## SDL PROJECT

# Disease Predictor


Group Members:

| | |
|---|---|
| Samidha Bharle | - TECOA111 |
| Priti Giramkar. | - TECOA133 |
| Sheetal Singh | - TECOA166 |
| Priti Vasekar | - TECOA169 |

Guided by : Prof. Alka Londhe.

# OUTLINE:

1. Problem Statement
2. Introduction
3. Scope
4. Objective
5. System Architecture
6. Hardware/Software Specification
7. Datasource

# PROBLEM STATEMENT:

**Disease Prediction** based on symptoms.

1. A disease predictor based on the symptoms entered by the user.
2. Machine Learning algorithms used to provide a computer aided prediction on the inputs (symptoms).

# INTRODUCTION:

- Challenges faced by many people are looking online for health information regarding diseases, diagnoses and different treatments.
- This system allows user to share their symptoms and issues. It then processes user's symptoms to check for various illnesses that could be associated with it.
- In our project we have tried accurately predict a disease by looking at the symptoms of the patient.

# SCOPE :

- We have designed an interactive interface to facilitate interaction with the system.
- It is a recommendation system made for doctors and medicine.
- This project aims to provide a web platform to predict the occurrences of disease on the basis of various symptoms
- The user can select various symptoms and can find the diseases with their probabilistic figures.

# OBJECTIVE:

- We are developing a project predicting the disease. The objective of Project is to effectively manage the prediction of disease with given symptoms.

- The main focus is on to use machine learning in healthcare.

- Predictive analysis with the help of efficient multiple machine learning algorithms.

# SYSTEM ARCHITECTURE:

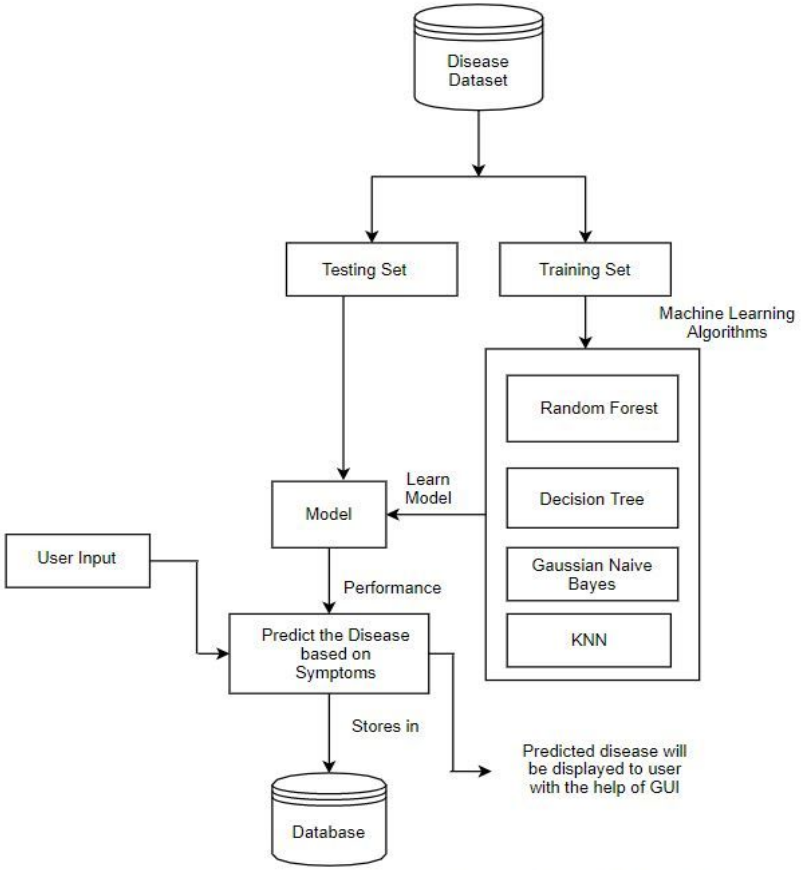
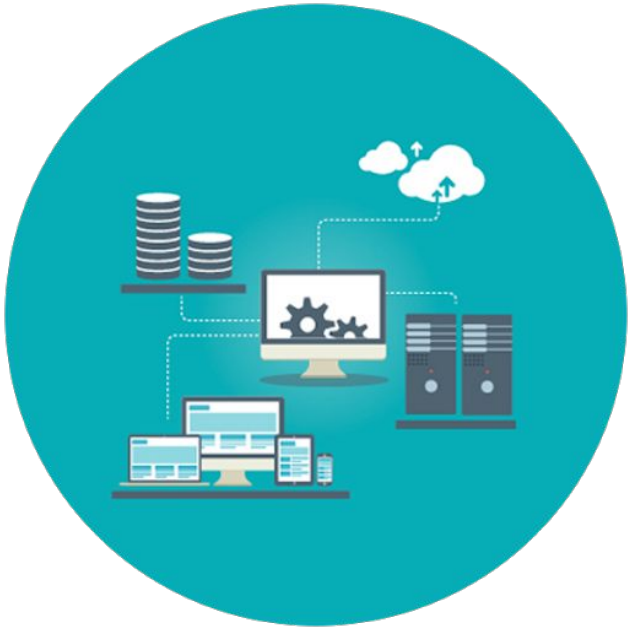



Fig:Architecture Diagram for Disease Predictor Project

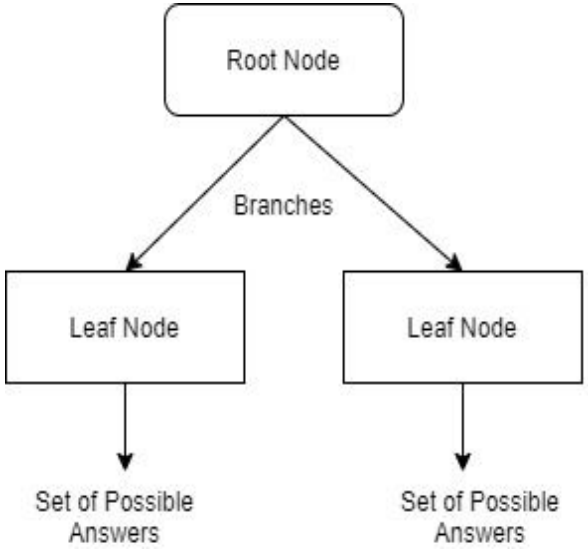


# ALGORITHMS USED

- Decision tree
- Random forest tree
- Gaussian Naïve Bayes
- KNN

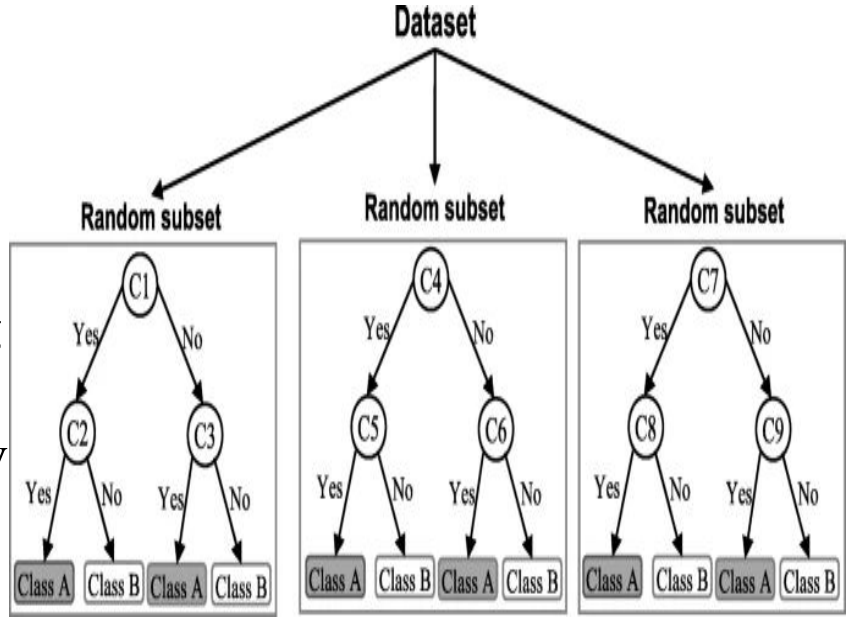
# 1. Decision Tree:

- Decision trees are used to approximate a discrete valued function.
- The tree does this by reducing all values to individual nodes.
- The Decision tree has been very useful in the various fields like medical diagnosis.
- Accuracy:



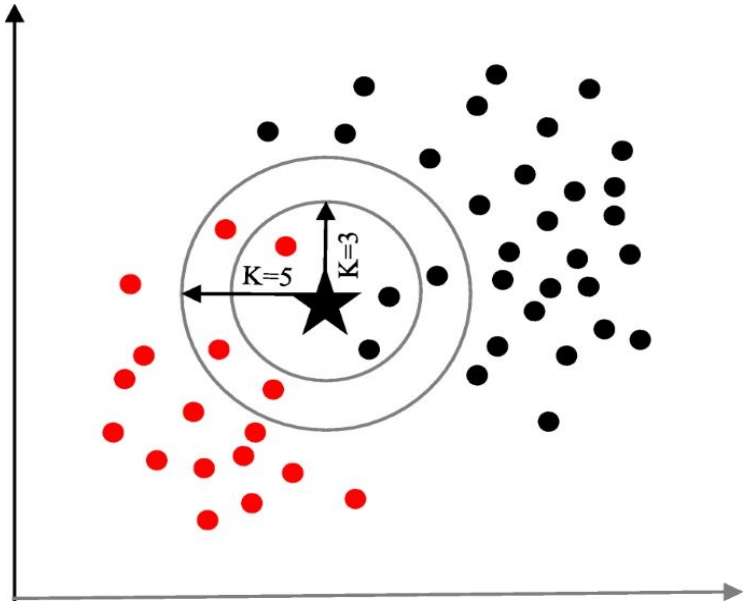
# 2. Random Forest:

- Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.
- Random forest adds additional randomness to the model, while growing the trees.
- Random forest results in a wide diversity that generally results in a better model.
- Accuracy:



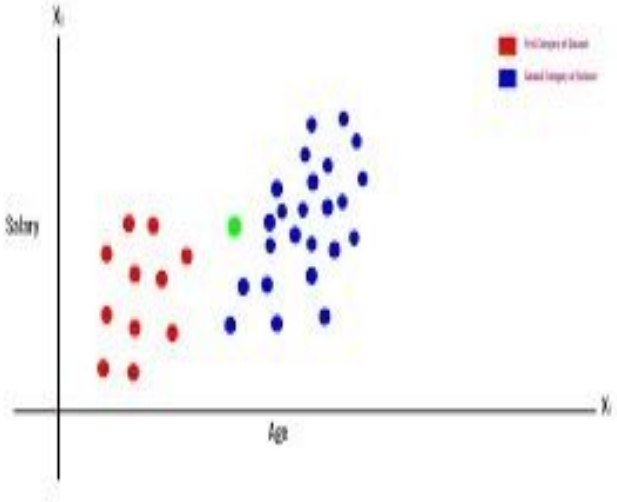
# 3. K Nearest Neighbour

- K Nearest Neighbour is a supervised learning algorithm. It is a basic yet essential algorithm.
- It finds extensive use in pattern finding and data mining.
- It works by finding a pattern in data which links data to results and it improves upon the pattern recognition with every iteration.
- Accuracy:



# 4. Naïve Bayes

- Naïve Bayes algorithm is a family of algorithms based on naïve bayes theorem.
- They share a common principle that is every pair of prediction is independent of each other.
- It also makes an assumption that features make an independent and equal contribution to the prediction.
- Accuracy:




$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

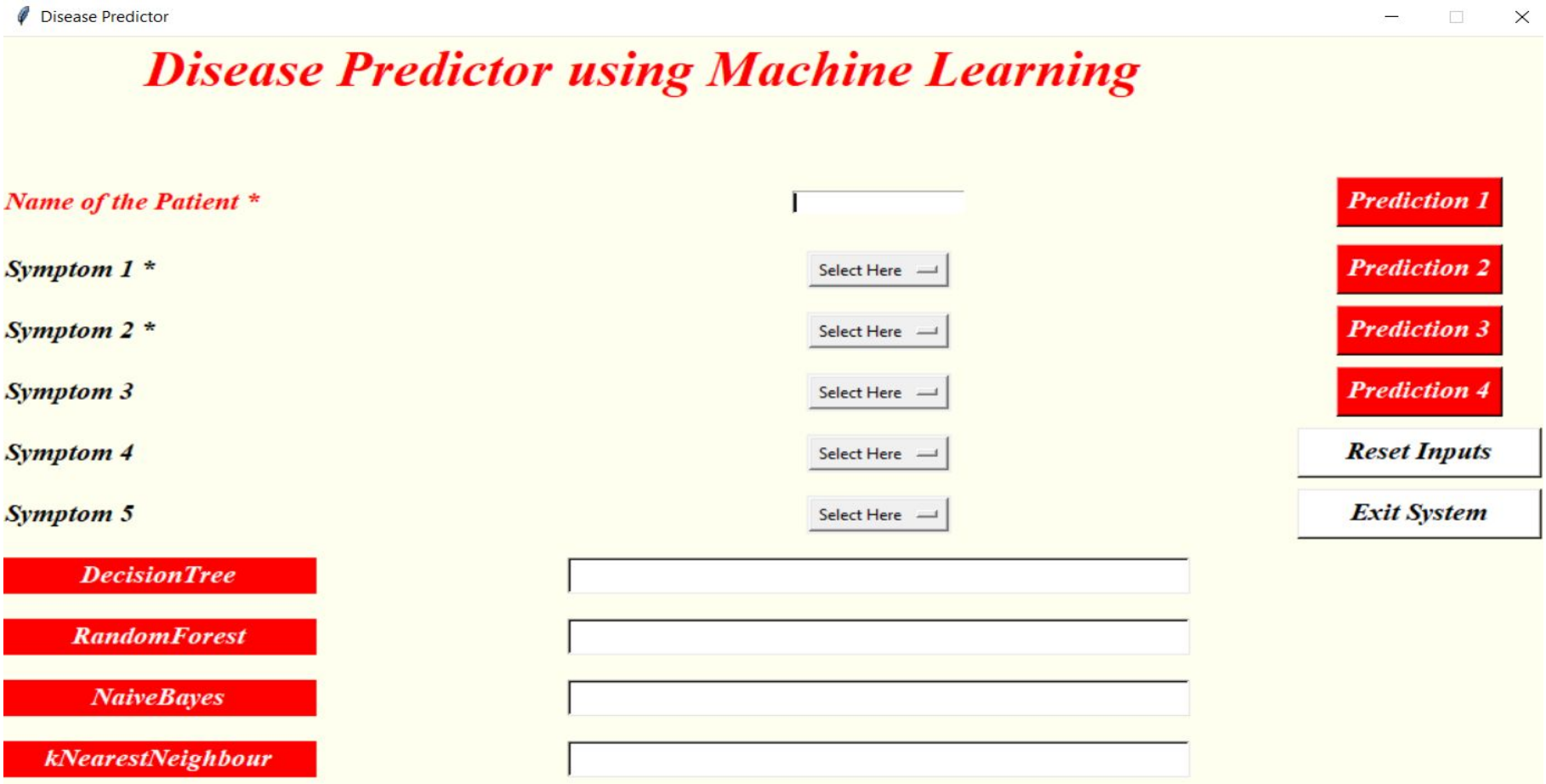
# HARDWARE AND SOFTWARE SPECIFICATIONS :

Machine Learning Algorithms

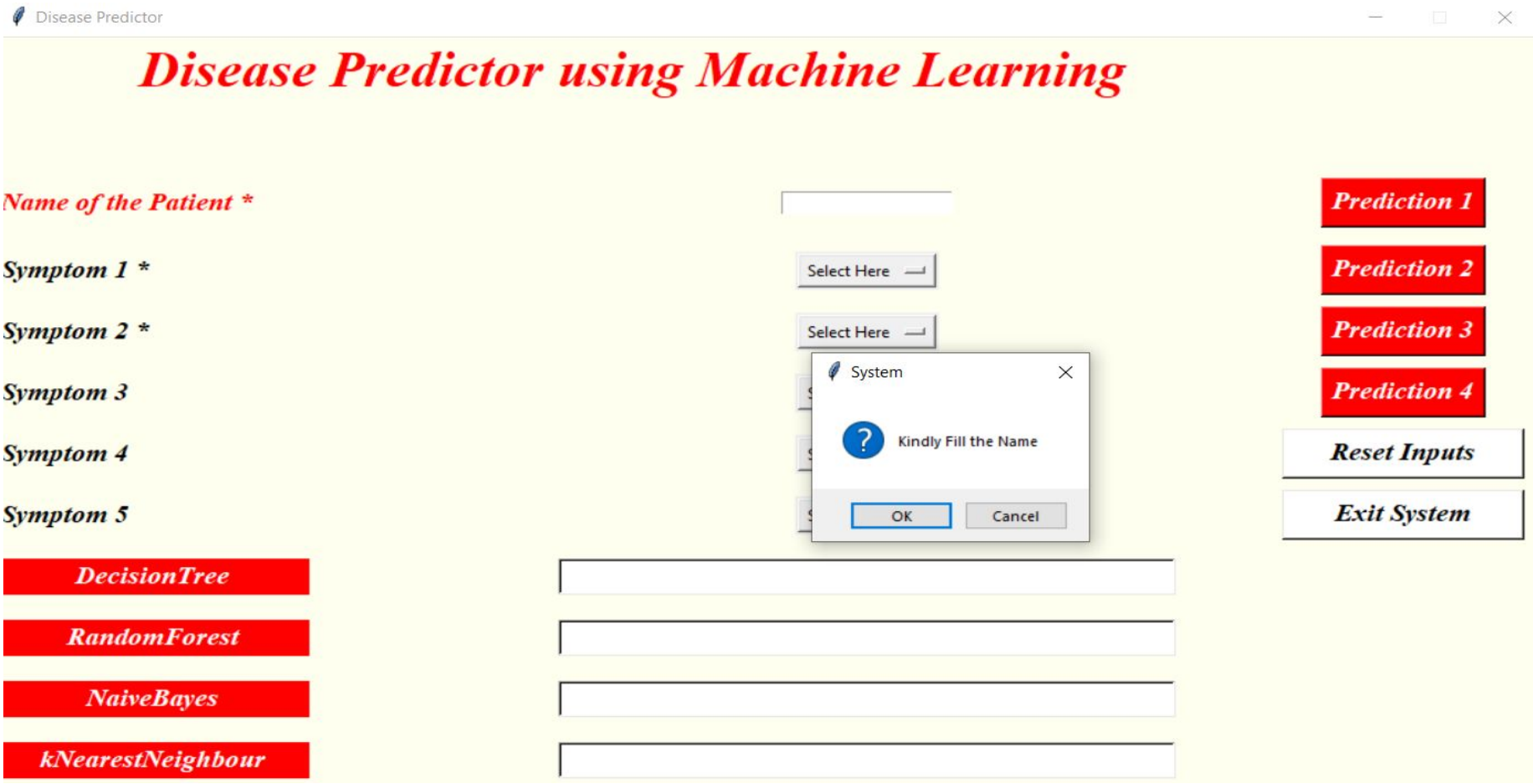

For Front-End Development: Python 3

For writing code: Jupyter Notebook

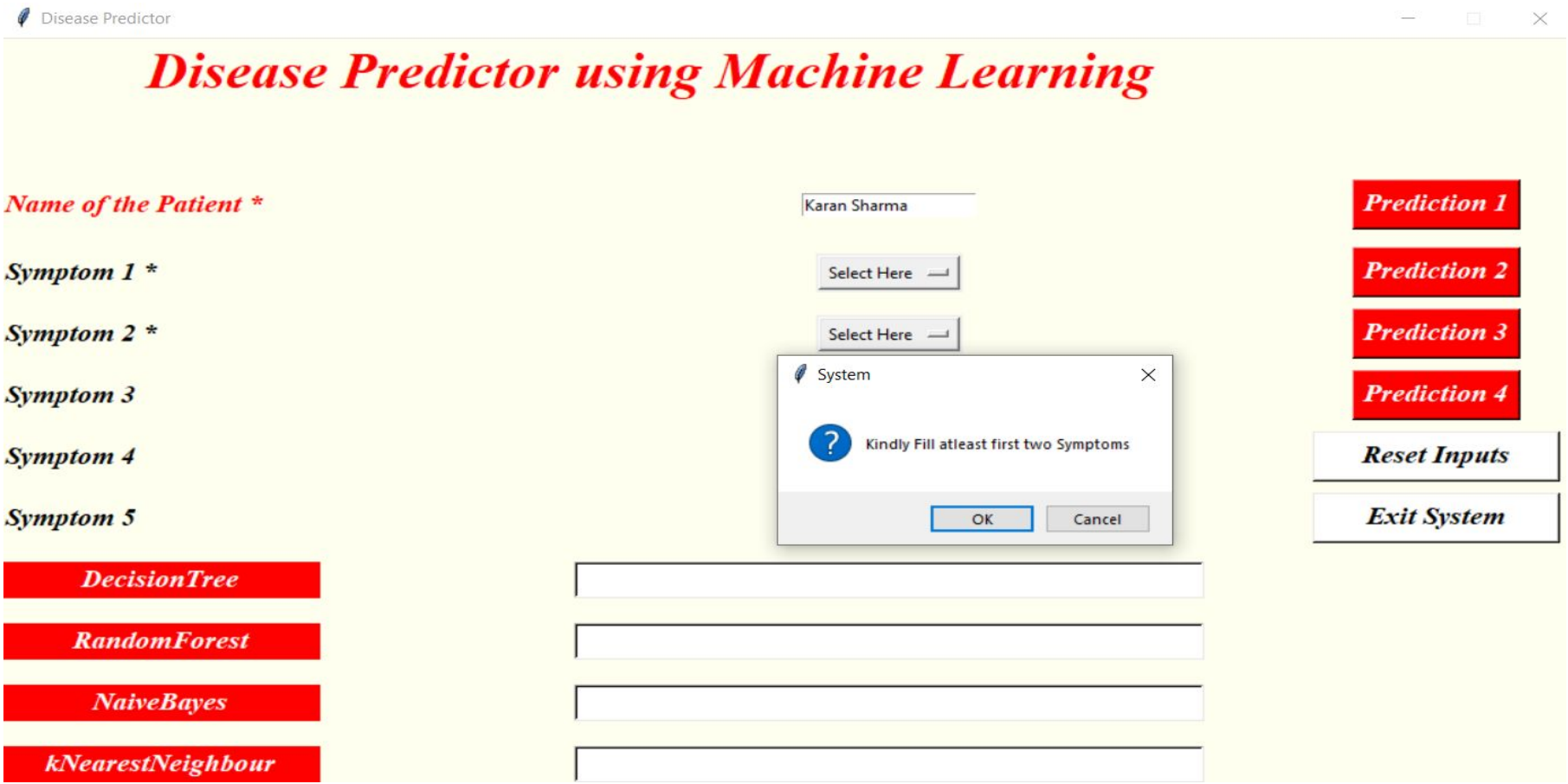
# IMPLEMENTATION STATUS :



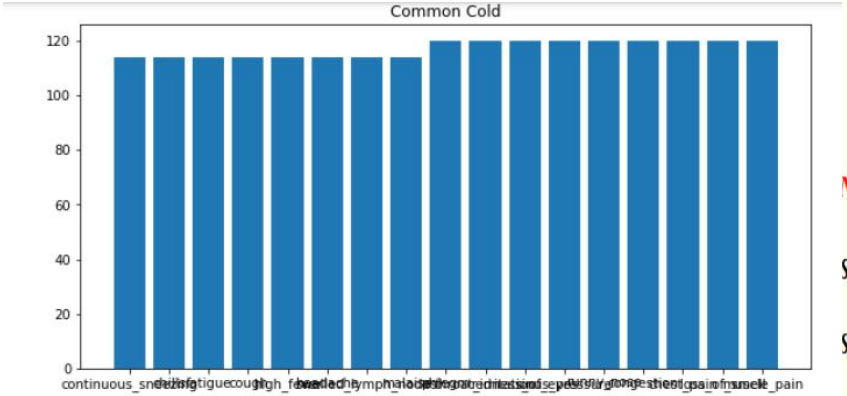
If user tries to run the gui without entering the name, then System will prompt the following message.



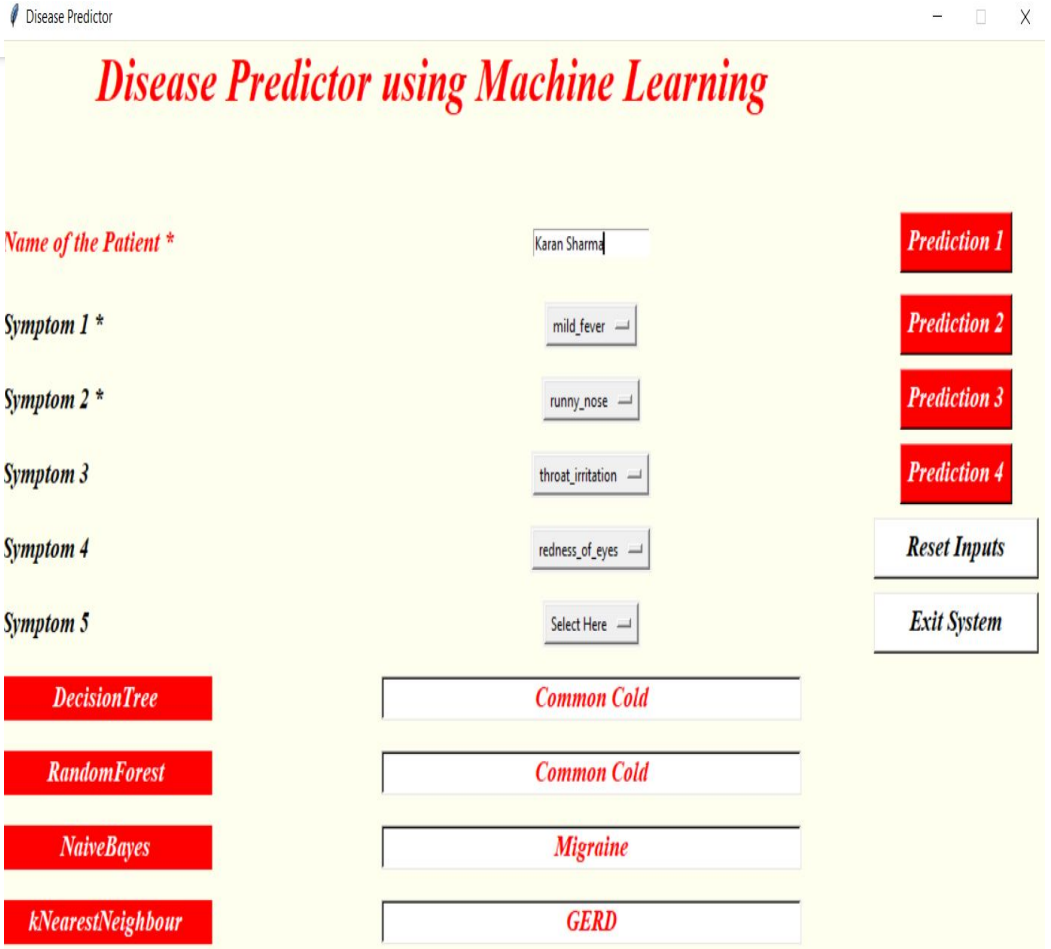
After filling the name, user have to fill five symptoms and out of which first two are compulsory. If user will not select atleast two symptoms, then following message will be prompt from the system.



The Barplot for the symptoms which are given by the user as input

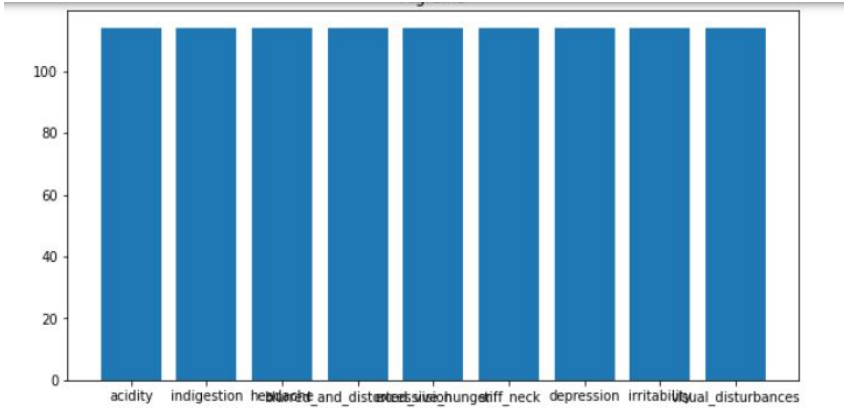



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

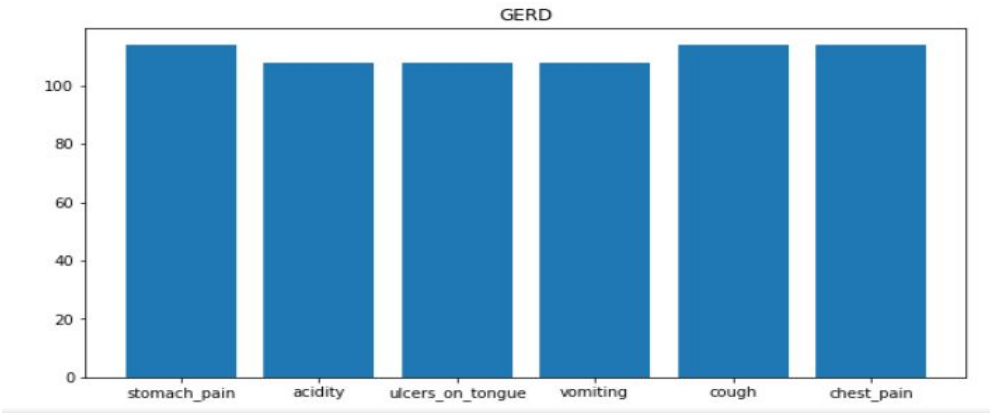


Accuracy of predicting the disease is printed using accuracy_score and confusion matrix is created using confusion_matrix which are imported from sklearn.metrices.

Graph for RandomForest



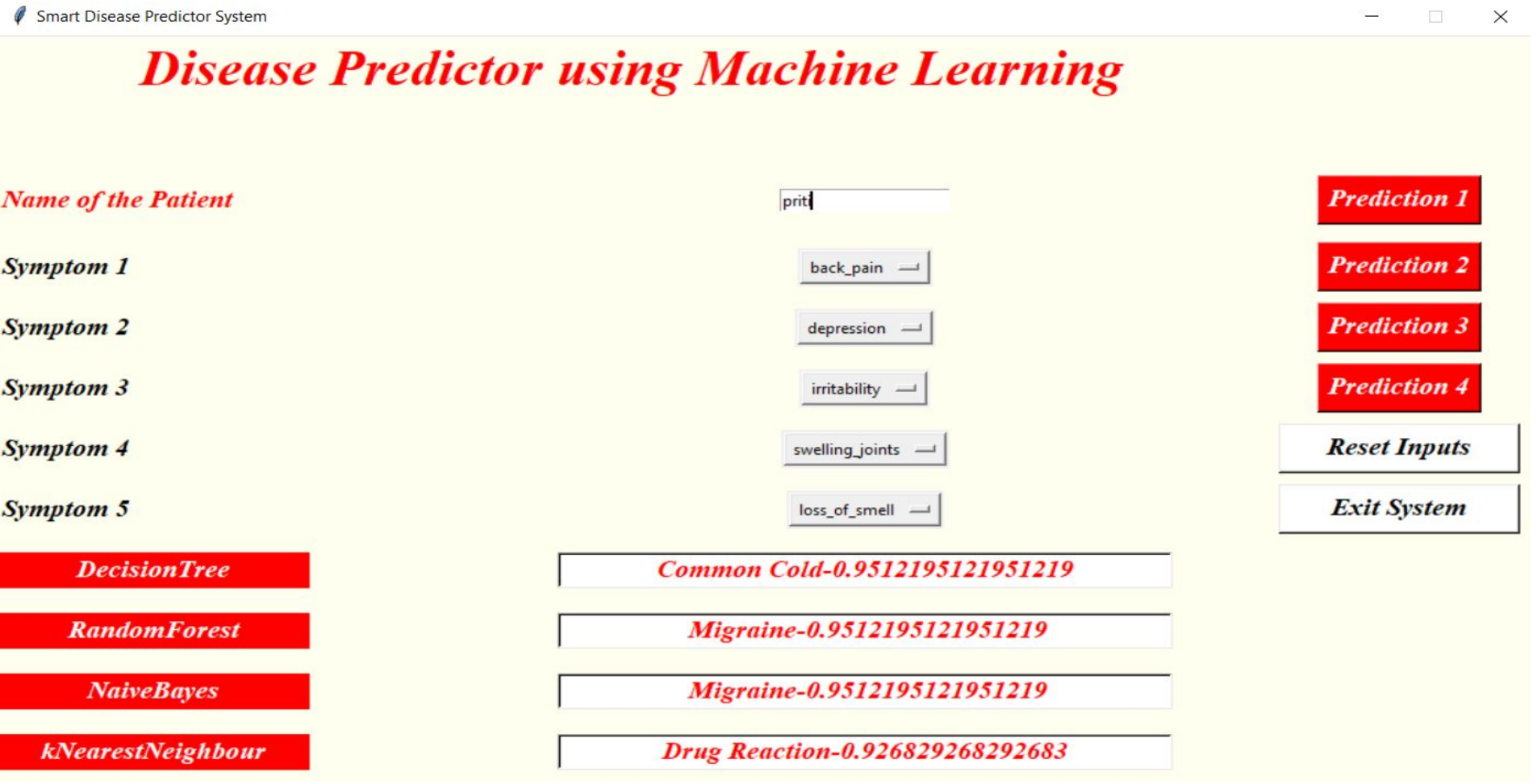

Graph for KNN



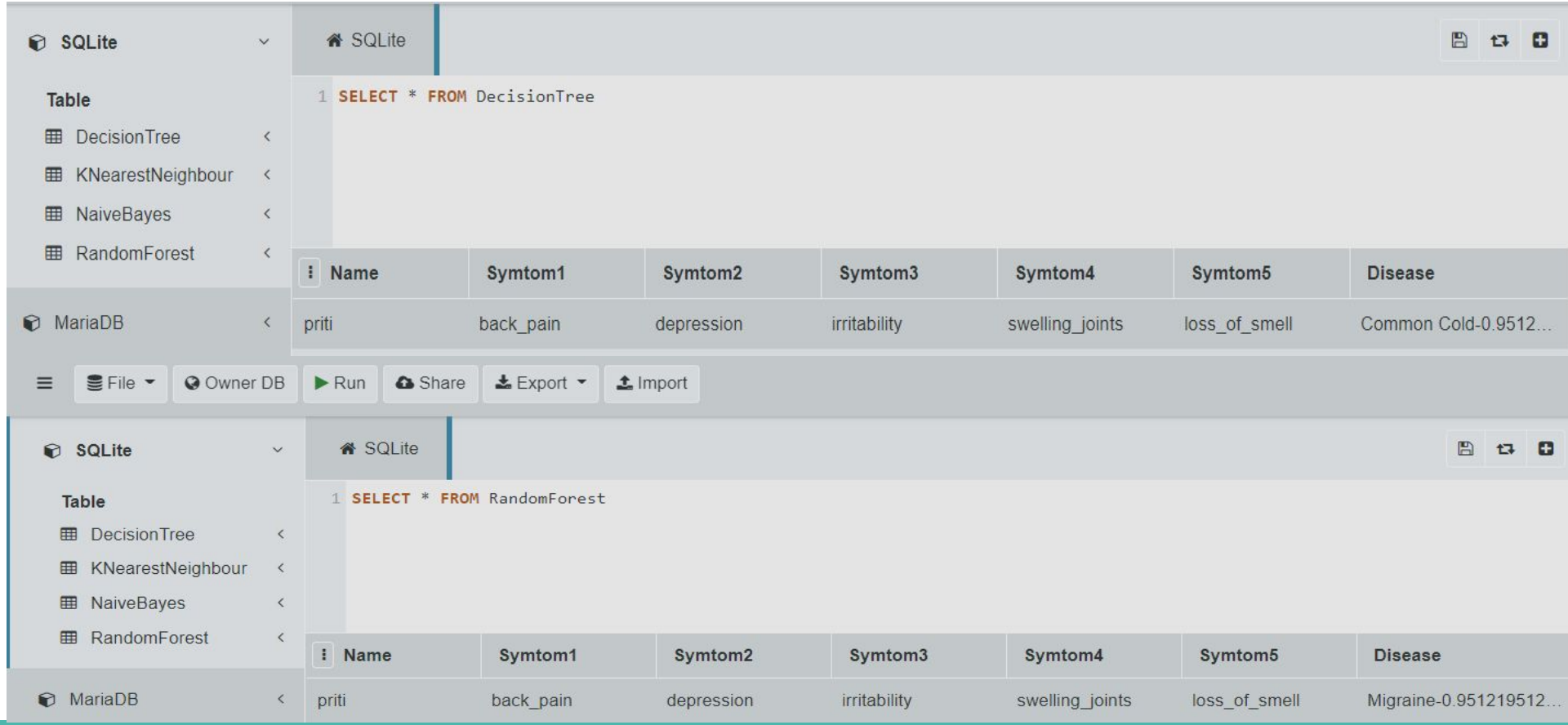

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

```
kNearest Neighbour
Accuracy
0.9268292682926833
8
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

The Final GUI presented to the user.



# The database Created using Sqlite3



# DATASOURCES:

- We have downloaded datasets from kaggle.

➢ https://www.kaggle.com/rabisingh/symptom-checker?select=Testing.csv

➢ http://people.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

# THANK YOU