

Abstract

**The project proposes a machine learning-based disease detection system to address the time-consuming and hectic nature of hospital visits and telephonic appointments. The system aims to provide immediate and as possible accurate disease detections based on user symptoms and detected severity. Different machine learning algorithms are used to ensure quick and accurate detections . The system uses a web/android application for easy portability and remote access, allowing users to manage their health more effectively. The system addresses the growing need for disease detection tools due to a variety of diseases and a reduced doctor-patient ratio. The system aims to provide proper guidance on healthy living and reduce the need for doctors to visit hospitals.**

Contents

**Topic Pg nos.**

1. **Introduction ………………………………………………………....1**
2. **Details Of Algorithms Used ………………………………. 4-8**
   1. **SVM …………………………………………………………………….4-5**
   2. **Naive Bayes Algorithm…………………………………………6-7**
   3. **RandomForestClassifier………………………………………….8**
3. **Flow Chart of Problem Solving……………………………..9**
4. **Code………….……………………………………………………15-22**
   1. **Required Libraries**
   2. **Reading Data For Training Dataset and splitting it for training and testing**
   3. **Performing K-Fold Cross Validation On Models**
   4. **SVM Classifier**
   5. **Naive Bayes Algorithm**
   6. **Random Forest Classifier- Decision Tree**
   7. **Operating On Test Data**
   8. **Fitting Whole Training Dataset Data into Models**
   9. **Testing Whole Testing Dataset Into Models**
5. **GUI - Converting System Into Flask Based Web App…………23-25**
   1. **GUI Output**

Disease Detection - Python

* Problem Definition

**Using Machine learning, my project proposes a disease detection system. For small problems, the users have to go personally to the hospital for check-up which is more time consuming.**

**Over the past decade, the use of the specific disease detection tools along with the concerning health has been increased due to a variety of diseases and less doctor-patient ratio. Thus, in this system, we are concentrating on providing immediate and accurate disease detection to the users about the symptoms. For detection of diseases, different machine learning algorithms are used to ensure quick and accurate detections. Therefore, this arrangement helps in easier health management.**

**Detecting diseases is a crucial endeavour in healthcare that can aid in early diagnosis and disease prevention. Based on medical characteristics, machine learning algorithms can be used to forecast the incidence of diseases. The goal of this research is to create disease detections models utilising machine learning algorithms, specifically Naive Bayes, Decision Tree, and Random Forest, and to assess how well these models perform in foretelling the development of heart disease based on specific medical characteristics.**

* Scope Of Further Development

**Further The Following Developments May Be Made In The Project:**

* **Add a database to store the user given data and collect it for future development and study and train the model according to it.**
* **Provide an android User Interactive Interface for input of symptoms and output of prognosis.**
* **Treatment for the detected disease may be prescribed in prognosis.**
* **Add a system to give users more tips about health care and self-hygiene and healthy dietary tips.**
* Demerits Of The Project

* **Uses Only Three Algorithms.**
* **No Search Bar Available.**
* **Limited Symptoms.**
* **Limited Prognosis Data.**
* Introduction - Dealing With The Problem

**During the time when Machine Learning and Deep Learning are booming so much , it is very important to understand that all this knowledge is not of any use if we can’t apply it to different areas and impact humanity.**

**In this project I have tried to create a Machine Learning Model using Python Programming Language. This Model uses the dataset from a renowned site https://www.kaggle .com . This dataset consists of two CSV files, one for training and one for testing. There are a total of 133 columns in the dataset out of which 132 columns represent the symptoms and the last column is the prognosis.**

**The Model Uses 3 ML Algorithms namely-**

* [**Support Vector Machine (SVM) Algorithm**](https://www.geeksforgeeks.org/support-vector-machine-algorithm/)
* **Naive Bayes Classifier**
* **RandomForestClassifier**

**for Disease Classification in order to obtain the most efficient result by taking mean of outputs from each model/algorithm and broadcasting the detected disease to the user.**

* **Implementation - Steps:**
* **Understanding and Cleaning Dataset:**

**Cleaning is the most important step in a machine learning project. it is always necessary to clean the data before feeding it to the model for training. In our dataset all the columns are numerical, the target column i.e. prognosis is a string type and is encoded to numerical form using a label encoder.**

* **Training Model:**

**I have used the K-Fold Cross Validation Method splitting training data into two parts 80% as Training Data And 20% Testing Data initially. After that I use some famous ML Algorithms for classification namely -**

**(1) SVM Classifier , (2) Naive Bayes Algorithm , (3) Random Forest Classifier.**

* **Validating Data/Testing The Model:**

**After training the three models we will detect the disease for the input symptoms by combining the detections of all three models. This makes our overall detection more robust and accurate.**

* **Other Utility and Libraries:**
* **Pandas**
* **Scikit-learn**
* **Numpy**
* **Matplotlib.**
* **Flask**
* **Statistics**
* Details Of Algorithms Used:

**Machine learning algorithms are essential components of artificial intelligence, allowing computers to learn from data and make detections without explicit programming. They are categorised into three main types here I am using:**

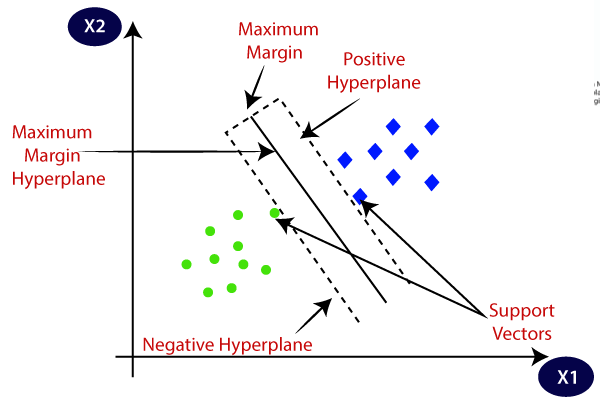
**Supervised Learning: Involves training on labelled data to detect outcomes. Common algorithms include linear regression, logistic regression, decision trees, and Support Vector Machines (SVMs).**

* SVM- Support Vector Machine Classifier:

**Support Vector Machines (SVMs) are a type of supervised learning algorithm used for classification and regression tasks. They are particularly effective in high-dimensional spaces and are known for their ability to find optimal decision boundaries (hyperplanes) that separate different classes of data.**

## Key Concepts

1. **Hyperplane: A hyperplane is a decision boundary that separates different classes in the feature space. In two dimensions, it is a line; in three dimensions, it is a plane.**
2. **Support Vectors: These are the data points that are closest to the hyperplane. They are critical in defining the position of the hyperplane and directly influence the margin.**



1. **Margin: The margin is the distance between the hyperplane and**

**the nearest support vectors from either class. SVMs aim to**

1. **maximise this margin, which helps improve the model's generalisation to unseen data.**

## Types of SVM

* **Linear SVM: Used when the data can be separated by a straight line (or hyperplane). It is efficient and interpretable.**
* **Non-linear SVM: Utilises kernel functions to transform the input space into a higher-dimensional space where a linear separation is possible. Common kernels include polynomial and radial basis function (RBF) kernels.**

**In summary, SVMs are powerful classifiers that excel in various applications, particularly when dealing with complex and high-dimensional datasets. Their ability to maximise the margin between classes makes them robust and effective for many machine learning tasks.**

* Naive Bayes Classifier

## Naive Bayes Classifier Explained

**Naive Bayes is a simple yet powerful supervised learning algorithm used for classification tasks. It is based on Bayes' theorem and makes detections by calculating the probability of an instance belonging to a particular class given the feature values.**

## Bayes' Theorem

**Bayes' theorem is the foundation of the Naive Bayes classifier. It states that:**

****

**Where:**

* **$P(A|B)$ is the probability of event A occurring given event B has occurred**
* **$P(B|A)$ is the probability of event B occurring given event A has occurred**
* **$P(A)$ and $P(B)$ are the probabilities of observing $A$ and $B$ independently**

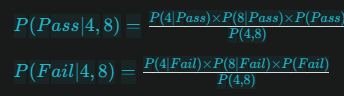
## How Naive Bayes Works

**Naive Bayes makes the assumption that features are independent of each other. This means that the presence of one feature does not affect the other. This assumption simplifies the calculations but is often violated in real-world scenarios.**

**Let's illustrate with an example. Suppose we want to classify whether a student will pass or fail an exam based on their study hours and sleep hours. The training data is as follows:To classify a new instance,**

| **Study Hours** | **Sleep Hours** | **Label** |
| --- | --- | --- |
| 5 | 7 | Pass |
| 10 | 4 | Pass |
| 2 | 10 | Fail |
| 7 | 6 | Pass |
| 3 | 9 | Fail |

**say (4, 8), we calculate the probability of passing and failing given the feature values:**

****

**The class with the higher probability is the detected label. In this case, (4, 8) is more likely to fail.**

**In summary, Naive Bayes is a simple and efficient classifier that makes detections based on Bayes' theorem and the assumption of feature independence.**

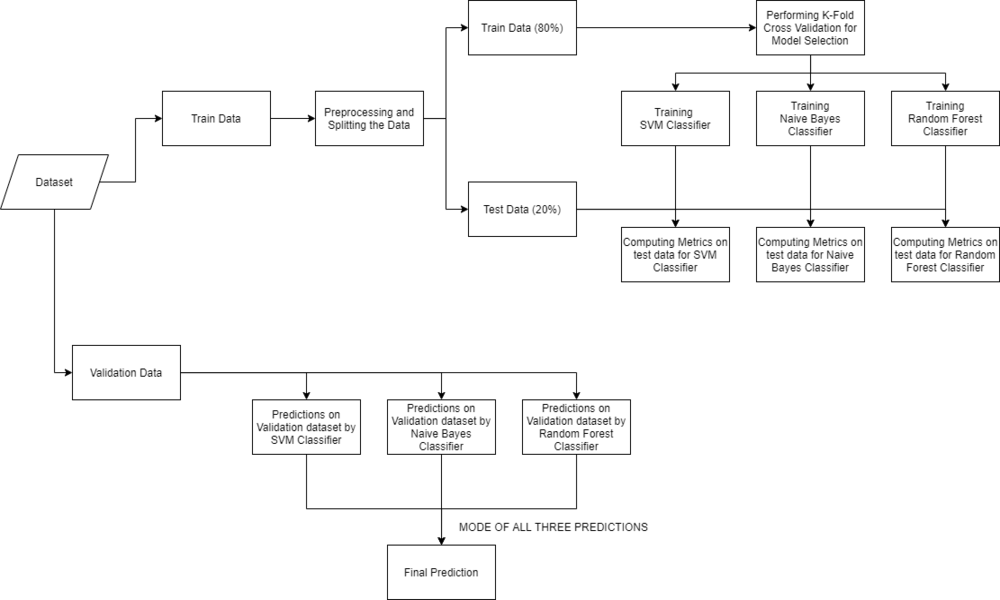
* **RandomForestClassifiers**

**Random Forest is an ensemble learning method used for classification tasks. It creates multiple decision trees from randomly selected subsets of the training data and aggregates their detections to improve accuracy and prevent overfitting.**

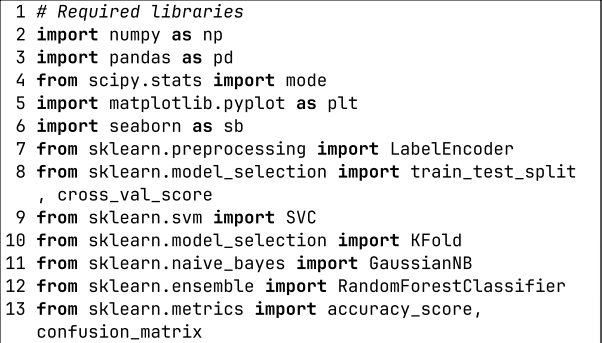
## **How it Works**

1. **Random subsets: The algorithm creates multiple decision trees using random subsets of the training data.**
2. **Feature selection: At each node split, a random subset of features is considered, and the best split is chosen based on criteria like information gain or Gini impurity.**
3. **Voting: Each tree in the forest makes a calculation and gives an output. The class with the most votes across all trees is the final detection.**
4. **Bagging: Random Forest uses a technique called Bootstrap Aggregating (Bagging) to create diverse subsets of the training data for each tree.**

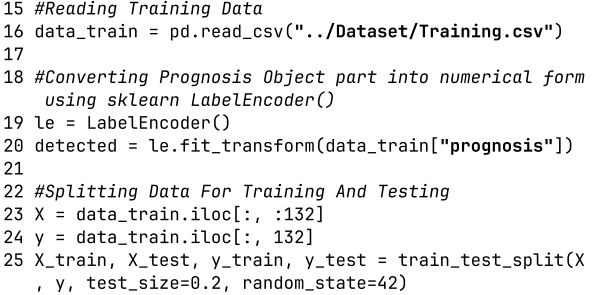
* Flow Chart Of Problem Solving



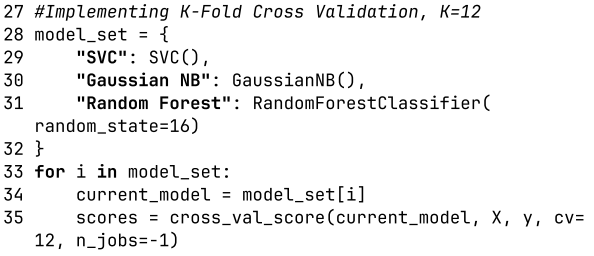
* **Code**
* **Required Libraries**

****

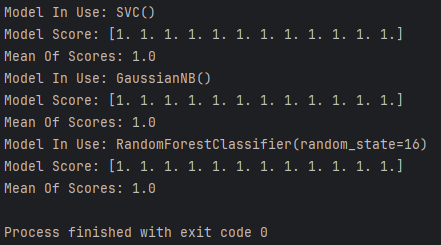
* **Reading Data For Training Dataset and splitting it for training and testing**



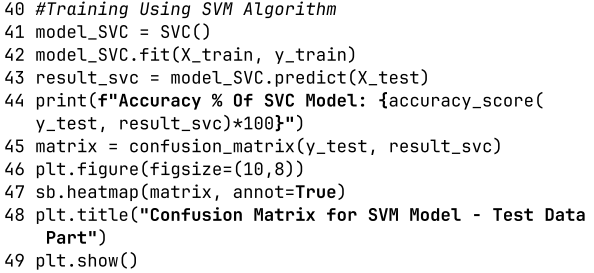
**Performing K-Fold Cross Validation On Models**

****

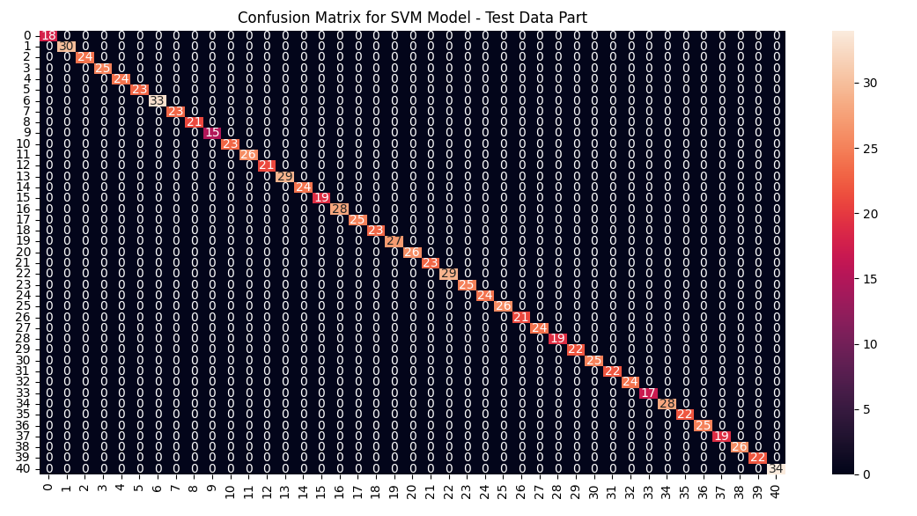
**Output**



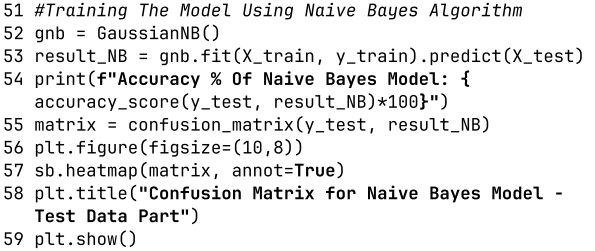
* **SVM Classifier**

****

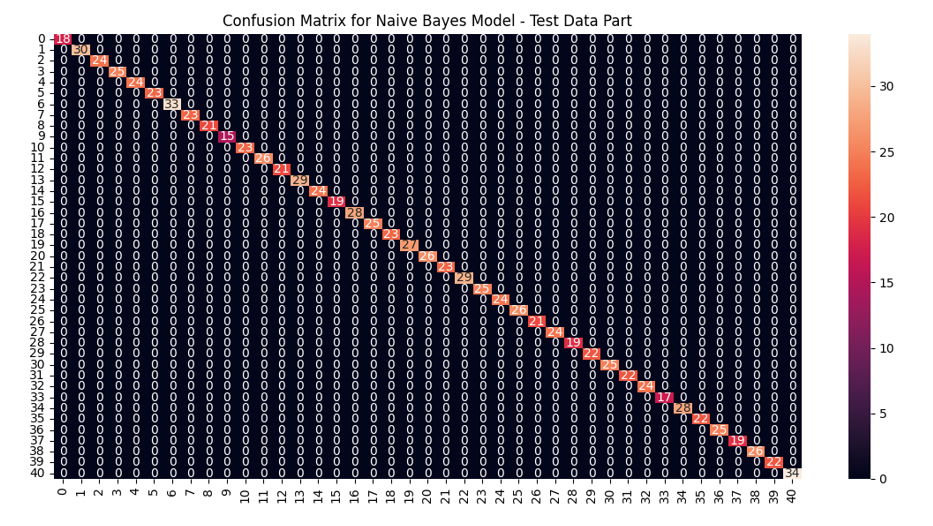
**Output:**

****

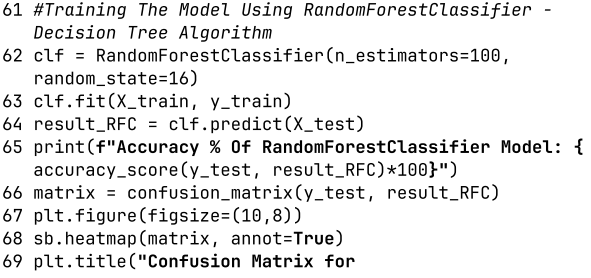
* **Naive Bayes Algorithm**

****

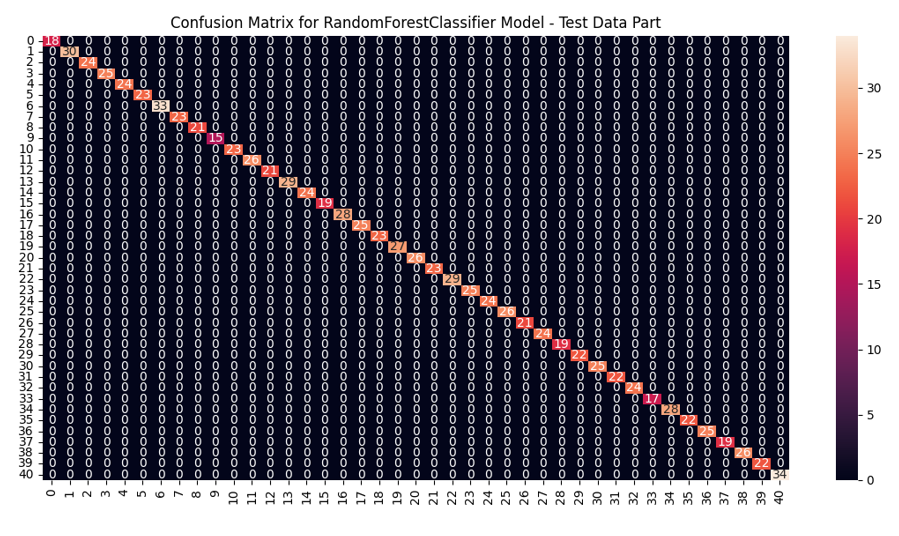
**Output:**

****

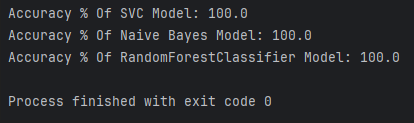
* **Random Forest Classifier- Decision Tree**

****

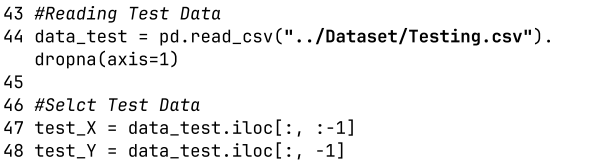
**Output:**

****

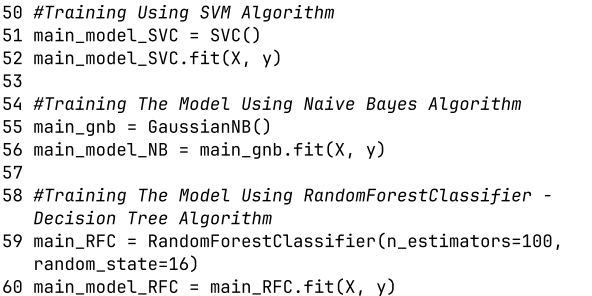
* **Output Of Accuracies Of All Models At Once:**

****

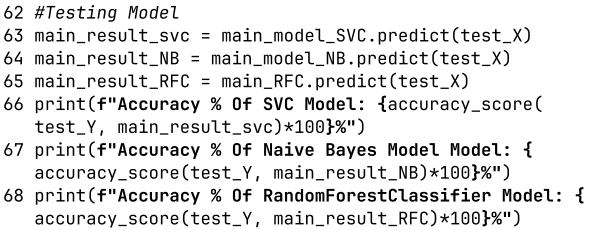
* **Operating On Test Data**

****

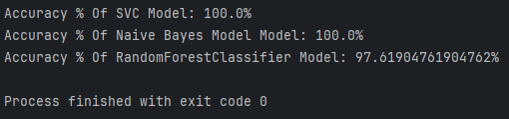
* **Fitting Whole Training Dataset Data into Models**

****

* **Testing Whole Testing Dataset With Models**

****

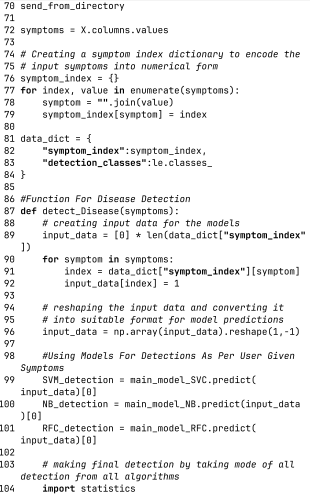
**Output:**

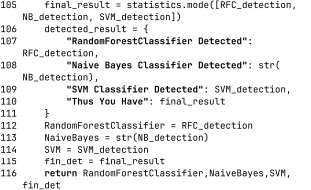
****

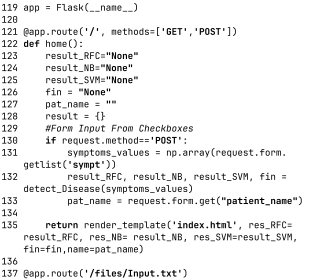
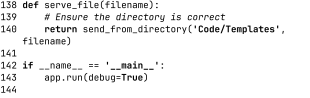
* **GUI - Converting System Into Flask Based Web App:**

**./Code/Main.py**

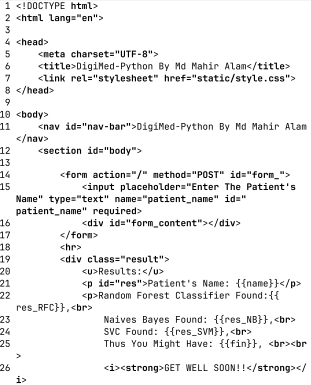
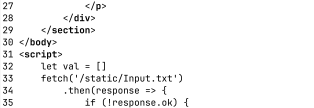


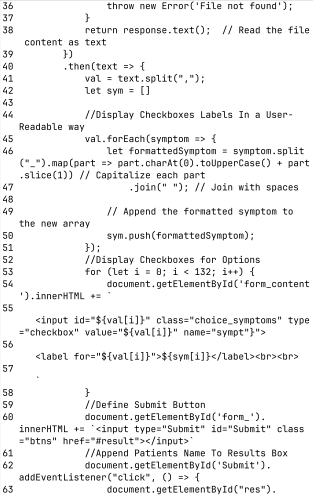
****

****

****

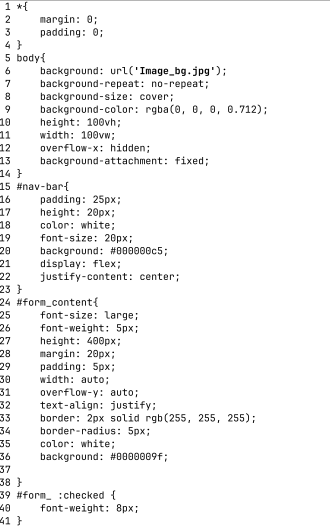
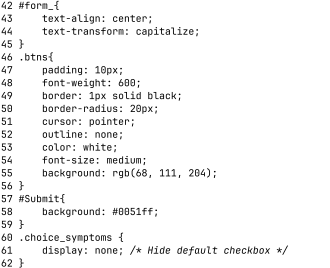
**./Code/Template/index.html**

****

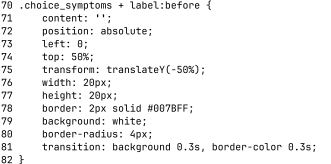
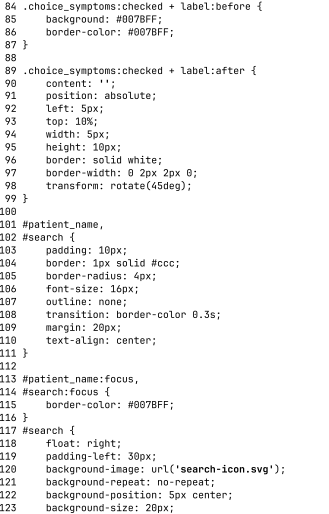
****

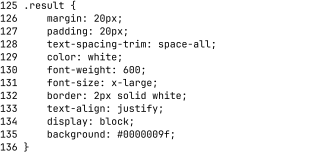
**Note: static/Input.txt contains all symptoms list**

**/Code/static/style.css**

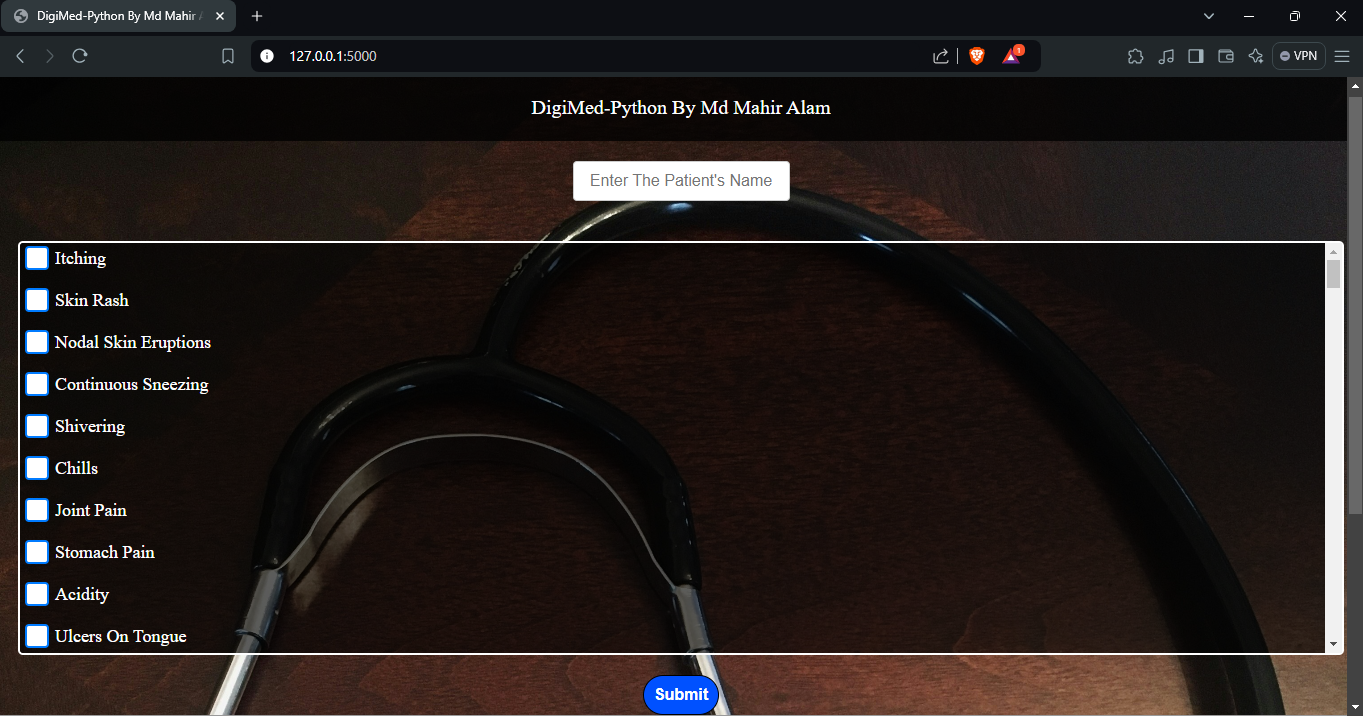
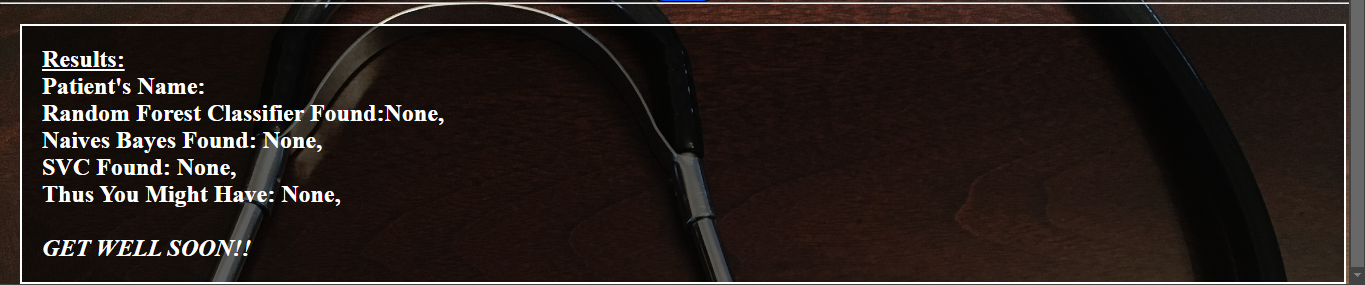
****



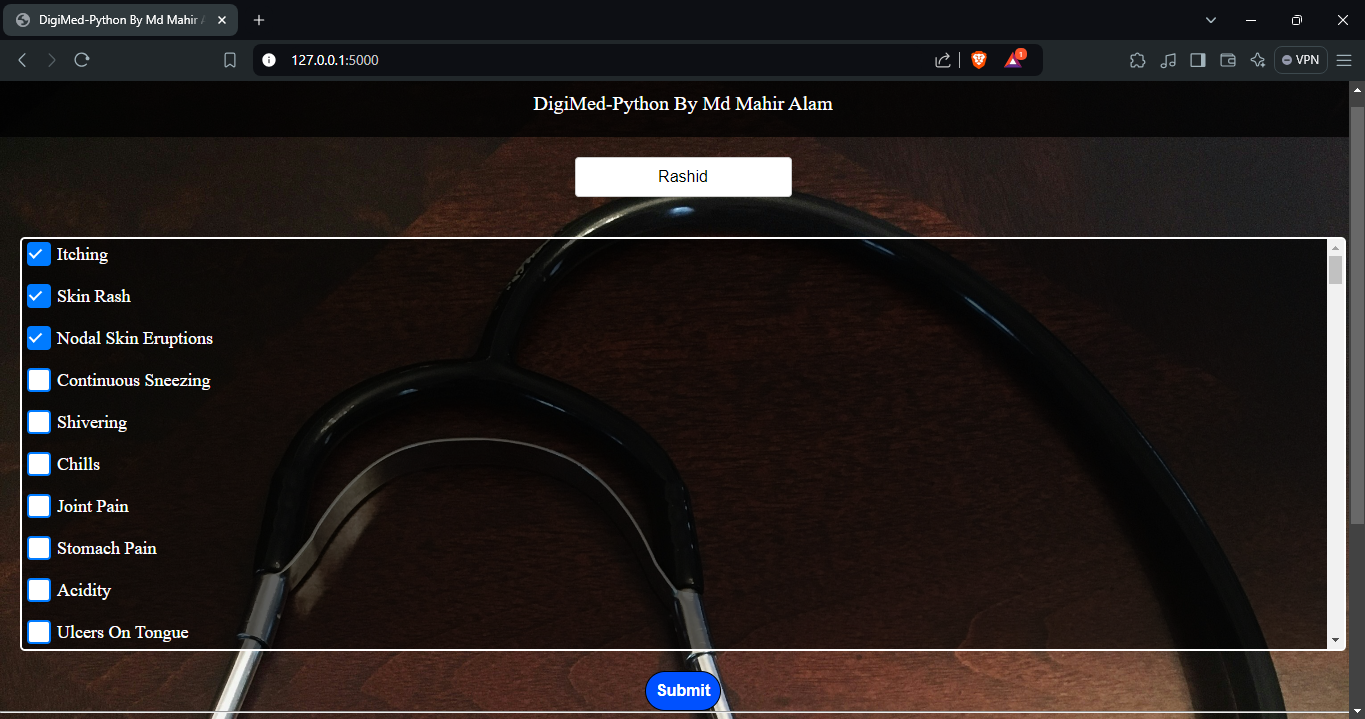
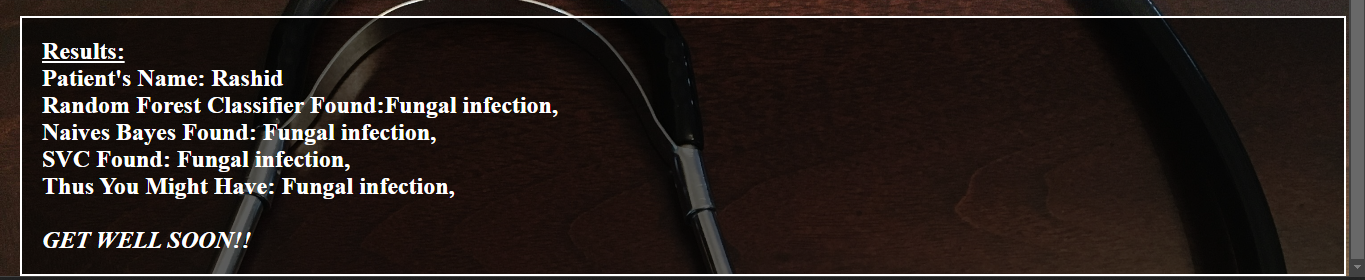
****

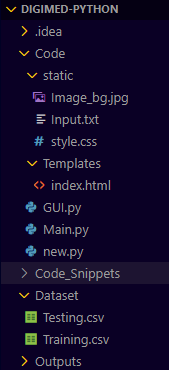


**GUI-Output:**

****

**After Input:**

****

**Pro**j**ect Structure:**

Conclusion

**This system was developed to predict diseases based on symptoms, reducing hospital OPD rushes and medical staff workload. Using three different algorithms, the system achieved an average accuracy of approximately ~99.2% . The system is reliable to a good extent and easy to use, with a user-friendly interface and various visual representations of collected data. The system also includes a database for future improvements and an easy-to-use interface, allowing for future improvements. The system's reliability and user-friendly nature make it a valuable tool for healthcare professionals.**

Bibliography

* [**https://geeksforgeeks.com/**](https://geeksforgeeks.com/)
* [**https://scikit-learn.org/**](https://geeksforgeeks.com/)
* [**https://flask.palletsprojects.com/**](https://flask.palletsprojects.com/)
* [**https://kaggle.com**](https://kaggle.com)