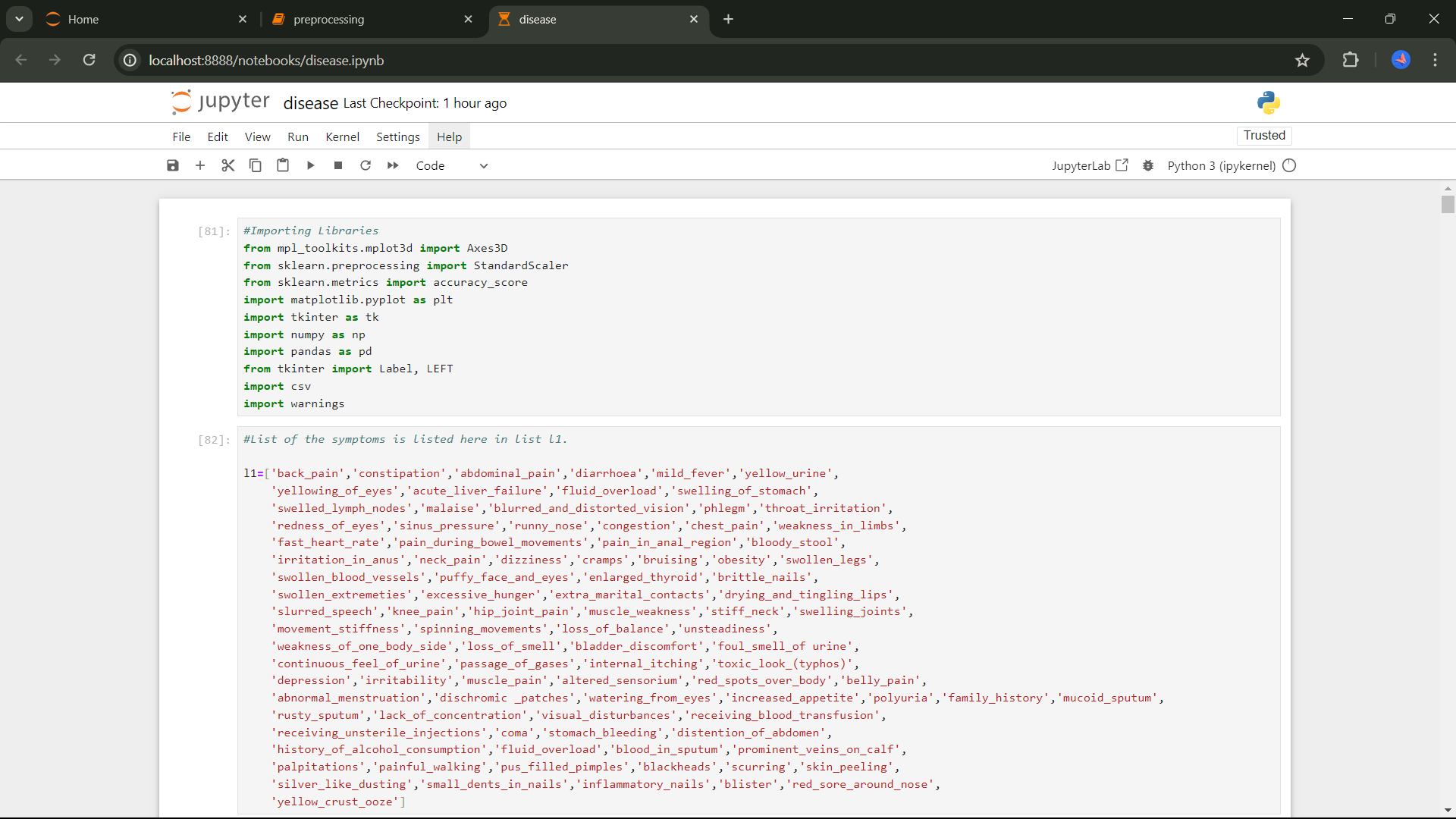


Figure 9.1 code in jupyter notebook

The Figure 9.1 shows the code presented in the jupyter notebook after clicking on run.

## GUI

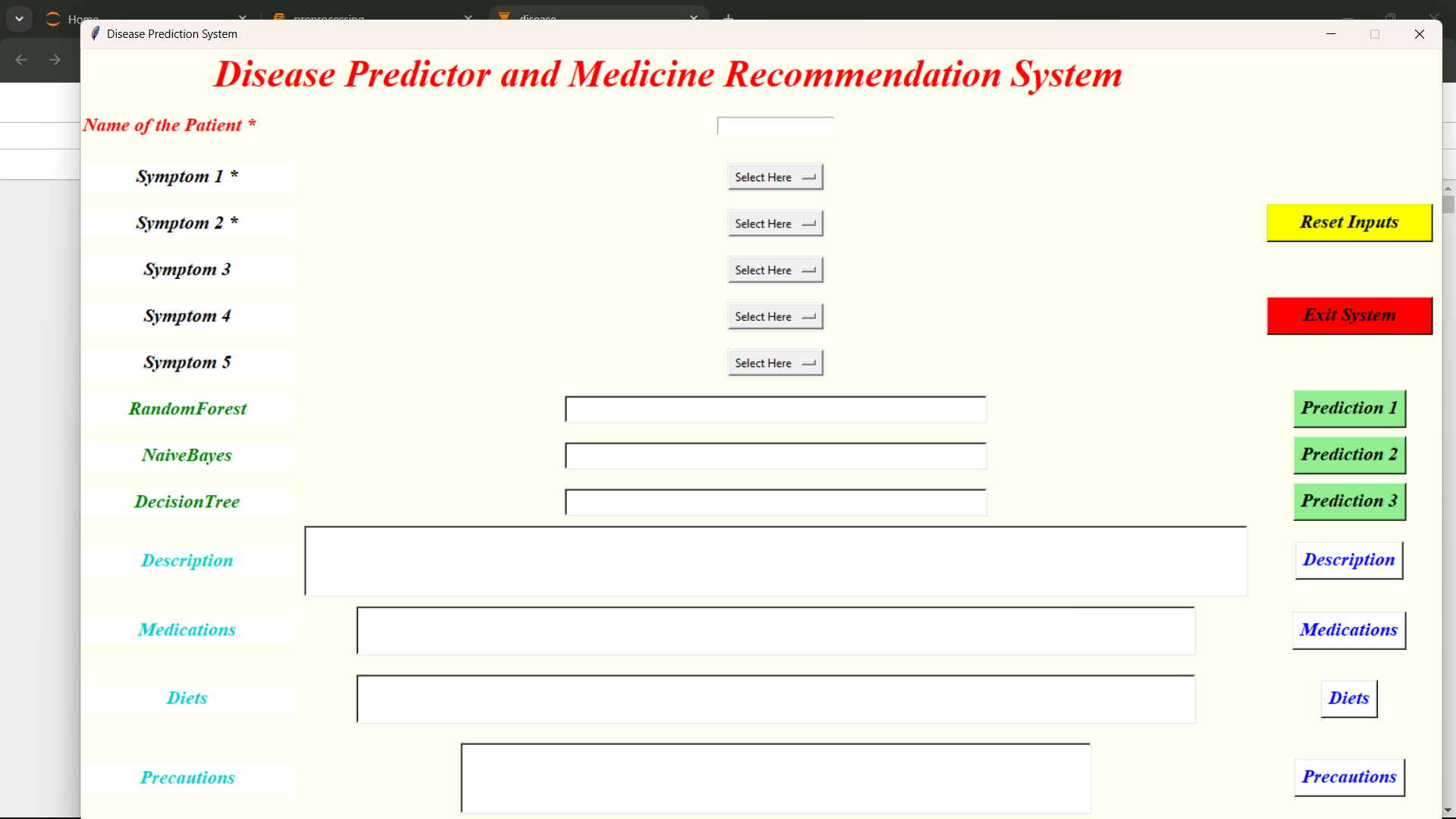


Figure 9.2 Output home page

The results are shown in the Figure 9.2 after the code has been executed.

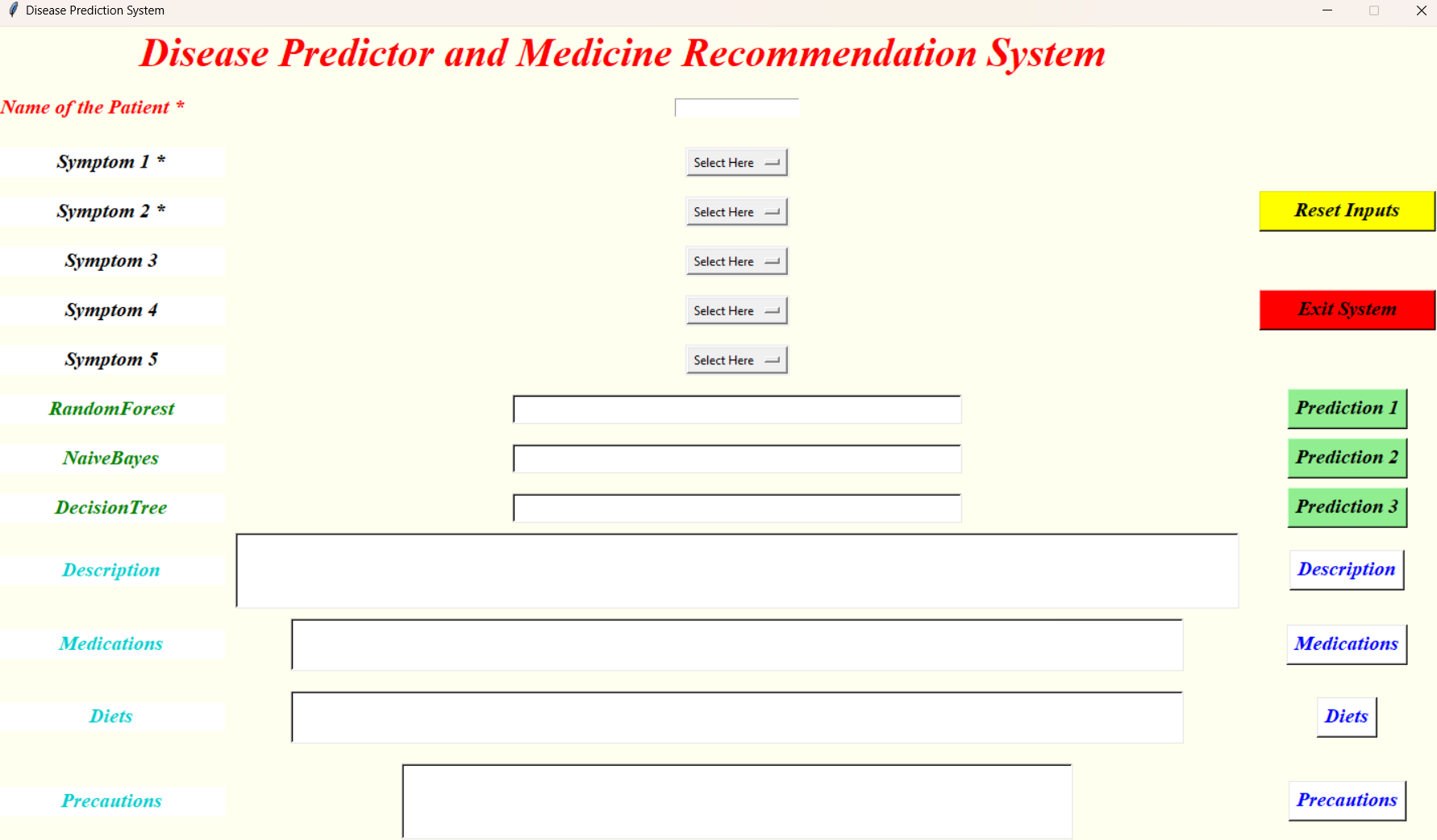


Figure 9.3 Tkinter system for predicting disease

In the Figure 9.3, we could see the home page showing the concept of Disease prediction and medicine recommendation system, by using Symptoms as inputs i.e., symptoms with respective list boxes, Buttons of algorithms beside their text boxes and in bottom, we have labels and their equivalent text boxes to display the disease, description, medications, diet, and precautions.

In the Figure 9.4, we could see the output home page showing the concept of symptoms as input to predict the disease.

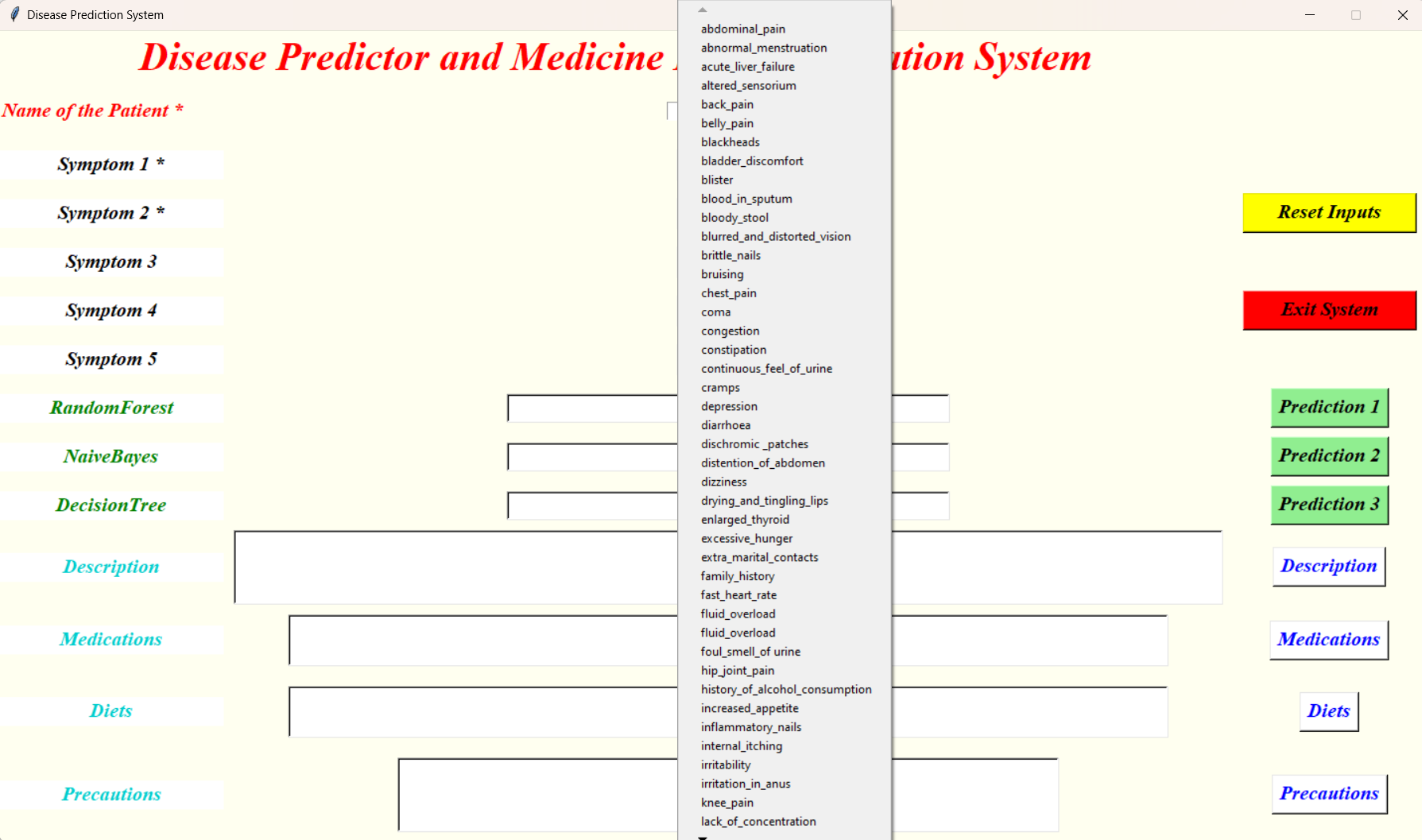


Figure 9.4 Symptoms in list box

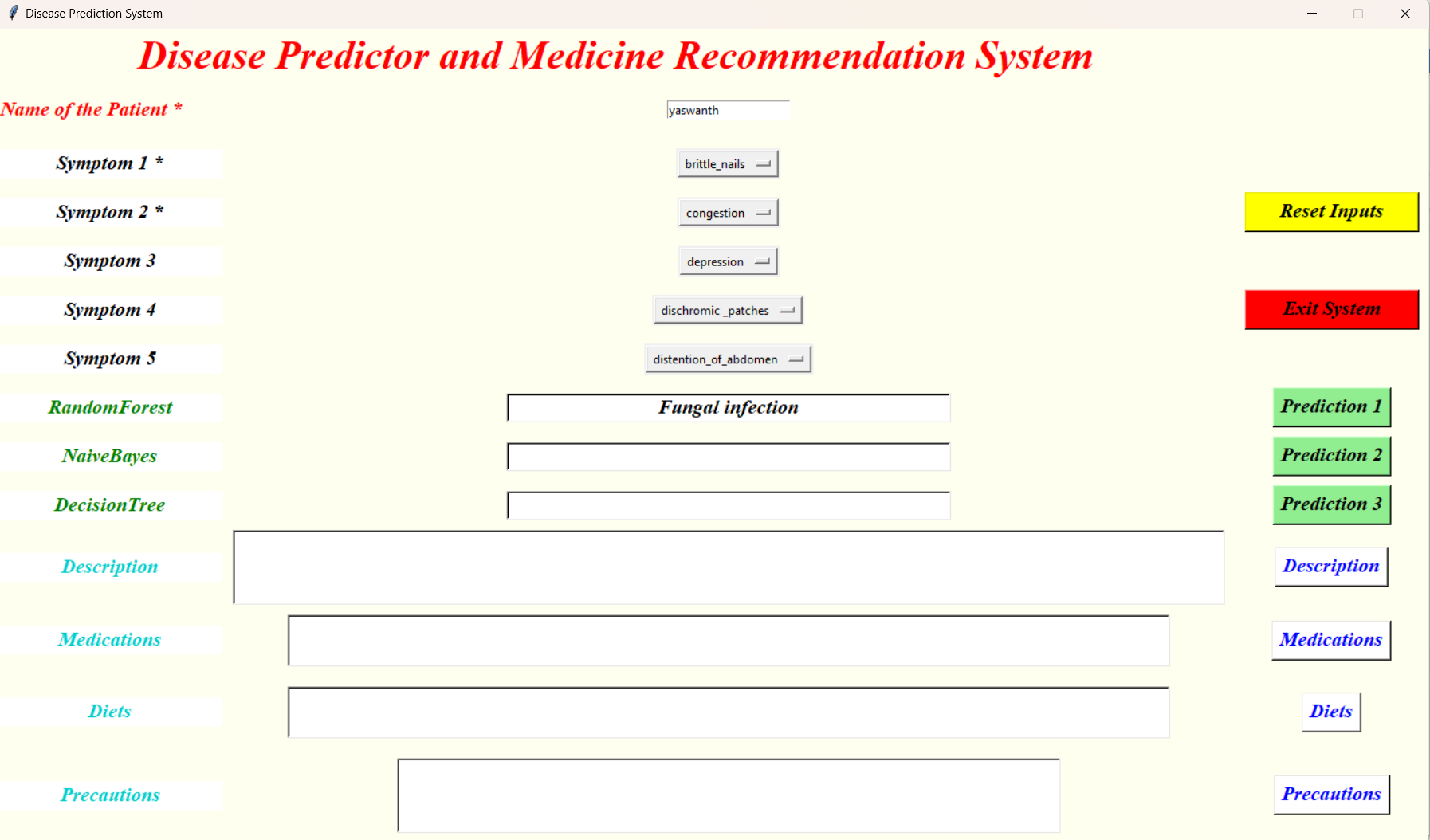


Figure 9.5 Random Forest Prediction

In the Figure 9.5, we could see the output showing the predicted disease and accuracy obtained on executing the source code for Random Forest algorithm.

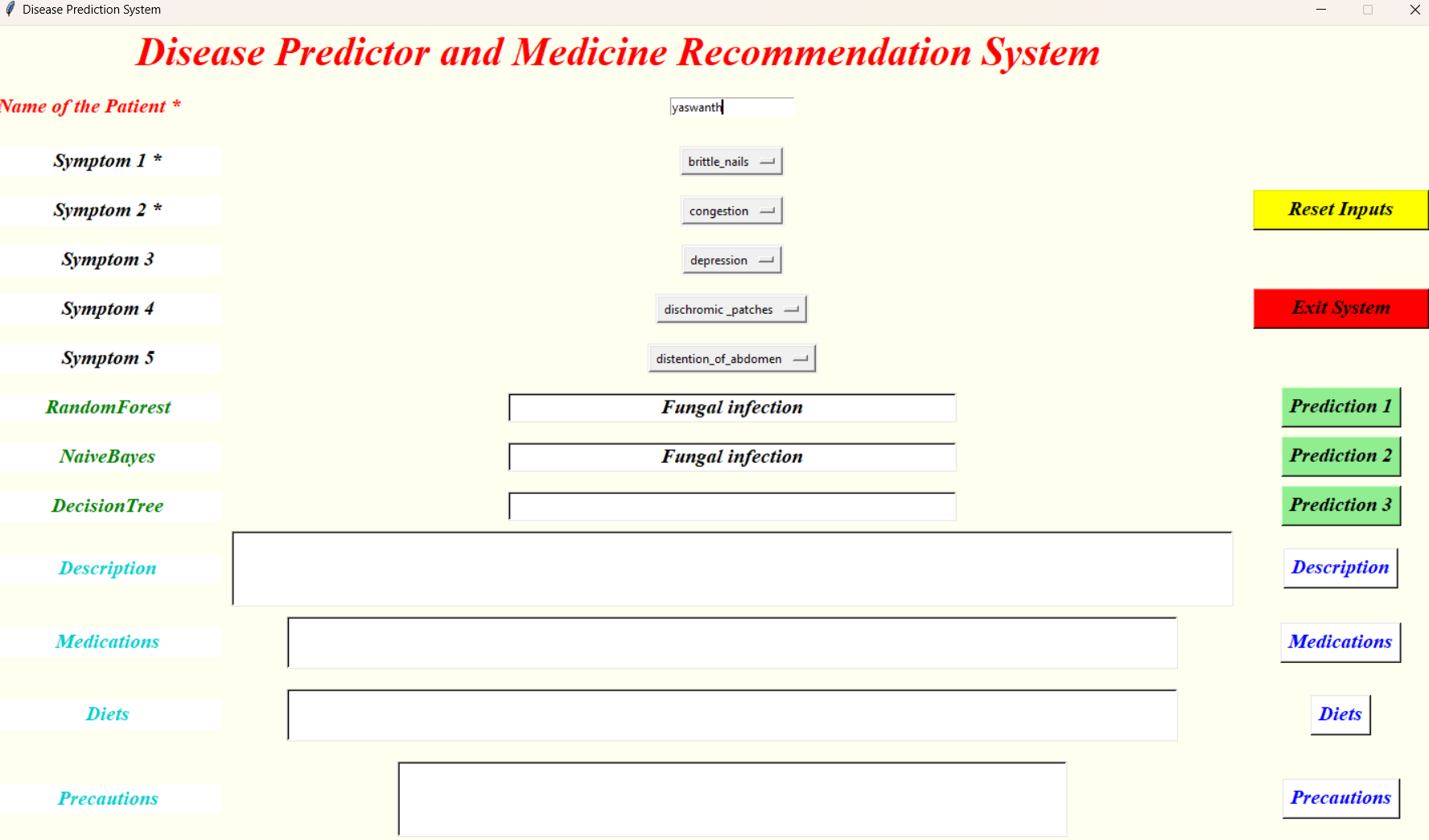


Figure 9.6 Naïve Bayes prediction

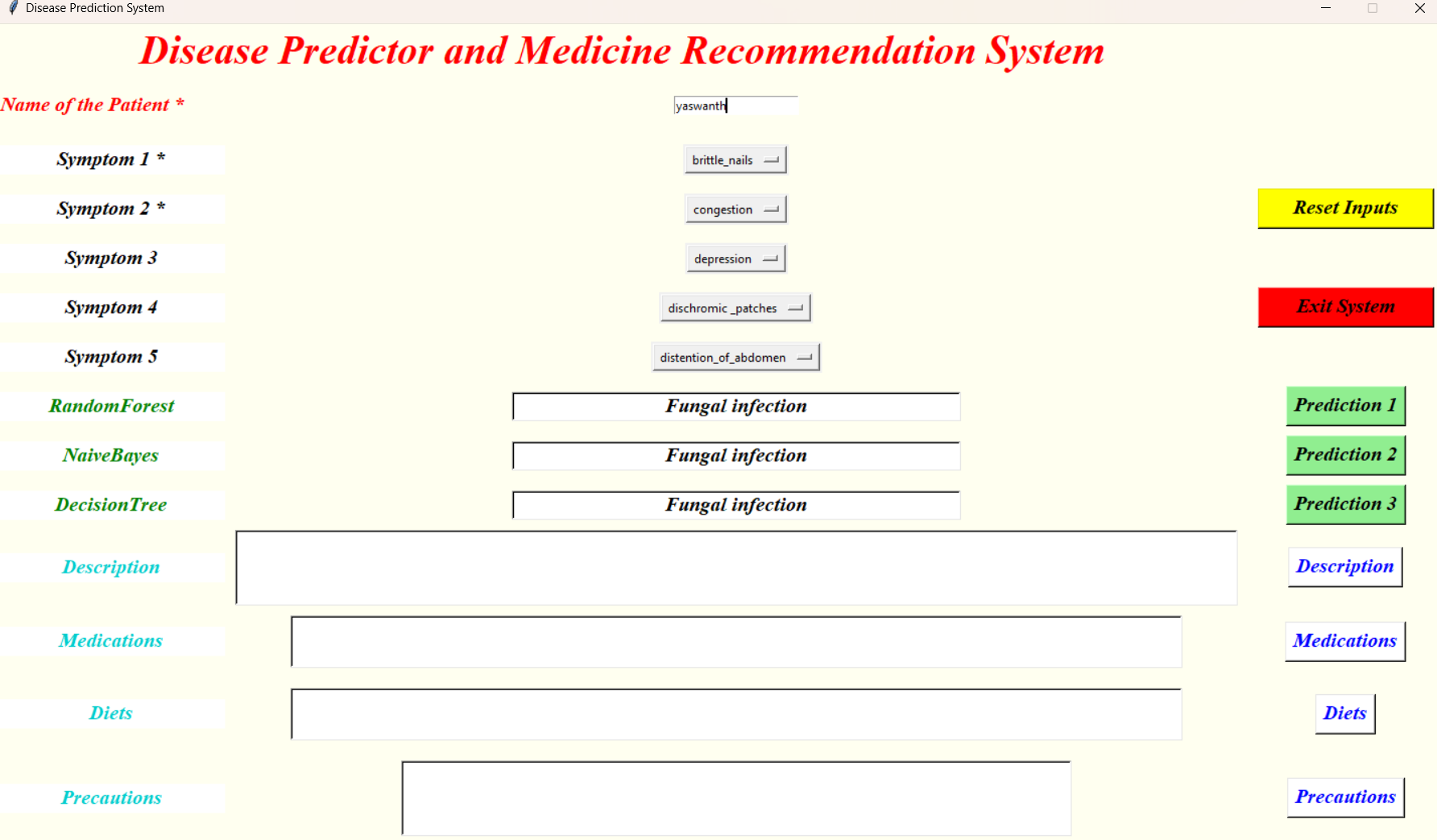


Figure 9.7 Decision tree prediction

In the Figure 9.6, we could see the output showing the predicted disease and accuracy obtained on executing the source code for Navie Bayes algorithm.

In the Figure 9.7, we could see the output showing the predicted disease and accuracy obtained on executing the source code for Decision Tree algorithm.

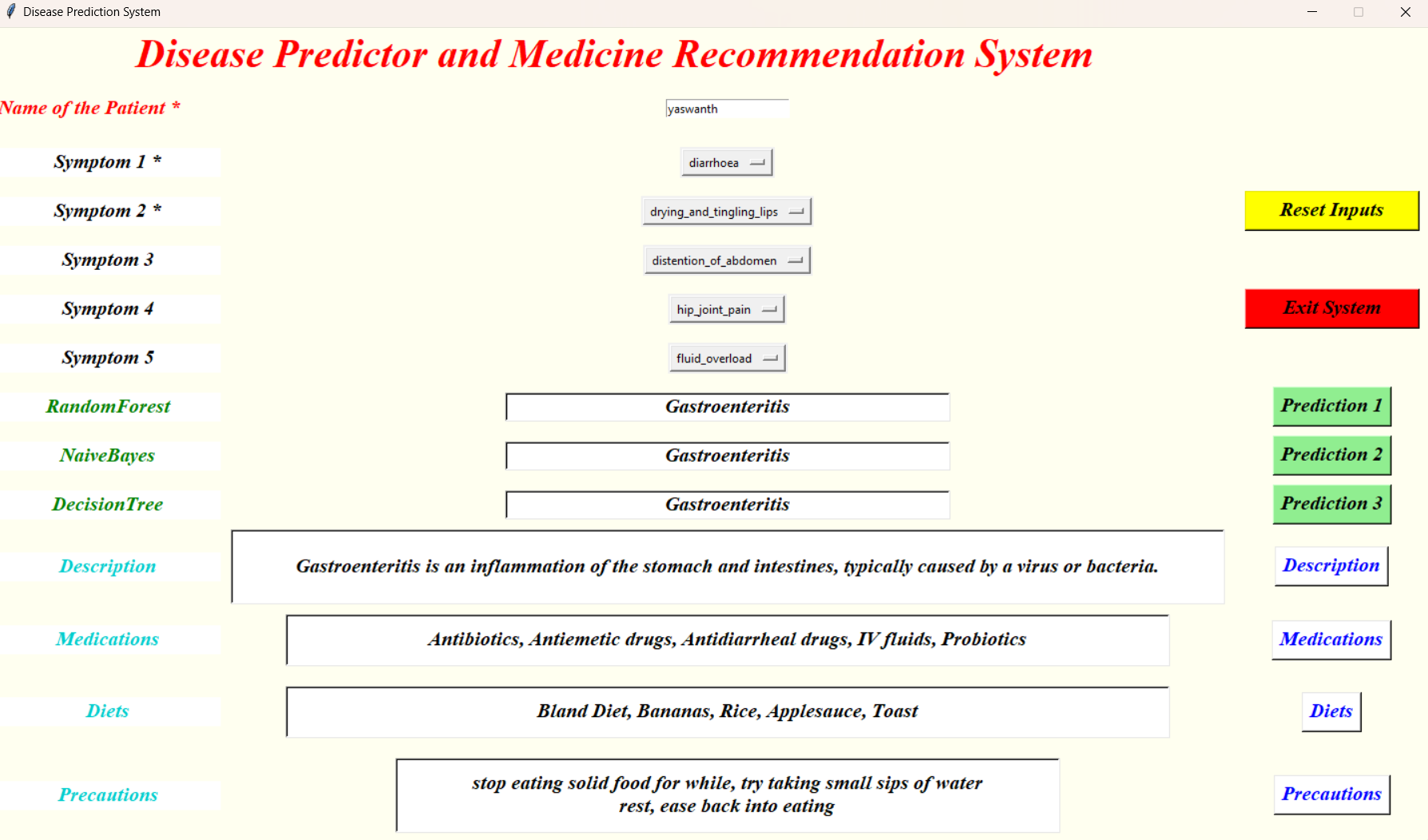


Figure 9.8 Prescribing medications associated with diet and precaution

In the Figure 9.8, Based on the disease predicted by the system it suggest some medications, diet, precautions, and description about the disease.

# Conclusion

In this work a disease prediction and medicine recommendation system has been developed using various machine learning algorithms like Naïve Bayes, Decision Tree and Random Forest. The system has been trained by mapping the various symptoms of the diseases in the dataset. Disease prediction level has also been analyzed based on the classify by the different classifiers. Moreover, our system also recommends the suitable medicine for the predicted diseases.

In this experiment we found that Random Forest gives the better accuracy (approx. 95%) than the Navie Bayes Classifier (approx. 94%) and Decision Tree algorithms (approx. 93%). This system can also analyses the mix of medicine for the predicted disease. The system's user interface, which uses symptoms as inputs (symptoms with corresponding list boxes, and algorithm buttons next to text boxes) and labels at the bottom that display the disease, description, medications, diet, and precautions.

# Future work

In future research focus on Incorporate additional healthcare datasets, such as genetic information, environmental factors, and patient demographics, to improve the accuracy and scope of disease prediction. Explore advanced machine learning techniques, including deep learning and ensemble methods, to further enhance the predictive capabilities of the system and accommodate complex data patterns.

Integrate the system with existing EHR systems to streamline data exchange and enable seamless access to patient information, facilitating more personalized and comprehensive healthcare recommendations. Extend the system's capabilities to telemedicine platforms, allowing remote healthcare providers to leverage its disease prediction and medication recommendation features to support virtual consultations and remote patient monitoring.

# References

|  |  |
| --- | --- |
| [1] | S. K. Nayak, m. garanayak, S. K. swain, S. K. Panda and D. Godavathi, "An Intelligent Disease Prediction and Drug Recommendation Prototype by Using Multiple Approaches of Machine Learning Algorithms," *IEEE Access,* vol. 11, no. 99304-99318, p. 16, 2023. |
| [2] | Jay Prakash Gupta, Ashutosh Singh and Ravi Kant Ku, "A Computer based Disease Prediction and Medicine Recommendation system using machine learning approach," *IJARET,* vol. 12, no. 3, pp. 673-683, 2021. |
| [3] | Varun A.Goyal, Dilip J. Parmar, Namaskar I. Joshi and Prof. Komal Champanerkar, "Medicine Recommendation system," *IRJET,* vol. 07, no. 03, 2020. |
| [4] | A. N. V. K. Swarupa, V. H. Sree, S. Nookambika, Y. K. S. Kishore and U. R. Teja, "Disease Prediction: Smart Disease Prediction System using Random Forest Algorithm," *IEEE International Conference on Intelligent Systems, Smart and Green Technologies,* pp. 48-51, 2021. |
| [5] | Ayrine John, Muhammed Ilyas H and Veena Vasudevan, "Medication Recommendation System Based on Clinical Documents," *International Conférence on Information Science (ICIS),* pp. 180-184, 2016. |
| [6] | Constanze Knahl, Mert Aydin and Karim Elish, "A Literature Review on Medicine Recommender," *International Journal of Advanced Computer Science and Applications,* vol. 10, no. 8, pp. 6-13, 2019. |