

# **Disease Prediction based on symptoms using machine learning algorithms**

**A PROJECT REPORT**

*Submitted by*

**Mr. T.G. Pranav Sri Vasthav - 20181CSE0738**

**Ms. Tejaswini G J - 20181CSE0746**

**Ms. Vada Swetha - 20181CSE0762**

**Ms. Vanishree R - 20181CSE770**

*Under the guidance of*

**Raghavendra T.S**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**At**



**Department of Computer Science and Engineering**

**School of Engineering**

**PRESIDENCY UNIVERSITY**

**BANGALORE**

**JUNE 2022**

**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING**

**SCHOOL OF ENGINEERING**

**PRESIDENCY UNIVERSITY**

**CERTIFICATE**

This is to certify that the Project report “**DISEASE PREDICTION BASED ON SYMPTOMS USING MACHINE LEARNING ALGORITHMS**” is being submitted by “T.G.Pranav Sri Vasthav, Tejaswini. G.J, Vada Swetha, Vanishree. R” bearing roll numbers “20181CSE0738, 20181CSE0746, 20181CSE0762, 20181CSE0770” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

**Dr. C. KALAIARASAN**  
Associate Dean-Admin  
Department of CSE  
Presidency University

**MR. Raghavendra T S**  
Assistant Professor  
Department of CSE  
Presidency University

**DEPARTMENT OF COMPUTER SCIENCE AND  
ENGINEERING  
SCHOOL OF ENGINEERING  
PRESIDENCY UNIVERSITY**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **DISEASE PREDICTION BASED ON SYMPTOMS USING MACHINE LEARNING ALGORITHMS** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering**, is a record of our investigations carried under the guidance of **Mr.RAGHAVENDRA T S, Assistant Professor, Department of Computer Science and Engineering, School of Engineering, Presidency University, Bangalore.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

**T.G. Pranav Sri Vasthav 20181CSE0738  
Tejaswini G J - 20181CSE0746  
Vada Swetha - 20181CSE0762  
Vanishree R - 20181CSE770**

**Abstract:**

Every day, we hear about the discovery of a new disease or new symptoms of an old condition. Predicting an illness by looking at the symptoms is an important aspect of treatment. Most of the time, doctors find it challenging to precisely identify ailments by hand. In the digital technology era, the globe must create an outstanding health system to ensure that citizens and communities are alive and well. As a result, healthcare practitioners require precise forecasts of the outcomes of various ailments that patients are suffering from. Furthermore, timing is an important factor that influences clinical decisions for precise forecasts.

As a result, various symptoms and sicknesses are supplied into this system in order to overcome this obstacle. Users can communicate their symptoms and problems through the system. It then examines the user's symptoms to see if there are any illnesses that might be linked to them and outputs the disease's probability. Support Vector Machine, Random Forest, Naive Bayes, and Decision Tree Classifiers are used to predict disease. The disease's likelihood is calculated by this algorithm. Whole work includes Registration, OTP login, uploading files, pre-processing the data, Splitting the data, choosing the symptoms then predicting the disease.

The main motto to develop this system is to provide better accuracy for the prediction and make it more user-friendly to ensure the citizens and community are alive and healthy.

---

## List of Figures

Sl. No.	Figure Name	Caption	Page No.
1	FIGURE2.1.1:	FRONTEND ROADMAP	3
2	FIGURE2.1.2:	BACKEND ROADMAP	4
3	FIGURE5.1.2.1:	RANDOM TREE CLASSIFIER	19
4	FIGURE5.1.2.2:	RANDOM TREE ALGORITHM EXAMPLE	20
5	FIGURE5.1.3.1:	DECISION TREE CLASSIFIER	23
6	FIGURE5.1.4.1:	SUPPORT VECTOR MACHINE	25
7	FIGURE 7.2.1	READING TRAINING DATA	31
8	FIGURE 7.2.2	DEALING WITH MISSING VALUES	31
9	FIGURE 7.2.3	CHECKING DATA TYPES	32
10	FIGURE 7.2.4	SPLITTING THE DATA (1)	32
11	FIGURE 7.2.5	SPLITTING THE DATA (2)	33
12	FIGURE 7.2.6	MODEL BUILDING	33
13	FIGURE 7.2.7	GAUSSIAN NAÏVE BAYES	34
14	FIGURE 7.2.8	RANDOM FOREST	34
15	FIGURE 7.2.9	DECISION TREE	35
16	FIGURE 7.2.10	SUPPORT VECTOR MACHINE	35
17	FIGURE 7.3.1	HOME PAGE	36

18	FIGURE 7.3.2	DESTINATION PAGE	36
19	FIGURE 7.3.3	REGISTRATION PAGE	37
20	FIGURE 7.3.4	LOGIN PAGE	37
21	FIGURE 7.3.5	DATA TABLE	37
22	FIGURE 7.3.6	INDEX PAGE	38
23	FIGURE 7.3.7	DATASET UPLOAD ACTION PAGE	38
24	FIGURE 7.3.8	SPLITTING DATASET PAGE	39
25	FIGURE 7.3.9	ALGORITHM PAGE	39
26	FIGURE 7.3.10	DISEASE PREDICTION PAGE	40
27	FIGURE 7.3.11	CONTACT US PAGE	40

---

---

## **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>ABSTRACT</b>	IV
	<b>ACKNOWLEDGEMENT</b>	VII
1.	<b>INTRODUCTION</b>	1
2.	<b>REQUIREMENTS ANALYSIS</b>	
	2.1: Technologies Used/Required:	2
	2.1.1: Machine Learning	2
	2.1.2: Web Development	3
	2.2: Software Used:	5
	2.2.1: PyCharm	5
	2.2.2: Xampp	5
	2.2.3: SQLyog	6
3.	<b>LITERATURE REVIEW</b>	7
4.	<b>EXISTING SYSTEM</b>	
	4.2: Related Work	14
	4.2: Drawbacks	14

5.	<b>PROPOSED WORK</b>	
	5.1: Algorithm	15
	5.1.1: Gaussian Naive Bayes:	16
	5.1.2: Random Forest Classifier:	18
	5.1.3: Decision Tree Classifier:	22
	5.1.4: Support Vector Machine:	25
	5.2: User Interface	27
6.	<b>SYSTEM DESIGN</b>	
	6.1: Flowchart	29
7.	<b>IMPLEMENTATION</b>	
	7.1: Divisions	30
	7.2: Training and Testing	31
	7.3 Website Implementation	36
8.	<b>TESTING</b>	41
9.	<b>CONCLUSION</b>	48



## ACKNOWLEDGEMENT

First of all, we are indebted to the **GOD ALMIGHTY** for allowing me to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Abdul Sharief**, Dean, School of Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved professors **Dr. C. Kalaiarasan** and **Dr.Mohamadi Begum** University Project-II In-charge, Associate Dean, Department of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide Prof. Name, Designation, Department of Computer Science and Engineering, Presidency University for his/her inspirational guidance, and valuable suggestions, and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We thank our friends for the strong support and inspiration they have provided us in bringing out this project.

**T.G.PRANAV SRI VASTHAV**  
**TEJASWINI G J**  
**VADA SWETHA**  
**VANISHREE R**

# CHAPTER-1

## Introduction:

Medicine and healthcare are critical components of the economy and human life. In the world we now live in, there has been a huge amount of change. Medical experts are having difficulty analyzing symptoms effectively and identifying diseases at an early stage due to large volumes of data. Machine Learning algorithms have shown tremendous promise in outperforming traditional illness diagnosis systems and assisting medical professionals in the early detection of high-risk disorders.

Machine learning is the process of teaching computers how to maximize their performance based on previous data or examples. Machine learning is the study of data-driven and experience-driven computer systems. The training and testing tracks of a machine learning system are separated. The patient's symptoms and medical history are used to predict the disease. In previous decades, machine learning technology has progressed. In the medical field, machine learning technology provides an unrivaled platform for quickly resolving healthcare challenges. We're using machine learning to keep track of all of the data from the hospital. Machine learning technology assists doctors to make better decisions about patient diagnoses and treatment options by allowing them to swiftly develop models, analyze data, and deliver results. The most prominent example of machine learning in the medical industry is healthcare.

Thus, in this system, we are concentrating on providing immediate and accurate disease prediction to the users about the symptoms they choose using different machine learning algorithms such as Support Vector Machine, Random Forest Classifier, Naïve Bayes and Decision Tree Classifier and then deploying the model into the website to be more user-friendly for the users. Whole work includes Registration, OTP login, uploading files, pre-processing the data, Splitting the data, choosing the symptoms then predicting the disease.

## **CHAPTER-2**

### **Requirement Analysis:**

#### **2.1Technologies Used/Required:**

##### **2.1.1 Machine Learning**

Artificial Intelligence and Machine Learning are two buzzwords that are often misunderstood in today's world. Artificial Intelligence (AI) includes Machine Learning (ML). ML is the science of creating and implementing algorithms that can learn from previous experiences. If you've observed a pattern of behavior before, you can predict if it'll happen again. That is to say, no predictions can be made if no past examples exist.

Applications:

ML can be used to address difficult problems like detecting credit card fraud, enabling self-driving automobiles, and detecting and recognizing faces.

## 2.1.2 Web Development

The term "web development" refers to the process of designing, developing, and managing websites. Web design, web publishing, web development, and database administration are all included. It is the building of an internet-based application, such as a website. The term "web development" is made up of two words: "web" and "development." Websites, web pages, and anything else that functions through the internet are referred to as the web. Development is the process of creating an application from the ground up.

There are two methods to categorize web development:

- Frontend Development
- Backend Development

### Frontend RoadMap

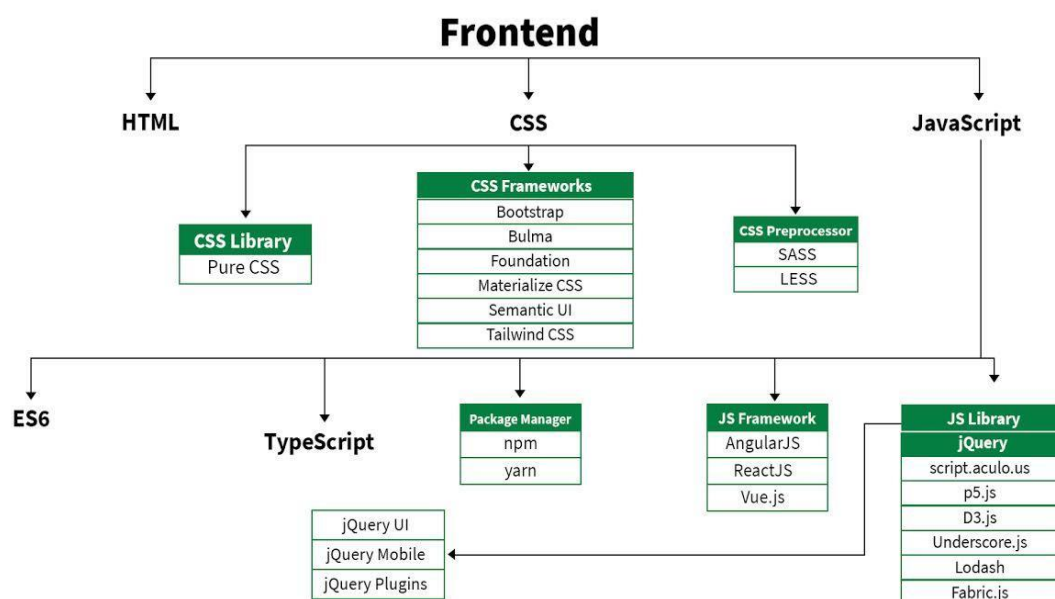


FIGURE 2.1.1: Frontend RoadMap

## Backend RoadMap

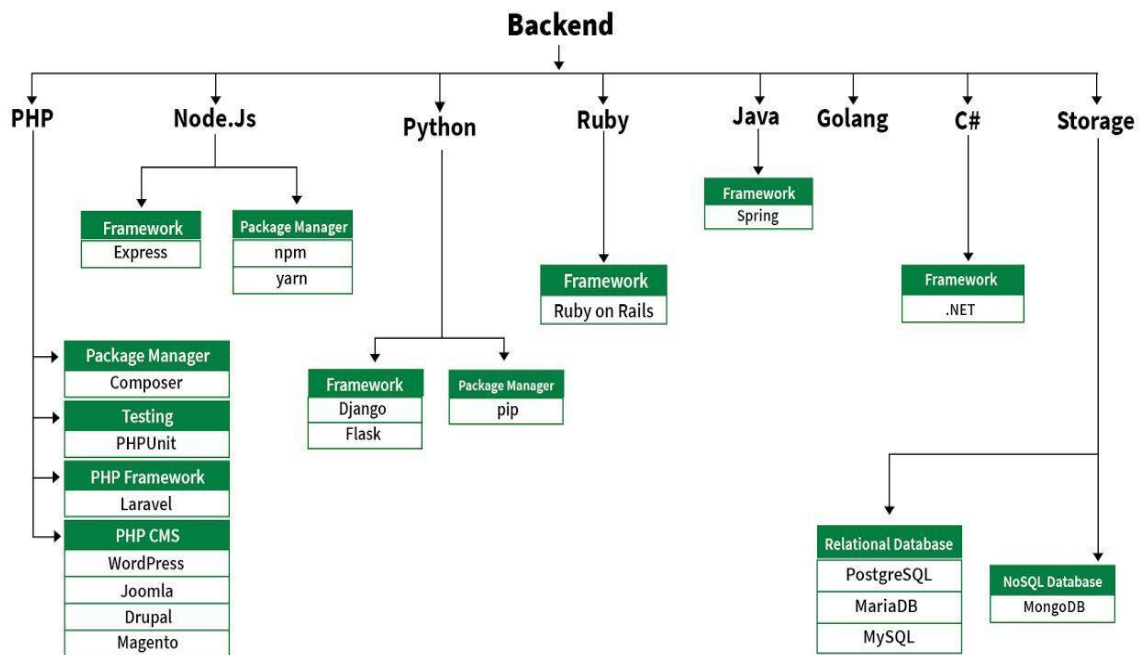


FIGURE2.1.2: Backend RoadMap

## 2.2 Software Used:

### 2.2.1 PyCharm

PyCharm is the most popular IDE used for Python scripting language. PyCharm offers some of the best features to its users and developers in the following aspects:

Code completion and inspection

Advanced debugging

Support for web programming and frameworks such as Django and Flask

### 2.2.2 Xampp

XAMPP is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver.

Advantages:

- It's easy to set up compared to other web servers like WAMP.
- It's Multi Cross-Platform, meaning it'll run on both Windows and Linux.
- You can start and terminate the full web server and database stack with a single command.
- It has a control panel that you can see contains start and stop buttons for specific mechanisms, such as Apache, which is running through its Control Panel.

Disadvantage:

- In comparison to the WAMP server, configuration and setting are more difficult.

### 2.2.3 SQLyog

SQLyog is a user-friendly graphical interface for managing MySQL servers and databases in physical, virtual, and cloud settings. SQLyog is not reliant on any runtimes (such as Microsoft .NET and Java)

SQLyog runs on Microsoft Windows. It may also run-on Linux and Unix via Wine.

## **CHAPTER-3**

### **LITERATURE REVIEW**

#### **1. Disease Prediction by Machine Learning Over Big Data from Healthcare Communities (2017)**

- **AUTHORS:**  
MIN CHEN, (Senior Member, IEEE), YIXUE HAO, KAI HWANG, (Life Fellow, IEEE), LU WANG, AND LIN WANG
- **METHODS /ALGORITHMS /TECHNIQUES USED:**  
Big data analytics, machine learning, Supervised and Unsupervised
- **MERITS:**  
work focused on both data types in the area of medical big data analytics.
- **DEMERITS:**  
accuracy of our proposed algorithm reaches only 94.8%



## 2. Disease Prediction Using Machine Learning (May-June-2021)

- **AUTHORS:** Gaurav Shilimkar, Shivam Pisal
- **METHODS /ALGORITHMS /TECHNIQUES USED:**  
Data Mining and Machine Learning Techniques, Decision Tree Algorithm
- **MERITS:**  
Decision trees are fairly easy models to comprehend
- **DEMERITS:**  
The present system covers only the general illnesses or the commonly occurring diseases

### 3. Implementation of Web Application for Disease Prediction Using AI (2020)

- AUTHORS: Manasvi Srivastava, Vikas Yadav, Swati Singh

- METHODS /ALGORITHMS /TECHNIQUES USED:

Web scraping, Python, Linear Regression

- MERITS:

Analysis performed showed a very similar disease

- DEMERITS:

The training model lacks the size of the database

**4. Machine learning-based disease prediction website using symptoms of a patient (2020)**

- **AUTHORS:** Varinder Garg, Harish Kumar, Surinder Rana, Bikramjeet Singh Kalsi, Siddhant Mukherjee
- **METHODS /ALGORITHMS /TECHNIQUES USED**  
machine learning decision Tree Classifier, FLASK, video-conferencing

**5: Multiple Disease Prediction Using Different Machine Learning Algorithms Comparatively**  
(2019)

- **AUTHORS:** Rudra A.Godse, Smita S. Gunjal, Karan A. Jagtap, Neha S. Mahamuni, Prof. Suchita Wankhade
- **METHODS /ALGORITHMS /TECHNIQUES USED:**  
Machine Learning Algorithms, Django, Python
- **MERITS:**  
Using many different ML Algorithms gives a faster and more effective result.

**6. Machine learning-based disease prediction website using symptoms of a patient (2020)**

- **AUTHORS:** Varinder Garg, Harish Kumar, Surinder Rana, Bikramjeet Singh Kalsi, Siddhant Mukherjee
- **METHODS /ALGORITHMS /TECHNIQUES USED:**  
machine learning decision Tree Classifier, FLASK, video-conferencing
- **MERITS:**  
Tested with different symptoms for different diseases.  
Google Maps feature has been used.
- **DEMERITS:**  
Prediction of disease result is not accurate i.e.70-75%

## 7. Multiple Disease Prediction Using Different Machine Learning Algorithms Comparatively (2019)

- **AUTHORS:** Rudra A. Godse, Smita S. Gunjal, Karan A. Jagtap, Neha S. Mahamuni, Prof. Suchita Wankhade
- **METHODS /ALGORITHMS /TECHNIQUES USED:**  
Machine Learning Algorithms, Django, Python
- **MERITS:**  
Using many different ML Algorithms gives a faster and more effective result.

## **CHAPTER-4**

### **EXISTING SYSTEM**

#### **4.1: RELATED WORK:**

In the existing methods, the user will add symptoms and the website will predict diseases based on a dataset prepared from different sources and good study and machine learning techniques applied to the dataset. For the dataset, they relied on a team of medical researchers to search for information from the internet which was verified, and the diseases were collected and classified into different categories. The data collected was used to discover different patterns with Neural networks, Decision trees, Support Vector Machines, etc. for different disease predictions, and the accuracy achieved was about 70-90%

#### **4.2: DRAWBACKS**

- Prediction of disease results is not accurate.
- Some of the data mining techniques are used which will not help for effective decision making.
- The website's user interface needs to be modified so that user gets a better experience of the website.
- The training model lacks in size of the database.
- Covers only general illnesses or the commonly occurring diseases.

## **CHAPTER-5**

### **PROPOSED WORK**

The main dairy of your project is to build the user interface for prediction of disease with sheer accuracy. Building the website both for the patient and the doctor utility services.

The steps we are taking to achieve this work is listed below

- ✓ Test and train with different algorithm
- ✓ Screening out and sticking with the best algorithm.
- ✓ User interface

### **5.1**

#### **Algorithm:**

In this we have four different algorithm.They are

- a. Gaussian Naive Bayes
- b. Random Tree Classifier
- c. Decision Tree Classifier
- d. Support Vector Machine



### **5.1.1 Gaussian Naive Bayes:**

#### **Defination:**

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

#### **For example**

A fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. The assumptions made are that each feature makes an independent and equal contribution to the outcome. The naïve Bayes Algorithm is a supervised learning algorithm and it is based on the Bayes theorem which is primarily used in solving classification problems.

#### **Working:**

Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

Naive Bayes is a classification technique that is based on Bayes' Theorem with an assumption that all the features that predicts the target value are independent of each other. It calculates the probability of each class and then pick the one with the highest probability. It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems.

Bayes' Theorem describes the probability of an event, based on a prior knowledge of conditions that might be related to that event.

## **Difference:**

The main difference between Naive Bayes(NB) and Random Forest (RF) are their model size. Naive Bayes model size is low and quite constant with respect to the data. The NB models cannot represent complex behavior so it won't get into over fitting. On the other hand, Random Forest model size is very large and if not carefully built, it results to over fitting. So, When your data is dynamic and keeps changing. NB can adapt quickly to the changes and new data while using a RF you would have to rebuild the forest every time something changes.

## **Advantages of Naive Bayes**

- This algorithm works very fast and can easily predict the class of a test dataset.
- You can use it to solve multi-class prediction problems as it's quite useful with them.
- Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds.
- If you have categorical input variables, the Naive Bayes algorithm performs exceptionally well in comparison to numerical variables.

## **Applications of Naive Bayes Algorithm**

As this algorithm is fast and efficient, you can use it to make real-time predictions.

- This algorithm is popular for multi-class predictions. You can find the probability of multiple target classes easily by using this algorithm.
- Email services (like Gmail) use this algorithm to figure out whether an email is a spam or not. This algorithm is excellent for spam filtering.
- Collaborative Filtering and the Naive Bayes algorithm work together to build recommendation systems. These systems use data mining and machine learning to predict if the user would like a particular resource or not.

We are using the Gaussian Naive Bayes algorithm of testing and training the data set, since we have continuous value of distribution. The accuracy for this we get **100%**.

### **5.1.2 Random Tree Classifier:**

#### **Definition:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

#### **How Random Forest works:**

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

**Ex:**

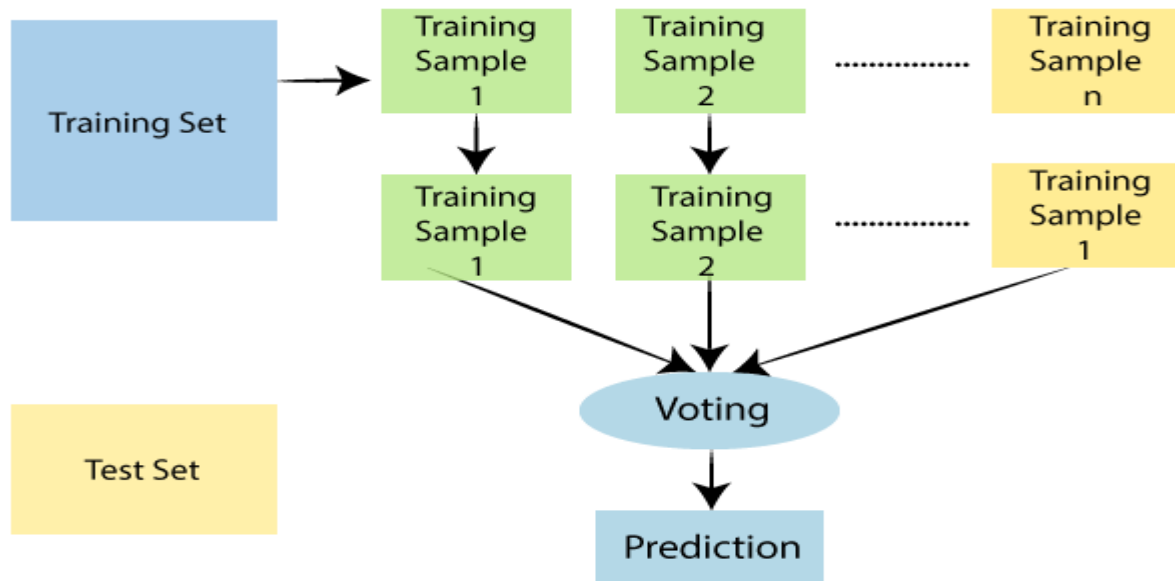


FIGURE5.1.2.1: Random Tree Classifier

### **Why Random Forest:**

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

### **How RF algorithm works:**

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes

**Ex:**

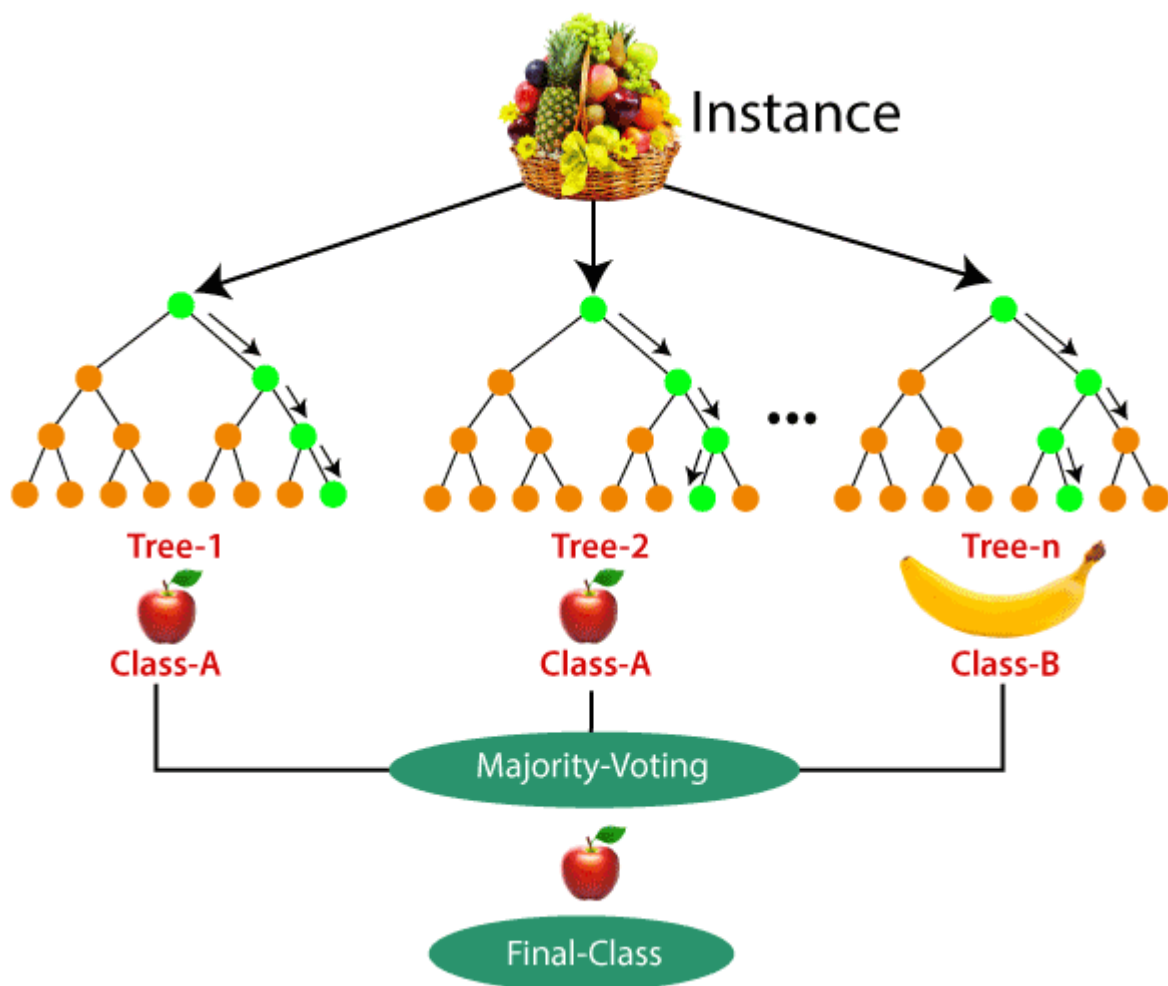


FIGURE5.1.2.2: Random Tree Algorithm

**Advantages:**

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

We selected this algorithm because the result will be displayed by considering the maximum number of outputs i.e. takes the majority vote rule. The accuracy we have achieved in this algorithm is also **100%**.

### **5.1.3 Decision Tree Classifier:**

#### **WHAT IS DECISION TREE CLASSIFIER?**

It is a supervised learning which can be used both for classification and regression, but it is widely used for solving classification model. It's a classifier module used to build the readable classification model, which is potentially accurate. It creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute.

#### **HOW THE MODEL WORKS?**

The model consists of mainly two nodes i.e. is decision node and the leaf node. Where the decision node is used for used to make any decision and have multiple branches, whereas Leaf node is the output of those decisions and do not contain any further branches.

The main reason for using DTC is because of the categorical numeric data (i.e. yes/no or 0/1)

#### **Algorithm flow:**

1. Begin the tree with the root node, says S, which contains the complete dataset.
2. Find the best attribute in the dataset using Attribute Selection Measure (ASM).
3. Divide the S into subsets that contains possible values for the best attributes.
4. Generate the decision tree node, which contains the best attribute.
5. Recursively make new decision trees using the subsets of the dataset created in 3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

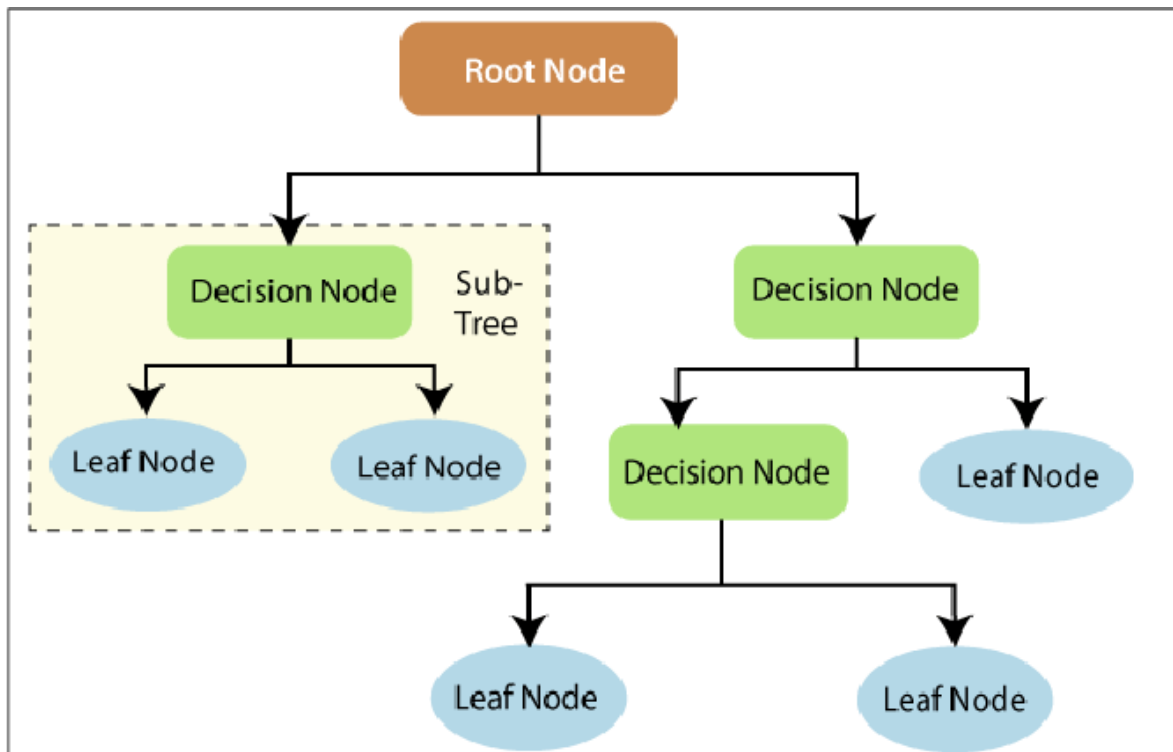


FIGURE5.1.3.1: Decision Tree Classifier

### Example:

Another example of binary classification would be deciding on whether circumstances are right to play tennis one morning. The instances of outlook ( that is, rain or sunny), temperature (hot or cold), humidity (high or low), wind (strong or not), etc. will form branches of the decision tree. A disjunction of different collected constraints will provide attributed value to the instance in question and in the present case, the instance being suitable weather to playtennis.

### Advantages of DTC

- Easy to understand, interpret, visualize.
- It can handle both continuous and categorical variables.
- While utilizing the decision tree algorithm, it is not necessary to credit the missing values.
- Normalization is not required in the Decision Tree.
- Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.



We consider this algorithm because we have the categorical data in the dataset. The accuracy we recorded for this algorithm is not that constant. Sometimes we get **100%** **and** sometimes we get **99.93%**

### 5.1.4 Support Vector Machine:

#### Definition:

“Support Vector Machine” (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

#### How does SVM model work?

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane.

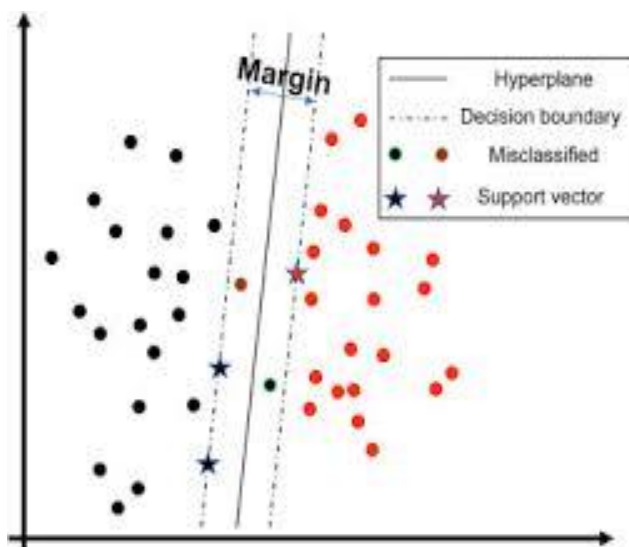


FIGURE 5.1.4.1: Support Vector Machine

## **Pros associated with SVM**

- o It works really well with a clear margin of separation
- o It is effective in high dimensional spaces.
- o It is effective in cases where the number of dimensions is greater than the number of samples.
- o It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

## **Applications:**

- Data Classification using SVM.
- Facial Expression Classification.
- Texture Classification using SVM.
- Text Classification.
- Speech Recognition.

The last algorithm we included for testing and training the dataset is SVM. We tried this algorithm since it can handle the high dimensionality data. The accuracy we noted for this algorithm is **99.79%**.

### 5.3

#### **User Interface:**

To enhance the project we are building the user interface model for this project. It consists of features like

- a. Home page
- b. Registration
- c. Login/Logout
- d. OTP verification
- e. Selection of algorithm
- f. Selection of Symptoms
- g. Disease prediction

In the Home page the user that is doctor or patient can access the features like registering themselves if they have not register yet or they may directly just clock on login to access the services available in the website. After the registration they can login using their name and their password. We are using mysql to store the data and the information of the user.

In this project we are providing the OTP verification for user. By adding this process we will be able to control the security of the data. In this the user needs enter his/her email id and they need to enter the password. After filling in the mail and password the user will receive the 4 - digit OTP for the entered mail id. After the verification of the OTP from the server the user will be allowed to access the further services in the website. With the validation of the OTP we can control the unknown login activity.

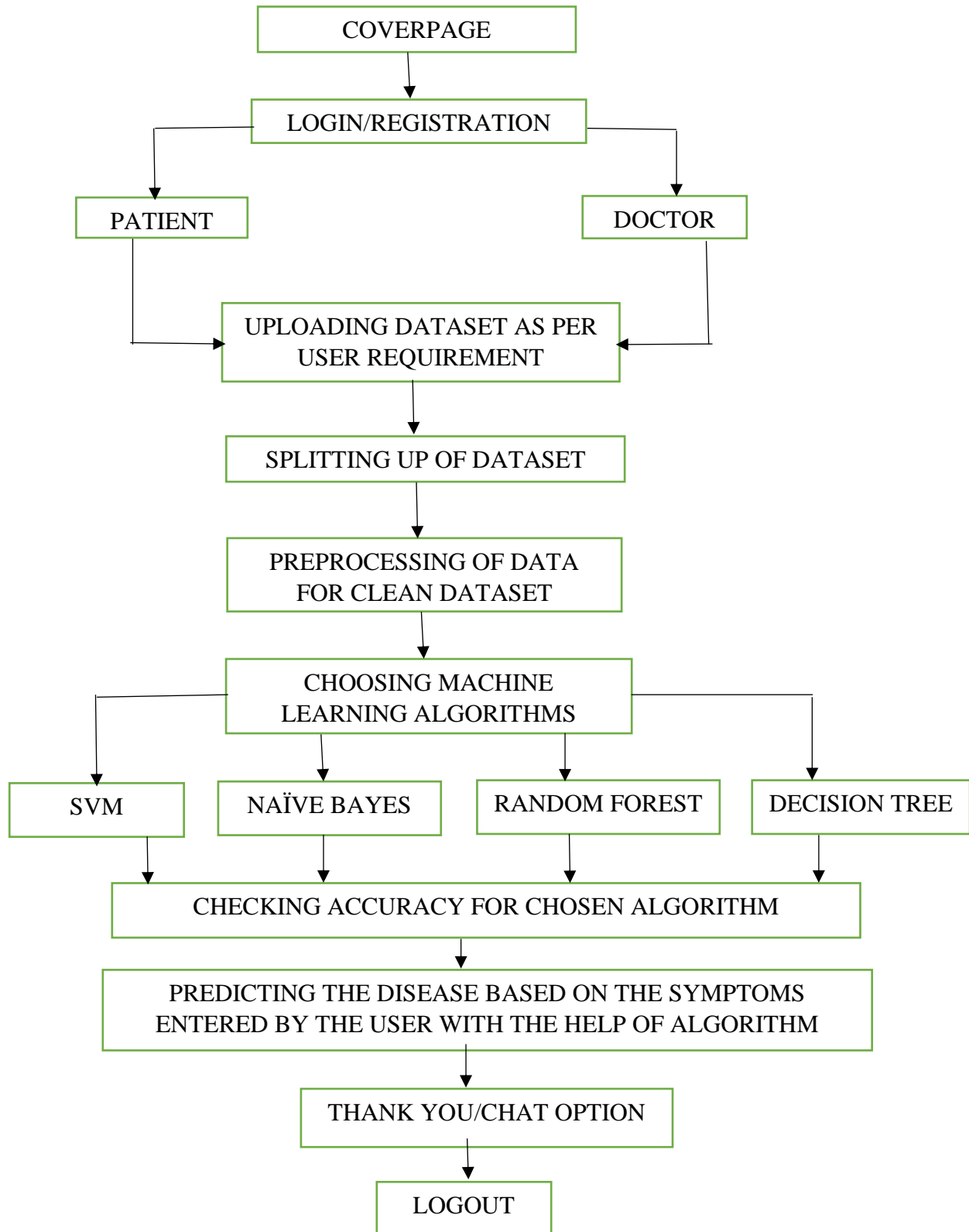
After the login the user will select the algorithm which they prefer and after selection process The user can upload their desired file for training the data and they can even select the size for splitting the data.

Once you are done with the completion of splitting the data you will be directed to the page where you need to select symptoms you are feeling or experiencing. Once they enter the data and submit the model will perform the prediction of the based on the input entered by the user.

## CHAPTER-6

### SYSTEM DESIGN

#### 6.1 FLOWCHART



## CHAPTER-7

### IMPLEMENTATION

#### 7.1 Divisions:

This project is modularized as the following:

1. Patient space
  2. Doctor space
  3. Load dataset
  4. Pre-process
  5. Model Training
- **Patient space:** This space consists of login, registration which is connected through SQL and data will be stored in the Database
  - **Doctor Space:** This space consists of login, registration which is connected through SQL and data will be stored in the Database
  - **Load Dataset:** Here we have to upload the datasets
  - **Preprocess:** The two datasets will be joined and will have an option to split the combined dataset
  - **Model Training:** Here we will be having the option to test the accuracy of the given algorithms for the dataset uploaded

## 7.2 Training and Testing:

The given below snapshot is of testing and training the data sets and combining those for checking greater accuracy and the images include dealing with missing values, checking data types, splitting the data for training and testing the model again, and Model training

Model training is done in four algorithms namely Gaussian Naïve Bayes, Random Forest, Decision Tree, and Support Vector Machine.

```
In [37]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

### Reading Training Data

```
In [38]: pd.set_option('display.max_columns', 500)
```

```
In [39]: TrainData=pd.read_csv(r'Training.csv')
TestData=pd.read_csv(r'Testing.csv')
```

FIGURE 7.2.1: READING TRAINING DATA

### Dealing with Missing values

```
In [42]: TrainData.shape, TestData.shape
```

```
Out[42]: ((4920, 133), (42, 133))
```

```
In [43]: TrainData.isnull().sum().sum()
```

```
Out[43]: 0
```

```
In [44]: TestData.isnull().sum().sum()
```

```
Out[44]: 0
```

Observation: There is no such null values

FIGURE 7.2.2: DEALING WITH MISSING VALUES



### Checking DataTypes

```
In [45]: TrainData.dtypes
Out[45]: itching                int64
skin rash                    int64
nodal skin eruptions         int64
continuous sneezing          int64
shivering                    int64
...
inflammatory nails           int64
blister                      int64
red sore around nose         int64
yellow crust ooze            int64
prognosis                    object
Length: 133, dtype: object
```

```
In [46]: TestData.dtypes
Out[46]: itching                int64
skin rash                    int64
nodal skin eruptions         int64
continuous sneezing          int64
shivering                    int64
...
inflammatory nails           int64
blister                      int64
red sore around nose         int64
yellow crust ooze            int64
prognosis                    object
Length: 133, dtype: object
```

FIGURE7.2.3: CHECKING DATA TYPES

### Splitting the data for training and testing model

```
In [64]: df.shape
```

```
Out[64]: (4962, 133)
```

```
In [65]: df.head(2)
```

```
Out[65]:
```

	itching	skin rash	nodal skin eruptions	continuous sneezing	shivering	chills	joint pain	stomach pain	acidity	ulcers on tongue	muscle wasting	vomiting	burning micturition	spotting urination	fatigue	weight gain	anxiety	haziness of vision
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

```
In [66]: col=(['itching','skin rash','continuous sneezing','joint pain','stomach pain','acidity',
              'ulcers on tongue','vomiting','burning micturition','spotting urination','fatigue','weight gain','anxiety',
              'restlessness','cough','high fever','breathlessness','dehydration','indigestion',
              'dark urine','nausea','back pain','constipation','yellowing of eyes','chest pain'])
```

```
In [67]: len(col)
```

```
Out[67]: 25
```

```
In [68]: from sklearn.model_selection import train_test_split
x = df.iloc[:, :-1]
y = df.iloc[:, -1]
```

FIGURE7.2.4: SPLITTING THE DATA (1)

```

In [71]: pca_x= pca.transform(x)

In [72]: pca_x= pd.DataFrame(pca_x, columns= col)
pca_x

```

0	-0.720407	-0.347159	-0.267203	-0.678152	0.502034	-0.453383	0.080398	-0.320001	-0.409643	0.288546	0.088062	-0.157589	0.484533	-0.1371
1	-0.795467	-0.282087	-0.330975	-0.458130	0.331851	-0.311991	-0.117324	-0.215825	-0.111919	0.137985	0.108707	-0.282943	0.280932	0.0131
2	-0.673029	-0.363211	-0.146401	-0.513322	0.110037	-0.205784	0.204020	-0.070778	-0.347601	0.186001	-0.022962	0.111392	0.412356	-0.0641
3	-0.700612	-0.336122	-0.254277	-0.645230	0.473463	-0.417578	0.073730	-0.292375	-0.372051	0.257763	0.078010	-0.137929	0.419687	-0.1191
4	-0.700612	-0.336122	-0.254277	-0.645230	0.473463	-0.417578	0.073730	-0.292375	-0.372051	0.257763	0.078010	-0.137929	0.419687	-0.1191
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4957	-0.823018	-0.287890	-0.356891	-0.462287	0.335414	-0.319678	-0.177469	-0.221038	-0.051172	0.114228	0.140356	-0.403442	0.257107	0.0951
4958	-0.820077	-0.330741	-0.249211	-0.332756	-0.123834	-0.047497	0.012040	0.088664	-0.033052	0.012955	-0.013123	-0.031351	0.362095	0.2921
4959	-0.669331	-0.425190	-0.477061	-0.506099	0.535077	-0.333683	-0.600894	-0.314083	0.672828	0.239067	0.324978	-1.018135	0.051814	-0.2081
4960	-0.601283	0.064497	-0.473850	-0.501986	0.478816	-0.450176	-0.540954	-0.054710	-0.383729	0.054886	0.645637	-0.142680	-0.115620	-0.2351
4961	-0.646691	-0.269863	-0.281817	-0.723960	0.733092	-0.534483	-0.107277	-0.376590	-0.273637	0.069156	0.254859	-0.193463	0.326171	-0.0701

4962 rows x 25 columns

```

In [73]: x_train, x_test, y_train, y_test=train_test_split(pca_x, y, random_state=42, test_size=0.3)

In [74]: print(f"X_TrainData: {x_train.shape}")
print(f"X_TestData: {x_test.shape}")
print(f"Y_TrainData: {y_train.shape}")
print(f"Y_TestData: {y_test.shape}")

X_TrainData: (3473, 25)
X_TestData: (1489, 25)
Y_TrainData: (3473,)
Y_TestData: (1489,)

```

FIGURE7.2.5: SPLITTING THE DATA (2)

```

Model Building

In [75]: from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

```

FIGURE7.2.6: MODEL BUILDING

## Gaussian Naive Bayes

```
In [76]: from sklearn.naive_bayes import GaussianNB  
nb = GaussianNB()
```

```
In [77]: nb.fit(x_train,y_train)  
pred1= nb.predict(x_test)  
print(accuracy_score(pred1,y_test))  
  
1.0
```

```
In [78]: print(classification_report(pred1,y_test))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	29
1	1.00	1.00	1.00	44
2	1.00	1.00	1.00	39
3	1.00	1.00	1.00	42
4	1.00	1.00	1.00	34
5	1.00	1.00	1.00	37
6	1.00	1.00	1.00	42
7	1.00	1.00	1.00	33
8	1.00	1.00	1.00	33
9	1.00	1.00	1.00	33
10	1.00	1.00	1.00	34
11	1.00	1.00	1.00	40
12	1.00	1.00	1.00	32
13	1.00	1.00	1.00	29
14	1.00	1.00	1.00	32
15	1.00	1.00	1.00	33
16	1.00	1.00	1.00	37

FIGURE 7.2.7: GAUSSIAN NAÏVE BAYES

## Random Forest

```
In [82]: from sklearn.ensemble import RandomForestClassifier  
rf = RandomForestClassifier()  
rf.fit(x_train,y_train)
```

```
Out[82]: RandomForestClassifier()
```

```
In [83]: pred=rf.predict(x_test)
```

```
In [84]: print(accuracy_score(pred,y_test))  
  
1.0
```

```
In [85]: print(classification_report(pred,y_test))
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	29
1	1.00	1.00	1.00	44
2	1.00	1.00	1.00	39
3	1.00	1.00	1.00	42
4	1.00	1.00	1.00	34
5	1.00	1.00	1.00	37
6	1.00	1.00	1.00	42

FIGURE 7.2.8: RANDOM FOREST

### Decision Tree

```
In [56]: from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)
```

```
Out[56]: DecisionTreeClassifier()
```

```
In [57]: pred=dt.predict(x_test)
```

```
In [58]: print(accuracy_score(pred,y_test))
print('=====')
```

```
print(classification_report(pred,y_test))

0.9993284083277367
=====
              precision    recall  f1-score   support

    0         1.00         1.00         1.00         29
    1         1.00         1.00         1.00         44
    2         1.00         1.00         1.00         39
    3         1.00         0.98         0.99         43
    4         1.00         1.00         1.00         34
    5         1.00         1.00         1.00         37
    6         1.00         1.00         1.00         42
    7         1.00         1.00         1.00         33
    8         1.00         1.00         1.00         33
    9         1.00         1.00         1.00         33
   10         1.00         1.00         1.00         34
   11         1.00         1.00         1.00         40
   12         1.00         1.00         1.00         32
   13         1.00         1.00         1.00         29
   14         1.00         1.00         1.00         32
   15         0.97         1.00         0.98         32
   16         1.00         1.00         1.00         37
   17         1.00         1.00         1.00         39
```

FIGURE 7.2.9: DECISION TREE

### Support Vector Machine

```
In [55]: from sklearn.svm import SVC
svm = SVC(kernel='linear')
```

```
In [59]: svm.fit(x_train,y_train)
pred=svm.predict(x_test)
```

```
In [60]: print(accuracy_score(pred,y_test))
print('=====')
```

```
print(classification_report(pred,y_test))

0.9979852249832102
=====
              precision    recall  f1-score   support

    0         1.00         1.00         1.00         29
    1         1.00         1.00         1.00         44
    2         1.00         0.93         0.96         42
    3         1.00         1.00         1.00         42
    4         1.00         1.00         1.00         34
    5         1.00         1.00         1.00         37
    6         1.00         1.00         1.00         42
    7         1.00         1.00         1.00         33
    8         1.00         1.00         1.00         33
    9         1.00         1.00         1.00         33
   10         1.00         1.00         1.00         34
   11         1.00         1.00         1.00         40
   12         1.00         1.00         1.00         32
   13         1.00         1.00         1.00         29
   14         1.00         1.00         1.00         32
   15         0.91         1.00         0.95         30
   16         1.00         1.00         1.00         37
   17         1.00         1.00         1.00         39
   18         1.00         1.00         1.00         37
   19         1.00         1.00         1.00         42
```

FIGURE 7.2.10: SUPPORT VECTOR MACHINE

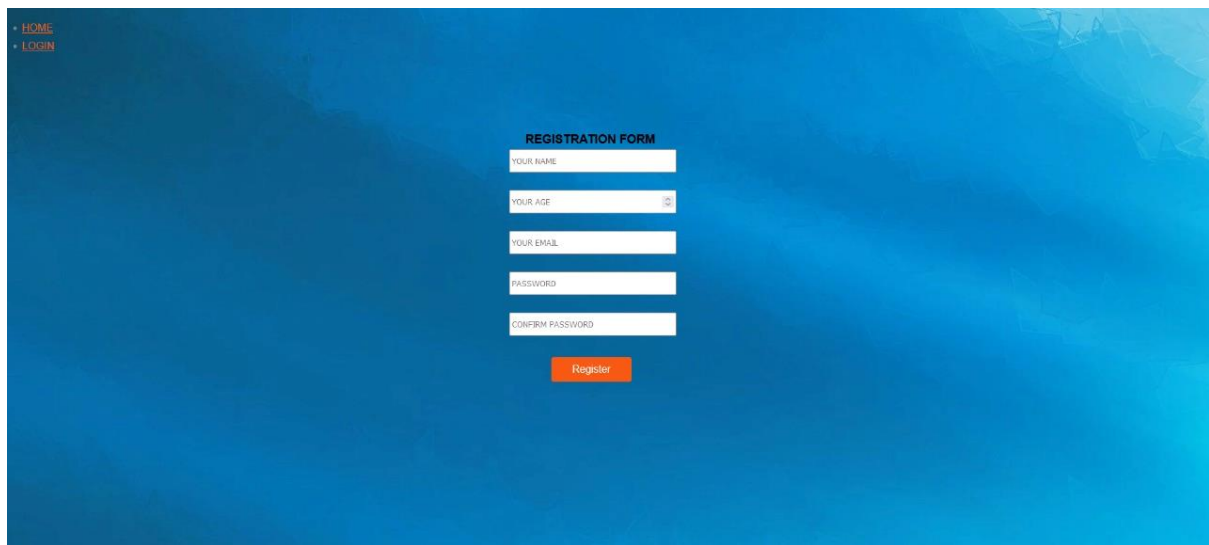
### 7.3 Website Implementation:



FIGURE7.3.1: HOME PAGE

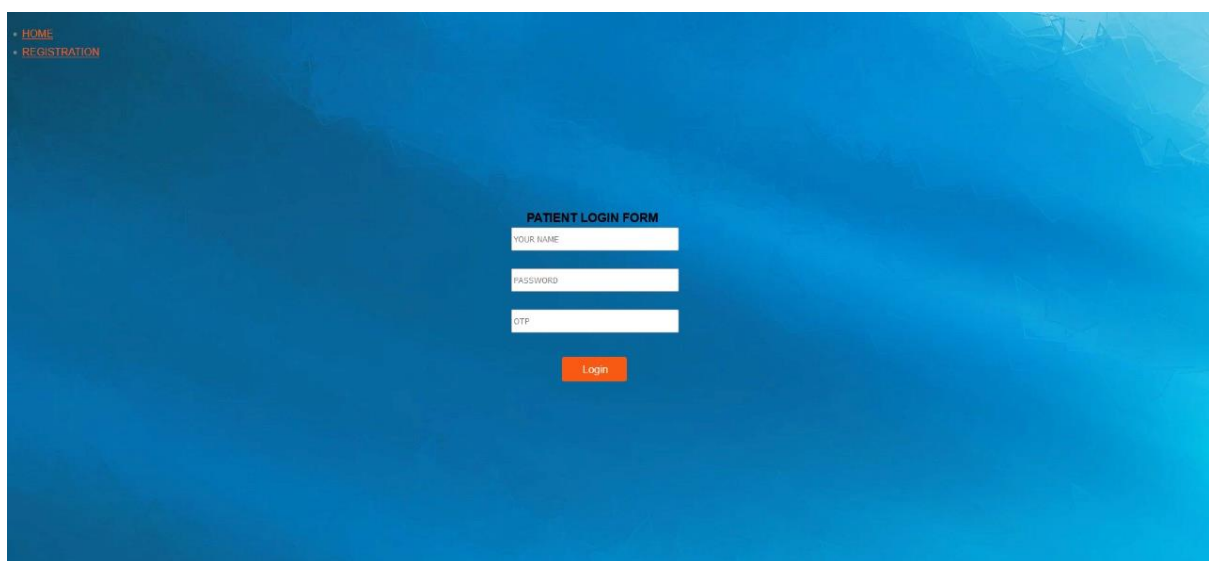


FIGURE7.3.2: DESIGNATION PAGE



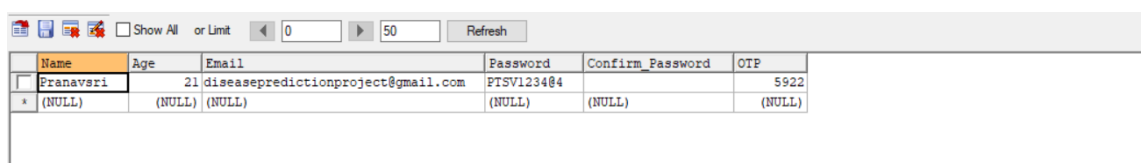
A screenshot of a web application's registration page. The background is a solid blue color. In the top-left corner, there are two links: 'HOME' and 'LOGIN'. Centered on the page is a 'REGISTRATION FORM' with five input fields: 'YOUR NAME', 'YOUR AGE', 'YOUR EMAIL', 'PASSWORD', and 'CONFIRM PASSWORD'. Below these fields is an orange 'Register' button.

FIGURE7.3.3: REGISTRATION PAGE



A screenshot of a web application's patient login page. The background is a solid blue color. In the top-left corner, there are two links: 'HOME' and 'REGISTRATION'. Centered on the page is a 'PATIENT LOGIN FORM' with three input fields: 'YOUR NAME', 'PASSWORD', and 'OTP'. Below these fields is an orange 'Login' button.

FIGURE7.3.4: LOGIN PAGE



A screenshot of a data table displayed in a web application. The table has seven columns: Name, Age, Email, Password, Confirm\_Password, and OTP. The first row shows a user named 'Pranavari' with age 21, email 'diseasepredictionproject@gmail.com', password 'PTSV1234@4', and OTP '5922'. The second row shows a user with all fields as '(NULL)'. Above the table, there is a toolbar with icons for various actions, a 'Show All' checkbox, a 'Limit' dropdown set to '0', a 'Refresh' button, and a '50' value.

	Name	Age	Email	Password	Confirm_Password	OTP
	Pranavari	21	diseasepredictionproject@gmail.com	PTSV1234@4		5922
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

FIGURE7.3.5: DATA TABLE



FIGURE7.3.6: INDEX PAGE

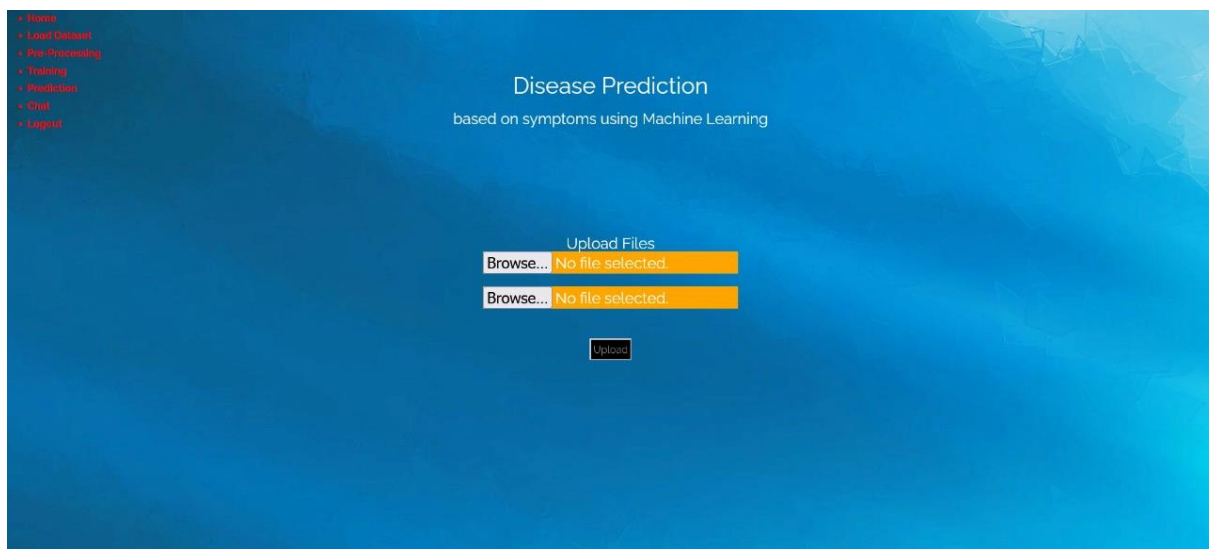


FIGURE7.3.7: DATASET UPLOAD ACTION PAGE

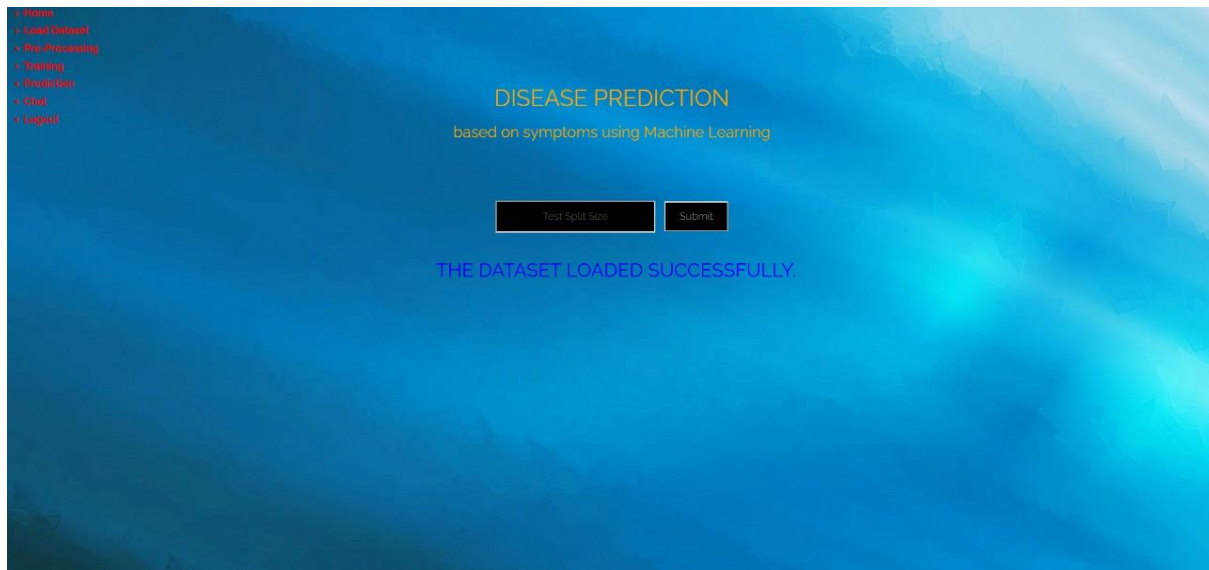


FIGURE7.3.8: SPLITTING DATASET PAGE

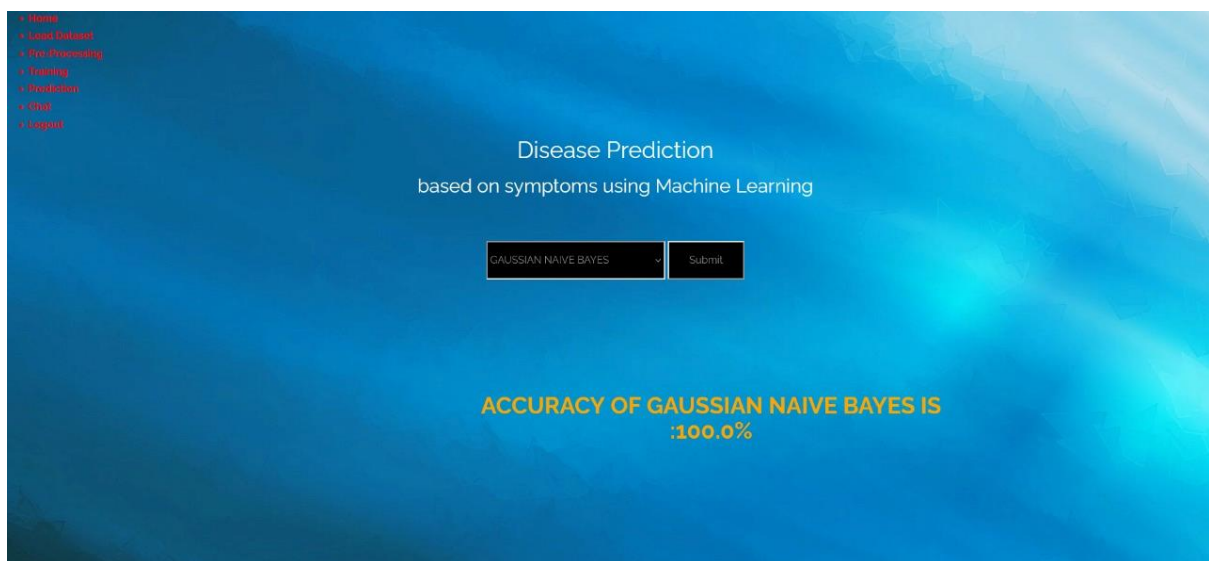


FIGURE7.3.9: ALGORITHM PAGE



- Home
- Load Dataset
- Pre-Processing
- Training
- Prediction
- Chat
- Logout

## DISEASE PREDICTION

With the help of Machine Learning

PREDICTED DISEASE IS:

**ptic ulcer disease**

ITCHING_RELATED :	<input type="text"/>	ANXIETY_RELATED :	<input type="text"/>
SKIN_RASH_RELATED :	<input type="text"/>	RESTLESSNESS_RELATED :	<input type="text"/>
CONTINUOUS_SNEEZING_RELATED :	<input type="text"/>	COUGH_RELATED :	<input type="text"/>
JOINT_PAIN_RELATED :	<input type="text"/>	HIGH_FEVER_RELATED :	<input type="text"/>
STOMACH_PAIN_RELATED :	<input type="text"/>	BREATHLESSNESS_RELATED :	<input type="text"/>
ACIDITY_RELATED :	<input type="text"/>	DEHYDRATION_RELATED :	<input type="text"/>
ULCER_ON_TONGUE_RELATED :	<input type="text"/>	INDIGESTION_RELATED :	<input type="text"/>
VOMITING_RELATED :	<input type="text"/>	DARK URINE_RELATED :	<input type="text"/>
BURNING_MICTURITION_RELATED :	<input type="text"/>	NAUSEA_RELATED :	<input type="text"/>
SPOTTING_URINATION_RELATED :	<input type="text"/>	BACK_PAIN_RELATED :	<input type="text"/>
FATIGUE_RELATED :	<input type="text"/>	CONSTIPATION_RELATED :	<input type="text"/>
WEIGHT_GAIN_RELATED :	<input type="text"/>	YELLOWING_EYES_RELATED :	<input type="text"/>
		CHEST_PAIN_RELATED :	<input type="text"/>

Predict

FIGURE7.3.10: DISEASE PREDICTION PAGE

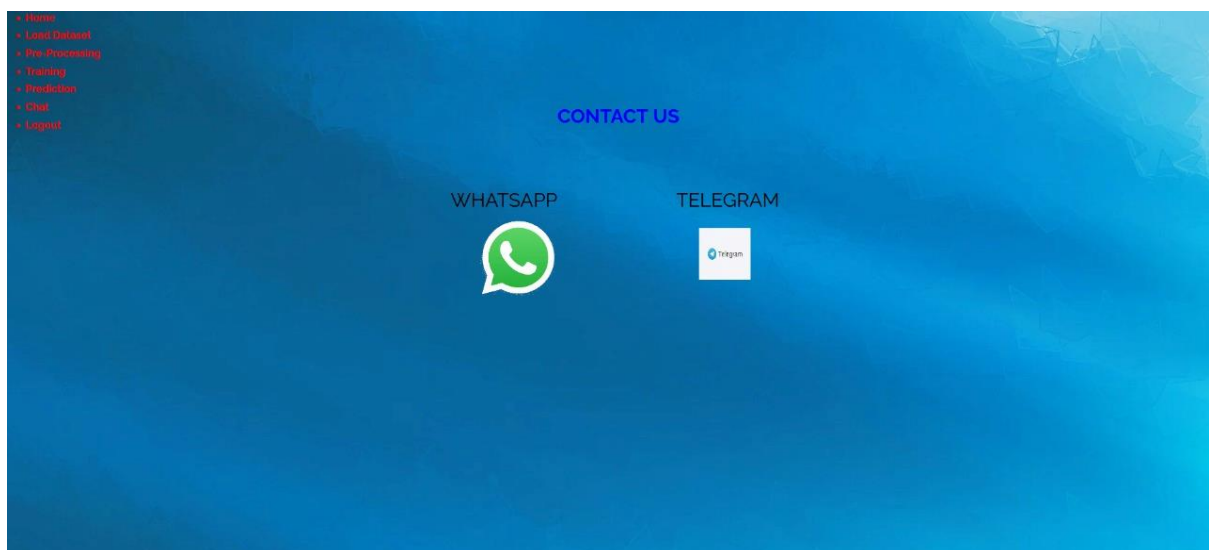


FIGURE7.3.11: CONTACT US PAGE

## CHAPTER-8

### Testing

A Definition Software testing is the process of finding errors in the developed product. It also checks whether the real outcomes can match expected results, as well as aids in the identification of defects, missing requirements, or gaps.

Testing is the penultimate step before the launch of the product to the market. It includes examination, analysis, observation, and evaluation of different aspects of a product.

Professional software testers use a combination of manual testing with automated tools. After conducting tests, the testers report the results to the development team. The end goal is to deliver a quality product to the customer, which is why software testing is so important.

### Importance of Software Testing

It's common for many startups to skip testing. They might say that their budget is the reason why they overlook such an important step. They think it would lead to no major consequences. But to make a strong and positive first impression, it needs to be top-notch. And for that, testing the product for bugs is a must.

To really understand why software testing is important, we need to correlate it with real world examples, which has caused serious issues in the past, a few examples includes;

- In October 2014, Flipkart an e-commerce in India company had an offer called the “Big Billion Sale.” When it was launched it had a lot of traffic and as a result, its website couldn't handle the enormous load of traffic leading to the website downtime, cancellation of orders etc. The reputation of the organization was badly impacted by this issue.
- In 2015, the Royal Bank of Scotland, due to a bug, couldn't process about 600,000 payments. Because of this, they were fined 66 million pounds
- Yahoo in September 2016, had a major data breach where 500 million users' credentials got compromised.
- Recently, Okta, an American authentication firm, had a digital breach due to a software bug that may have affected their user's details. This has also affected the reputation of Okta.

Disease prediction based on symptoms using machine learning algorithms was tested to check whether the system satisfied user needs and expectations. Four testing criteria namely: unit testing, integration testing, system testing and user acceptance testing will be used for checking the effectiveness of the project. The testing strategies were done as follows:

➤ **Unit Testing:**

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

- It is correlated with functional correctness of the independent modules. Unit Testing is defined as a type of software testing where individual components of a software are tested. Unit Testing of software product is carried out during the development of an application. An individual component may be either an individual function or a procedure.
- Unit Tests, when integrated with build gives the quality of the build as well. Black Box Testing - Using which the user interface, input and output are tested.
- White Box Testing - used to test each one of those functions behaviour is tested.

- **Objective of Unit Testing:**

- The objective of Unit Testing are:
  - 1 To isolate a section of code.
  - 2 To verify the correctness of code.
  - 3 To test every function and procedure.
  - 4 To fix bug early in development cycle and to save costs.
  - 5 To help the developers to understand the code base and enable them to make changes quickly.
  - 6 To help for code reuse.

Several units of Disease prediction based on symptoms using machine learning algorithms were tested against the input into the system to validate and verify their functionality.

- Patient page
- Doctor page

Patient page:

- In this page the patient should enter their details and we get an error if the email or password is wrong.

After sign up it checks for the correct otp ,if the entered otp is wrong we cant proceed further.

Doctor page:

- In this page doctors are supposed to enter their password and Otp , if the entered otp is wrong we cant proceed further.

### ➤ **Integration Testing:**

- Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

- Entailed testing of all modules to check on quality assurance, verification and validation or reliability. The units that were tested as in unit testing above were tested as a whole to point to their performance.

➤ **Main Objectives of integration testing are:**

- Building confidence in the quality of the system for this automated regression test could be used.
  - Finding defects (which maybe in the interfaces themselves or within the components or systems).
  - Reducing risk.
  - Verifying whether the functional and non-functional behaviors of the system are as designed, specified and working as per requirement.
  - Preventing defects from escaping to higher test levels.
- 
- Primarily there are two different types of integration Testing:
    - Component integration testing
    - System integration testing

➤ **System Testing:**

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system. In other words, a computer system consists of a group of software to perform the various tasks, but only software cannot perform the task; for that software must be interfaced with compatible hardware. System testing is a series of different type of tests with the purpose to exercise and examine the full working of an integrated software computer system against requirements.

The whole system functionality was tested. The researcher acknowledges that all the modules were tried on various devices using various inputs and system showed consistency in giving the outputs as it had been required to do.

In the course of system testing, a QA team can notice the following defects:

- Misuse of system resources
- Unexpected combinations of user-level data
- Incompatibility with the environment
- Unexpected use cases
- Missing/incorrect functionality
- The inconvenience of use, etc.

System testing is the inspection of a fully integrated system. The purpose of this checkup is to verify the system's conformance with both functional and non-functional requirements.

In the course of system testing, a QA team can notice the following defects:

- Misuse of system resources
- Unexpected combinations of user-level data
- Incompatibility with the environment
- Unexpected use cases
- Missing/incorrect functionality
- The inconvenience of use, etc.

System testing uses the black-box method. It means that a QA team doesn't know how the backend works. They aren't aware of the internal structure of software and check the features using frontend – the 'facade' of the program. This allows modeling user behavior closely since users interact with software in the same way.

### **Objectives of System Testing**

The goal of system testing is to minimize the risks associated with the behavior of the system in a particular environment. For this, testers use the environment as close as possible to the one where a product will be installed after the release.

The objectives are some smaller steps that allow achieving the goal. In system testing, there are several milestones that make the release of a flawlessly functioning system (read: software without critical bugs on production) possible. So the primary objectives are:

- Reducing risks, for bug-free components don't always perform well as a system.
- Preventing as many defects and critical bugs as possible by careful examination.
- Verifying the conformance of design, features, and performance with the specifications stated in the product requirements.
- Validating the confidence in the system as a whole before moving to the final stage – acceptance testing that takes place right before users get access to a product.

### ➤ **User Acceptance Testing:**

➤ The "User Acceptance Testing" phase is crucial to ensure that you and your end-user are both satisfied with the final solution. It is essential because:

- It helps confirm that the product meets the specific work requirements.
- It helps identify a problem that might have been missed or overlooked by you or your team.
- It tells if the product is actually ready to be launched into the market.
- It helps you know if you will have any problems with the solution later when launched.
- It helps identify any additional work to complete the project.

It involves a series of specific tests that helped to indicate whether this project meets the users needs and expectations or not. The testing was to continue even after software release.

- Acceptance testing is a formal testing conducted to determine whether a system satisfies its acceptance criteria – the criteria the system must satisfy to be accepted by the customer.
- It helps the customer to determine whether or not to accept the system.
- The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.
- Acceptance testing is performed after System Testing and before making the system available for actual use.
- There are various forms of Acceptance Testing :
  1. User Acceptance Testing
  2. Business Acceptance Testing
  3. Alpha Testing
  4. Beta Testing

### **Objectives of Acceptance Testing :**

Following are the three major objectives of Acceptance Testing :

- Confirm that the system meets the agreed-upon criteria.
- Identify and resolve discrepancies, if there are any.
- Determine the readiness of the system for cut-over to live operations. The final acceptance of a system for deployment is conditioned upon the outcome of the acceptance testing. The acceptance test team produces an acceptance test report which outlines the acceptance conditions.



## **CHAPTER-9**

### **Conclusion:**

Machine learning is not a new technique when it comes to predicting the chances of contracting illnesses, but it is becoming more diversified and accurate as time goes on. However, our prediction system which is more user-friendly gives accurate results and perform well on all the algorithms. The testing accuracy was approximately 99% again it is based on the what percent we are splitting the data for testing. This work has more future scope as we can work on many other datasets we can upload, pre-process and **split** the data with slight changes in the frontend. The work is mainly focused on Doctors/Hospital Management side. Future development includes both doctors and patient focused website and also advanced frontend design.

## Appendix-A

The reason we decided that we have to develop the website backend using “Flask” is that it is the best web application framework available that too the language used is the python which is very comfortable for writing and designing.



**A sneak-peak in our flask code:**

```
app.route('/logout')
def logout():
    return redirect(url_for('index'))

if __name__ == "__main__":
    app.run(debug=True, port=8000)
```

## Appendix-B

We have tried other algorithm but our fixed agenda is accuracy to give the best results for both sides of the health sector so we haven't included those in the project. The unused algorithm we tried is:

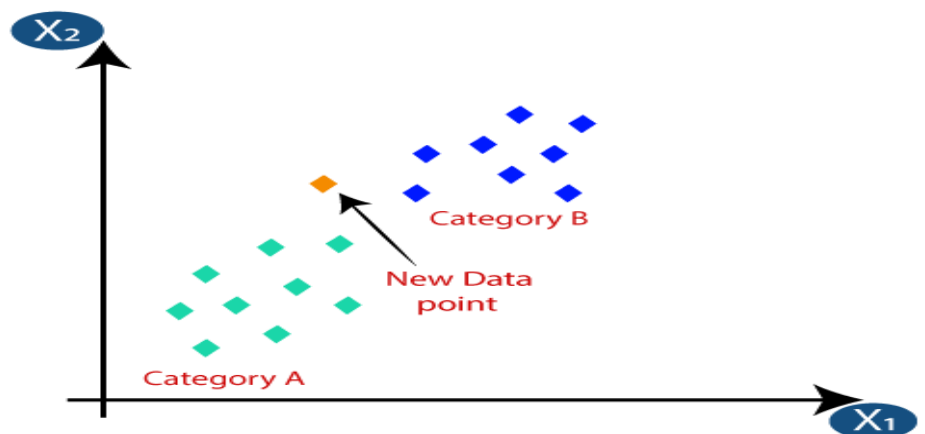
- **KNN:**

Because KNN algorithms have very less space management it can't handle the larger datasets

How does KNN work?

1. Selecting the number **K** of the neighbors
2. Calculating the Euclidean distance of **K number of neighbors**
3. Taking the **K** nearest neighbors as per the calculated Euclidean distance.
4. Among these **k** neighbors, count the number of data points in each category.
5. Assigning the new data points to that category for which the number of the neighbor is maximum.

**EX:**



**Advantages:**

- It is simple to implement.
- It is robust to the noisy training data

**Disadvantages:**

- Always needs to determine the value of  $K$  which may be complex sometimes.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

## References:

- [1].Decision Tree algorithm study reference is done by using the greek platform <https://www.geeksforgeeks.org/decision-tree/> and also we referred javapoint <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
- [2].Flask module references link of websites and the website we referred is <https://flask.palletsprojects.com/en/2.1.x/> Flask web development by Miguel Grinberg installation and setup Chapter 1 Use of templates in framework Chapter3.([https://coddyschool.com/upload/Flask\\_Web\\_Development\\_Developing.pdf](https://coddyschool.com/upload/Flask_Web_Development_Developing.pdf) )
- [3].Gaurav Shilimkar, Shivam Pisal (2021) ‘Disease Prediction Using Machine Learning’
- [4].Manasvi Srivastava, Vikas Yadav, Swati Singh (2020) ‘Implementation of Web Application for Disease Prediction Using AI’
- [5].MIN CHEN, (Senior Member, IEEE), YIXUE HAO, KAI HWANG, (Life Fellow, IEEE), LU WANG, AND LIN WANG (2017) ‘Disease Prediction by Machine Learning Over Big Data from Healthcare Communities’
- [6].Mysql.connector is been referred with the help of W3schools website [https://www.w3schools.com/python/python\\_mysql\\_getstarted.asp](https://www.w3schools.com/python/python_mysql_getstarted.asp)

- [7]. Naive Bayes algorithm <https://www.geeksforgeeks.org/ml-naive-bayes-scratch-implementation-using-python/> and <https://www.geeksforgeeks.org/naive-bayes-classifiers/>
- [8]. OS is background research work and understanding the concept <https://www.tutorialsteacher.com/python/os-module#:~:text=The%20OS%20module%20in%20Python,with%20the%20underlying%20operating%20system.>
- [9]. OTP verification the websites we referred in implementing this in project is <https://thecleverprogrammer.com/2021/04/14/otp-verification-using-python/>
- [10]. Random Forest algorithm <https://www.geeksforgeeks.org/random-forest-regression-in-python/>
- [11]. Random function use and its meaning is learnt with the help of w3school [https://www.w3schools.com/python/python\\_mysql\\_getstarted.asp](https://www.w3schools.com/python/python_mysql_getstarted.asp)
- [12]. Rudra A. Godse, Smita S. Gunjal, Karan A. Jagtap, Neha S. Mahamuni, Prof. Suchita Wankhade (2019) 'Multiple Disease Prediction Using Different Machine Learning Algorithms Comparatively'
- [13]. SVM algorithm references <https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8>
- [14]. Varinder Garg, Harish Kumar, Surinder Rana, Bikramjeet Singh Kalsi, Siddhant Mukherjee (2020) 'Machine learning based disease prediction website using symptoms of a patient'