DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

Major Project Report

(Multi-Disease Prediction using Machine learning)

By

Prajwal Kittur-ENG18CS0210

Sangamesh-ENG18CS0243

Shankar Shindhe-ENG18CS0255

Under the supervision of

Dr. Jayavrinda Vrindavanam V

Department of CSE(AI&ML)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,

SCHOOL OF ENGINEERING

DAYANANDA SAGAR UNIVERSITY,

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"MULTI-DISEASE PREDICTION USING MACHINE LEARNING"** is carried out by **Prajwal Kittur (ENG18CS0210), Sangamesh (ENG18CS0243), Shankar Shindhe (ENG18CS0255),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr.Jayavrinda Vrindavanam V	**Dr Girisha G S**	**Dr. A Srinivas**
Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Prajwal Kittur (ENG18CS0210), Sangamesh (ENG18CS0243), Shankar Shindhe (ENG18CS0255),** are students of the seventh semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the phase-II project titled **"Multi-Disease Prediction Using Machine Learning"** has been carried out by us and submitted in partial fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Signature

Student

Name1: Prajwal Kittur
USN : ENG18CS0210

Name2: Sangamesh
USN : ENG18CS0243

Name3: Shankar Shindhe
USN : ENG18CS0255

Place : Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice.It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr.Jayavrinda Vrindavanam V Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator Dr. Meenakshi **Malhotra** and **Dr,Bharanidharan N**, all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

List of Abbreviations

RAM - Random Access Memory

GB – Gigabytes

ANN- Artificial Neural Networks

RF – Random Forest

DT – Decision Tree

CART – Classification and Regression Trees

HTML5- Hypertext Markup Language 5

CSS3- Cascading Style Sheets 3

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A Machine Learning Approach to Disease Prediction is based on a prediction model, which predicts disease in patients based on symptoms reported by users as an i/p to the system. This project illustrates the concept of utilizing Machine Learning algorithms to forecast numerous illnesses. We will leverage the notion of supervised Machine Learning in this case, with implementation done by using Decision Tree, Logistic Regression, Random Forest, CNN algorithms, which will aid in the accurate prediction of illnesses and improved patient treatment. The findings assured that the system would be effective and user-friendly for patients in order to provide rapid illness diagnoses.

CHAPTER 1 INTRODUCTION

The Earth is going through a purplish period of technology, with an increase in the demand for intelligence and precision following it. People today are most likely glued to the internet, yet they are unconcerned about their physical health. People disregard minor issues and do not attend hospitals, which leads to the development of major ailments over time. Taking use of this developing technology, our primary goal is to create a system that can forecast many diseases based on symptoms provided by patients without requiring them to visit hospitals or clinics.

The goal is to develop a good Machine Learning model that is efficient and accurate for illness prediction. The supervised Machine Learning concept is applied to forecast illnesses in this article. The key feature will be Machine Learning, in which we will use algorithms such as Logistic Regression,Random Forest,Decision Tree,CNN to aid with accurate illness prediction and better patient care.

This project considered Diabetes analysis, Heart disease, and Malaria detection data sets. In the future, many other diseases like skin diseases can be included, fever-related diseases, and many more. This analysis is flexible that later included many diseases for analysis. While adding any new disease analysis to this existing API, the developer has to add the model file related to the analysis of the new disease.

The aim of the proposed model is to predict diseases like Diabetes, Heart, and Malaria. And it is also used to prevent the mortality ratio from increasing day by day by warning the patients in advance based on their health conditions. Due to many disease models and predictions done in one place the cost of patient analysis can be reduced.

1.1 PURPOSE

The fundamental goal of disease prediction is to estimate the likelihood of an individual developing a disease in the future. For various diseases in different populations, there is a huge variety of contributing factors that must be considered. Complex and varied individual and disease differences must be differentiated, as well as a set of traits with extraordinarily wide dimensions that must be discovered. Manually doing these jobs is not only difficult but also consumes a lot of human and financial resources.

1.2 SCOPE

The scope of the project is to create an efficient and accurate Machine Learning model for disease prediction. In this project, the supervised Machine Learning concept is used to forecast diseases. The main feature will be Machine Learning, in which we will employ algorithms like Random Forest, Logistic Regression, DT, and CNN to help with accurate sickness prediction and better patient care. The disease diagnosis system will permit end-users to predict heart disease and diabetes.

CHAPTER 2 PROBLEM STATEMENT

The primary goal is to develop a prediction engine that will allow the users to check whether they have Diabetes, Heart disease, and Malaria sitting at home. The user need not visit the doctor unless he has diabetes or heart disease, for further treatment. The prediction engine requires a large dataset and efficient machine learning algorithms to predict the presence of the disease. Pre-processing the dataset to train the machine learning models, removing redundant, null, or invalid data for optimal performance of the prediction engine.

Doctors rely on common knowledge for treatment. When common knowledge is lacking, studies are summarized after a number of cases have been studied. But this process takes time, whereas if machine learning is used, the patterns can be identified earlier. For using machine learning, a huge amount of data is required. There is a very limited amount of data available depending on the disease. Also, the number of samples having no diseases is very high compared to the number of samples having the disease.

CHAPTER 3 LITERATURE REVIEW

SL NO.	TITLE	AUTHORS	DESCRIPTION	YEAR
1	**Diabetes Disease prediction using ML algorithms**	Arwatki Chen Lyngdoh,Nurul Amin Choudhury, Soumen Moulik.	This paper analyses diabetes disease prediction using five supervised machine learning algorithms: K-Nearest Neighbors, Nave Baye, Decision Tree Classifier, Random Forest, and Support Vector Machine. We were able to get a steady and greatest accuracy of 76% with the KNN classifier, while the remaining classifiers also provide a stable accuracy of over 70 %.	2020
2	**Using Machine Learning Algorithms For Prediction Of Diabetes Mellitus**	Aeshah Saad Alanazi, Mohd A. Mezher.	This paper consists of the proposed model that integrates two machine learning methods, which are Support Vector Machine and Random Forest. Using a genuine dataset from the Security Force Primary Health Care. The suggested model has a 98% accuracy and a 99% ROC. The results reveal that the Random Forest method outperforms the Support Vector Machine approach in terms of accuracy.	2021
3	**Comparison of Machine Learning Algorithms for Prediction of Diabetes**	Naomi Estera Costea,Elisa Valentina Moisi,Daniela	This paper compares the experimental findings gained using three machine learning algorithms in the prediction of diabetes. The three algorithms under	2021

		Elena Popescu.	consideration are support vector machine, Naive Bayes, and random forest. We discovered that support vector machine and random forest had an accuracy of more than 80%	
4	**Heart Disease Prediction using Machine Learning Algorithms.**	Santhana Krishnan J, Geetha S.	The datasets are processed in python programming using two main Machine Learning algorithms namely Decision Tree Algorithm and Naive Bayes Algorithm which shows the best algorithm among these two in terms of accuracy level of heart disease.	2019
5	**Human Heart Disease Prediction Using Data Mining Techniques**	Imran Mirza, Arnav Mahapatra, Daryl Rego,Kenneth Mascarenhas	This paper proposes the model using RBF SVM and Linear SVM classifiers, as well as KNN and Naive Bayes classifiers, to categorise users into groups that are non-zero for the severity of presence and zero for the absence of heart disease, and to test the effectiveness of our classifiers.	2019

6	**Machine Learning for Real-Time Heart Disease Prediction**	Dimitris Bertsimas, Luca Mingardi, Bartolomeo Stellato	This paper uses the XGBoost algorithm, a popular machine learning approach, to train models with out-of-sample F1 Scores ranging from 0.93 to 0.99. This is, to the best of our knowledge, the first paper that reports great performance across hospitals, nations, and recording standards.	2021
7	**Malaria Disease Prediction Based on Machine Learning**	Octave Iradukunda; Haiying Che; Josiane Uwineza; Jean Yves Bayingana; Muhammad S Bin-Imam; Ibrahim Niyonzima	Machine learning techniques like SVM, KNN, CART, RF, CNN, VGG16, RESNET with an accuracy of 99%, 28 seconds cost time, 0.0095 Misclassification Error, and 98% precision	2019
8	**Predicting malarial outbreak using MachineLearning and Deep Learning approach: Areviewand analysis**	Godson Kalipe, Vikas Gautham, Rajat Kumar Behera	In this paper, we tried to find out which algorithm is best suited for modeling the discovered relationship. For that purpose, historical meteorological data and records of malarial cases, collected over six years, have been combined and aggregated in order to be analyzed with various classification techniques such as KNN, Naive Bayes, and Extreme Gradient Boost among others.	2018

CHAPTER 4 PROJECT DESCRIPTION

4.1 Proposed Flowchart

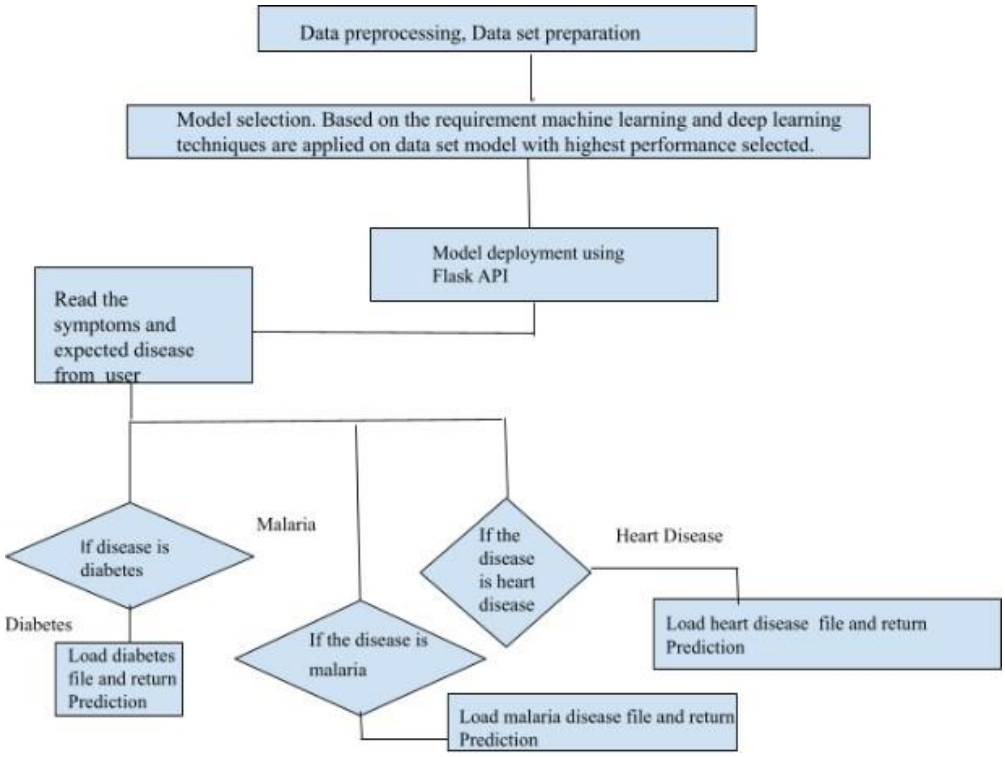


Figure 4.1 Flowchart of diseases

The first step is to preprocess the dataset. After the data set is prepared, selection of the model will be done based on the requirements, techniques related to machine learning and deep learning are applied on the prepared data set model that has the highest accuracy algorithm. The selected model will be deployed using Flask API. Then we read the input from the user for the different diseases the user has clicked on. Once the input has been given for the specific disease, the model predicts whether he or she is suffering from the selected disease or not.

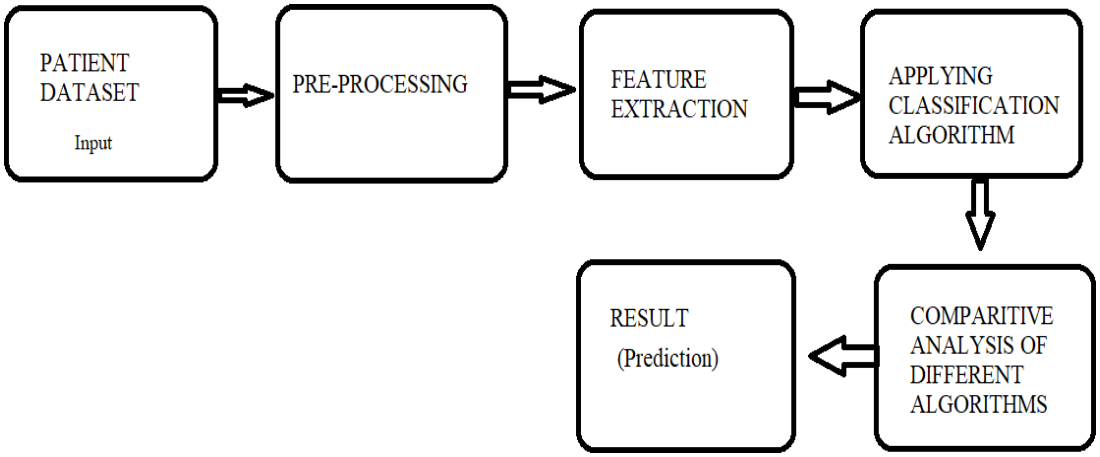


Figure 4.1.1 Diagrammatic flow of each disease

Once the datasets have been taken, we preprocess the dataset to find the null values or missing values present in the dataset. In the feature extraction step, we check for the most dominant attribute which can assist in predicting the disease with more accuracy. For heart disease and diabetes, we have considered DT, RF and LR algorithms and for malaria disease, CNN algorithm is opted. Then the comparative analysis of the algorithm is done and the algorithm with the highest accuracy is selected. Finally, we deploy the model with the highest accuracy for the prediction.

4.2 Proposed design

The project's purpose is to examine the model's ability to forecast different diseases with greater precision. To forecast different diseases, we will test several classification and ensemble techniques. The phase will be discussed briefly in the following sections

CHAPTER 5 REQUIREMENTS

5.1 Functional Requirements

- Web scraping/ Data Mining: Collect all the images and other data from open-source websites.
- Perform Exploratory data analysis of Textual data.
- Feature Engineering.
- Image and data pre-processing module to prepare the data we mined before subjecting it to a Model building.
- Machine Learning models, out of which we will select the best one for deployment.
- Develop and deploy a dynamic web app.

5.2 Non-Functional Requirements

5.2.1 Usability

The system should be easy to use. The system also should be user-friendly for users because anyone can use it instead of programmers.

5.2.2 Reliability

This software will be developed with machine learning, feature engineering, and deep learning techniques. So, in this step, there is no certain reliable percentage that is measurable. Also, user-provided data will be used to compare with results and measure reliability. With recent machine learning techniques, user-gained data should be enough for reliability if enough data is obtained.

5.2.3 Performance

Processing time and response time should be as little as possible providing the result at a faster rate when compared to other methods.

5.2.4 Supportability

The system should require Python knowledge for maintenance. If any problem is acquired in the user side and machine learning methods, it requires code knowledge and a machine learning background to solve.

5.3 Software Requirements

- Python
- A functioning Web browser
- Visual Studio Code
- Pycharm

.

5.4 Hardware Requirements

- Windows 8 or above
- 4 GB Ram
- 256 GB Internal Storage
- Intel I5 and above/ Ryzen 5 and above

CHAPTER 6 METHODOLOGY

.

6.1 Dataset Description

In this project, we will create three separate machine learning models for diagnosing three different diseases, therefore we will use various datasets from the UCI Machine Learning repository and Kaggle for each ailment.

Diabetes Dataset

Table 6.1.1 Diabetes Attributes

Sl no.	Attributes
1	Pregnancy
2	Glucose
3	Blood Pressure
4	Skin Thickness
5	BMI(Body Mass Index)
6	Diabetes Pedigree Function
7	Insulin
8	Age

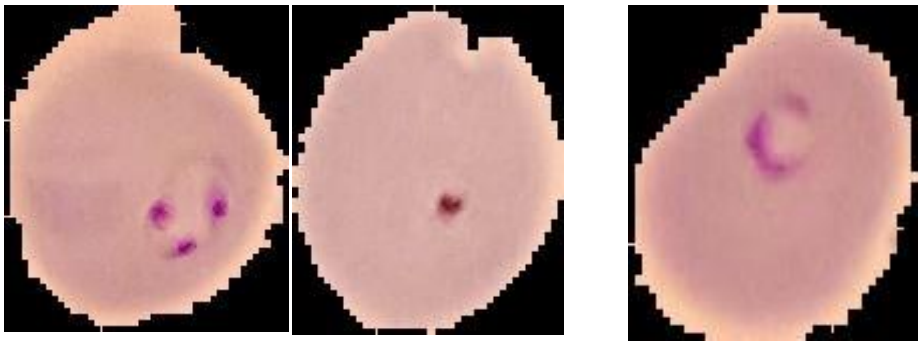
Heart Disease Dataset

Table 6.1.2 Heart Disease Attributes

Sl no.	Attributes
1	Age (age in years)
2	Sex (1 = male; 0 = female)
3	ChestPainType (chest pain type)
4	RestingBP (resting blood pressure)
5	Cholesterol (serum cholestoral in mg/dl)
6	FastingBS (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7	RestingECG (resting electrocardiographic results)
8	MaxHR(Maximum Heart Rate)
9	ExerciseAngina
10	Oldpeak
11	ST_Slope

Malaria Dataset

Parasite



Uninfected

6.2 Data Preprocessing

The most crucial procedure is data preprocessing. The majority of healthcare-related data has missing values and other contaminants that might reduce the efficiency. To boost quality and efficiency data preprocessing is mandatory. The method is critical for accurate findings and good prediction when using Machine Learning Techniques on a dataset. To assist the model in producing better predictions for the three separate diseases.

6.3 Feature Extraction

The Feature Extraction procedure is used to update the crucial data for outcome characteristics. This method aims to reduce the number of resources required to explain a huge quantity of data. The process of minimizing the number of characteristics is known as feature extraction. This is also used to increase the speed and efficacy of supervised learning.

6.4 Apply Machine Learning

Applying machine learning algorithms to predict disease.

The algorithms are as follows

- Logistic Regression
- Decision Tree
- Random Forest
- CNN(Convolutional Neural Network)

CHAPTER 7 EXPERIMENTATION

- We have used different machine learning algorithms to train and test for the different diseases in order to obtain the best accuracy.
- With the best accuracy algorithm, we will be deploying the machine learning model. For diabetes, we used DT,LR,RF algorithms and got 82% accuracy with RF.
- Here we used DT for heart disease and got an accuracy of 91%.
- And for Malaria we have used the CNN algorithm where the accuracy for the dataset is 93%.
- Flask API is used for web app.

7.1 LOGISTIC REGRESSION

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

7.2 DECISION TREE

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contain possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -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.

7.3 CNN

CNN is a model which is designed to process arrays of data such as images. The first step here will be to resize all images as CNN cannot train images of different sizes. We compute the mean for both dimensions and resize all the images.

The sequential model is used here. In the first layer which is the Convolution layer, we will place the filter on top of the input matrix and then compute the value and will be doing a stride jump of 1. This extracts features from the image. Also, we can use Padding if the filter does not fit perfectly in the input image. Here we will be using the Relu activation function.

Max pooling selects the maximum element and extracts the most prominent features from the image. Last is the fully connected layer, where the input to the fully connected layer will be the output from the Max Pooling Layer; it is flattened and then fed into the fully connected layer

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size

7.5 Random Forest

1. Randomly select "n" features from the total "k" features. Where n < < k
2. Among the "n" features, calculate the node "n" using the best split point.
3. Categorize the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until "l" number of nodes has been reached.
5. Build a forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

CHAPTER 8 TESTING AND RESULT

Figure 8.1 Diabetes disease Prediction

Figure 8.2 Heart disease Prediction

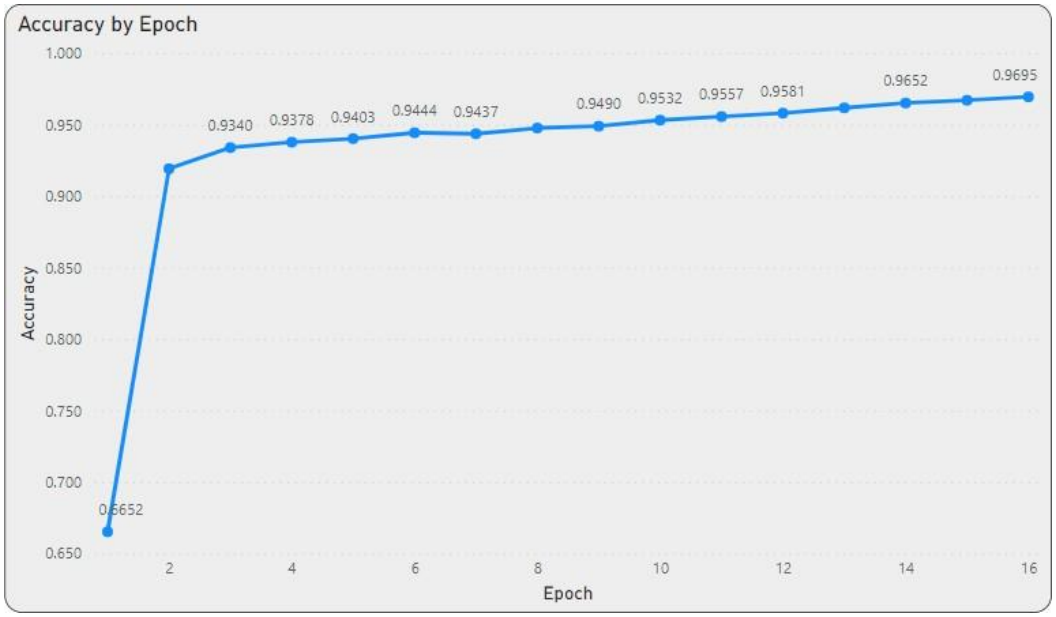


Figure 8.3 Number of Epoch Used for malaria disease along with accuracy

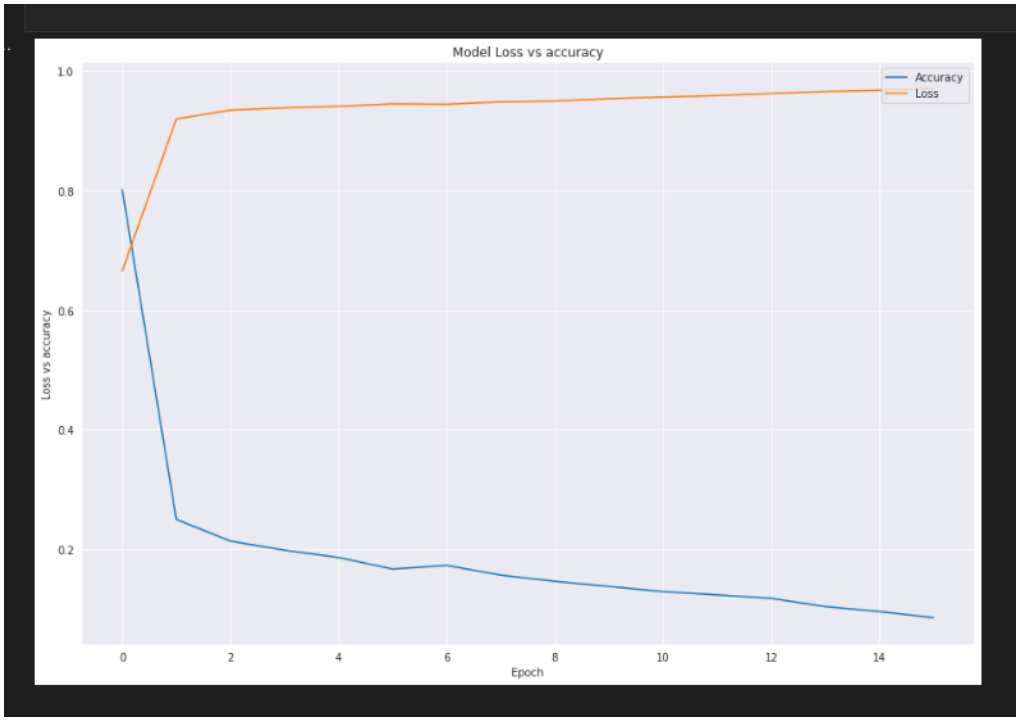


Figure 8.4 Graphical Representation of Epoch

	precision	recall	f1-score	support
Infected	0.99	0.86	0.92	1600
Normal	0.88	0.99	0.93	1600
accuracy			0.93	3200
macro avg	0.93	0.93	0.93	3200
weighted avg	0.93	0.93	0.93	3200

Figure 8.5 Accuracy of Malaria Disease

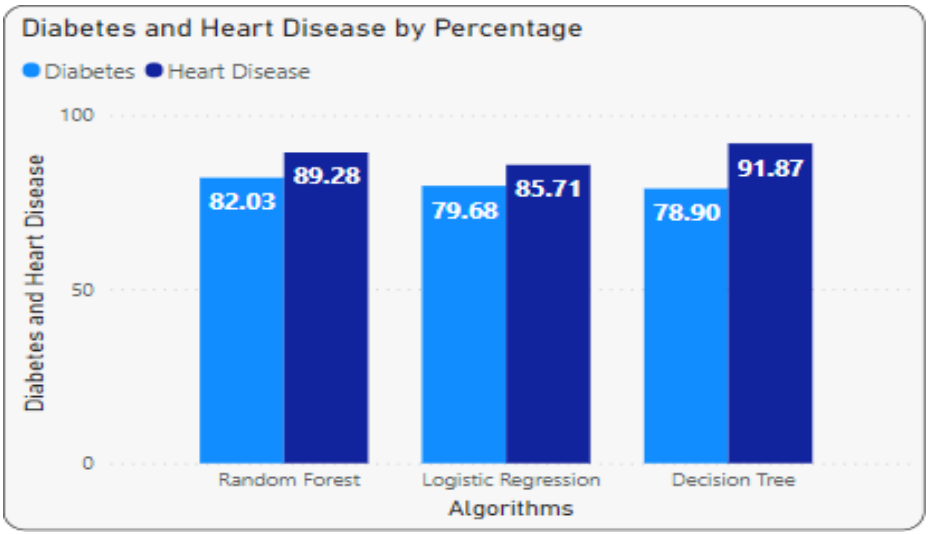


Figure 8.6 Accuracy of algorithms

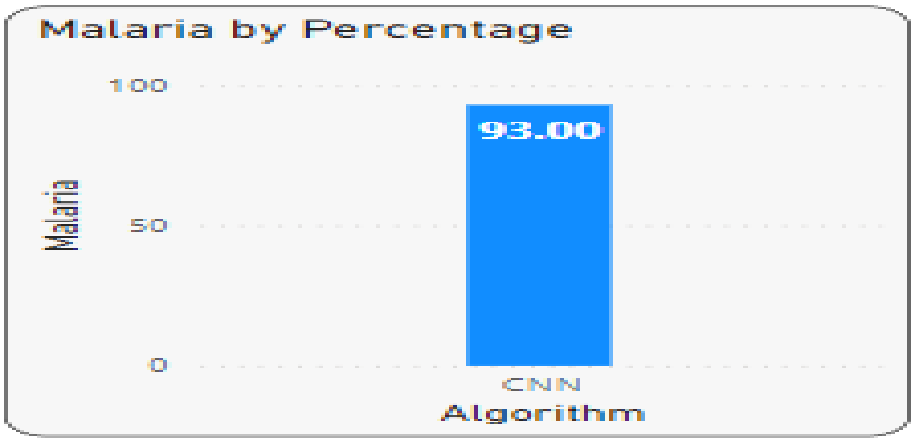


Figure 8.7 Accuracy of CNN Algorithm

Algorithms/Disease	Diabetes	Heart Disease
Decision Tree	78.90	91.87
Logistic Regression	79.68	85.71
Random Forest	82.03	89.28

Figure 8.8 Accuracy of Models

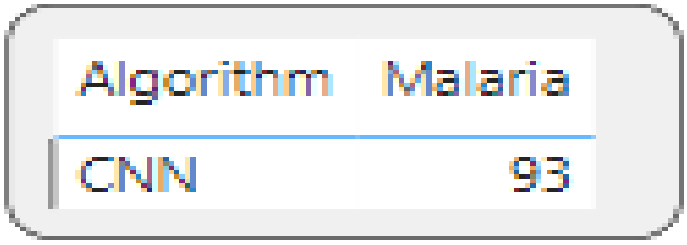


Figure 8.9 Accuracy of model

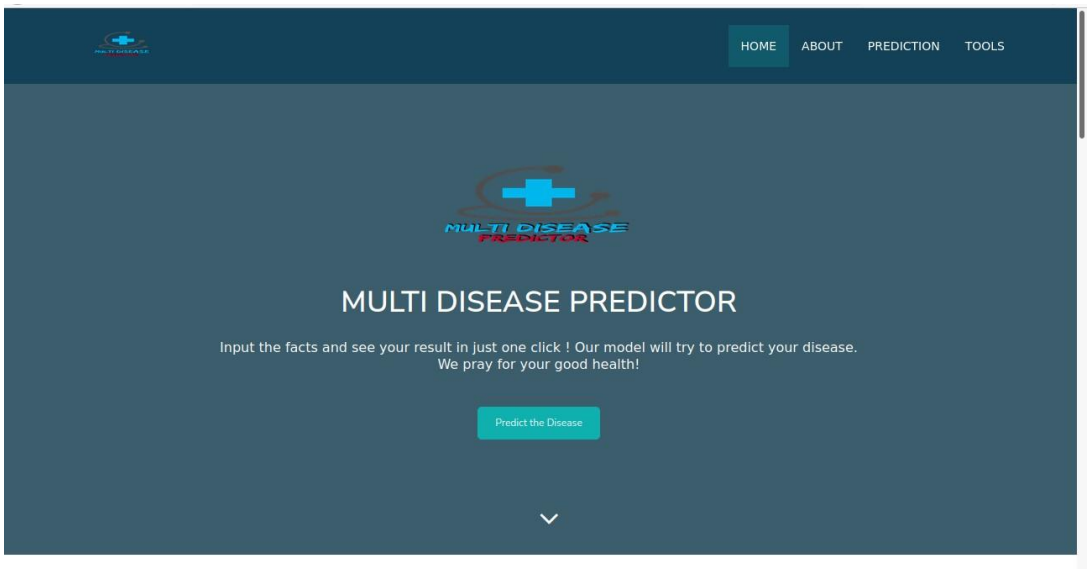


Figure 8.10 Home Page

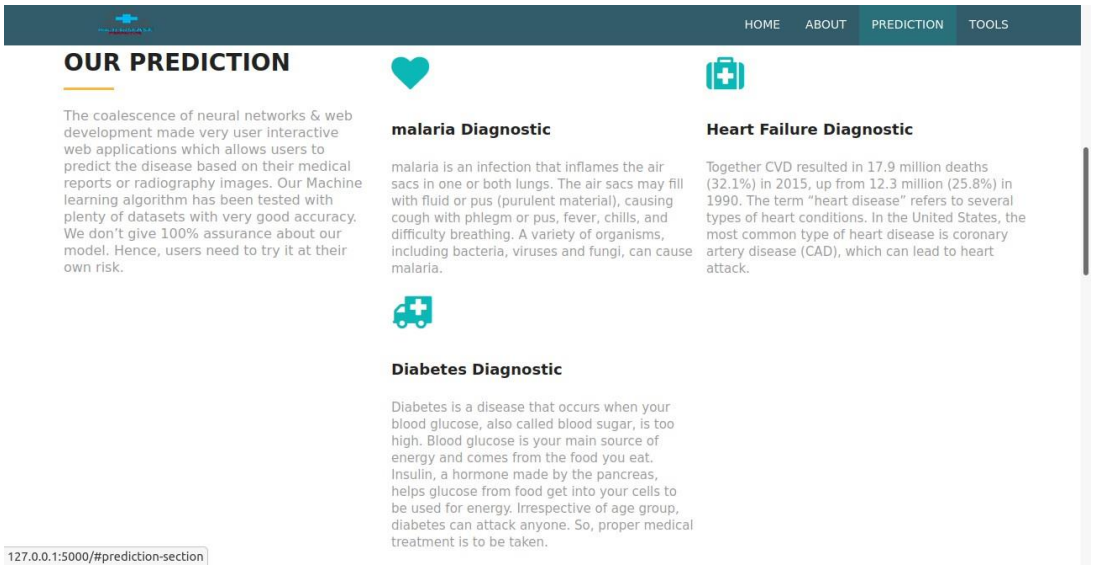


Figure 8.11 Predictions

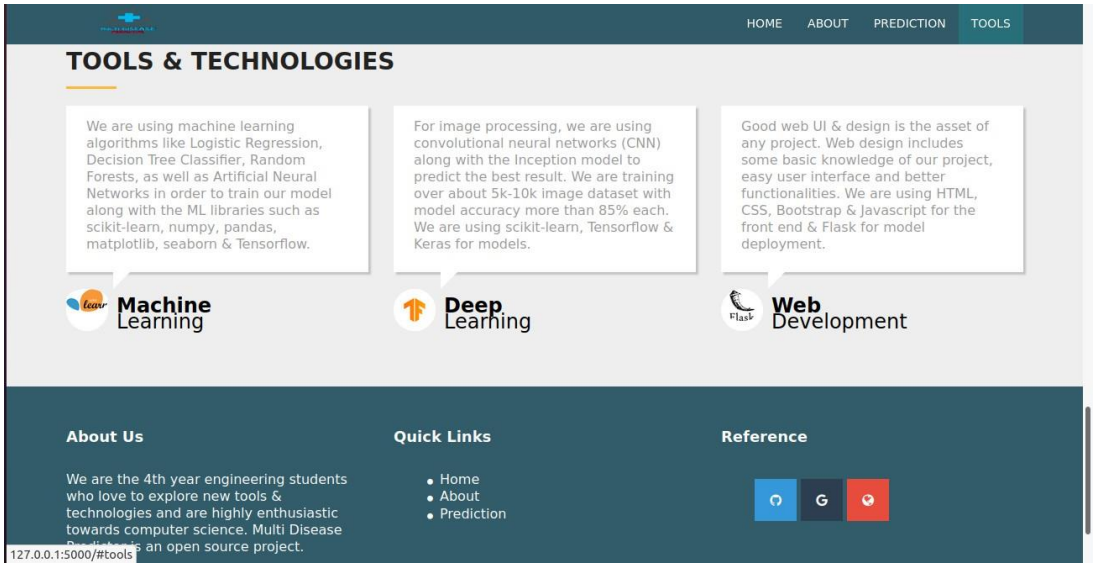


Figure 8.12 Tools

Malaria

Figure 8.13 Malaria Prediction

Diabetes

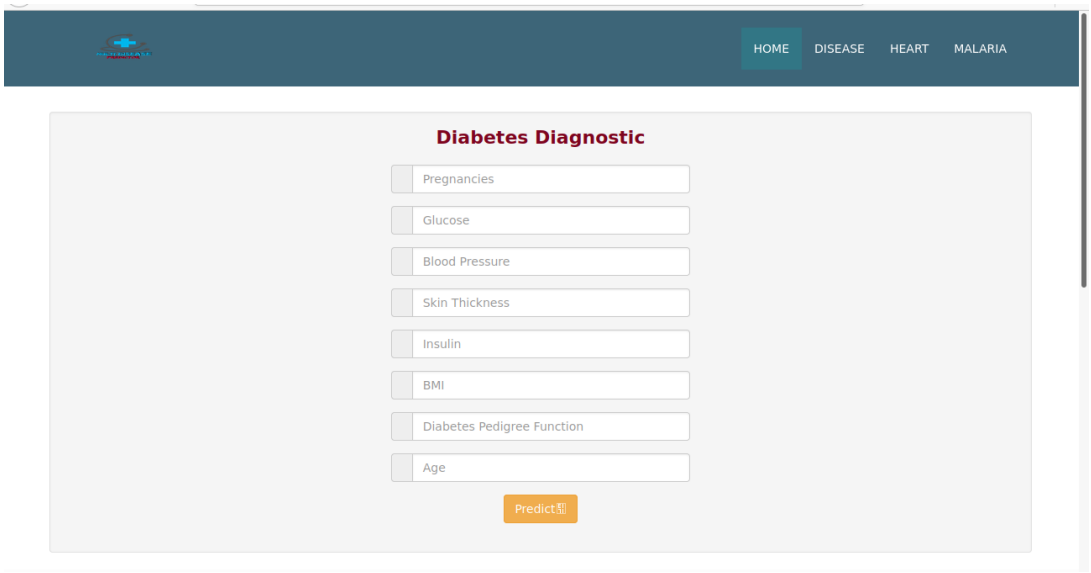


Figure 8.14 Diabetes Prediction

Heart

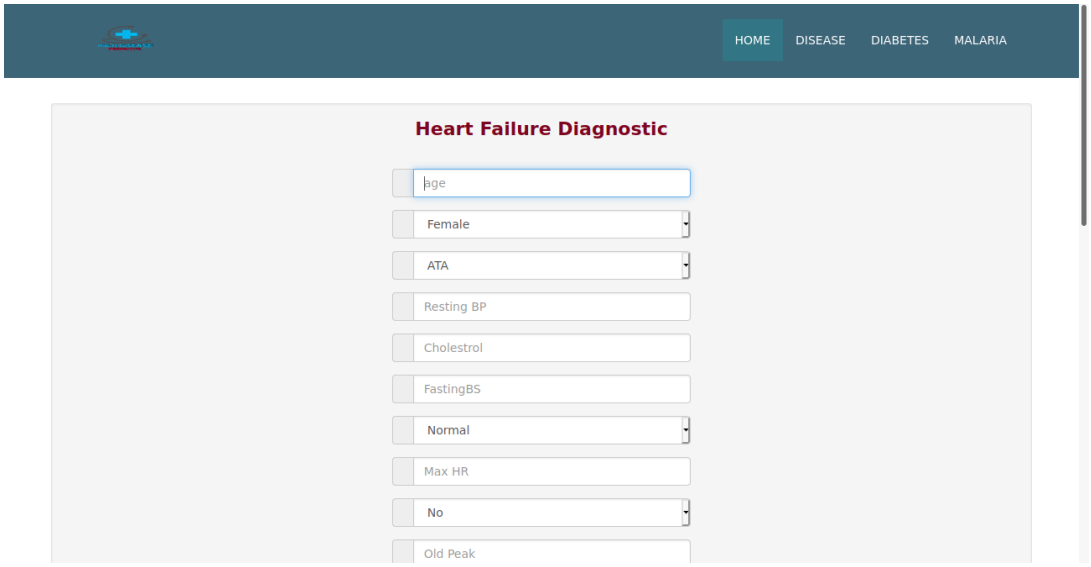


Figure 8.15 Heart Disease Prediction

CHAPTER 9 CONCLUSION AND FUTURE WORK

The use of different ML algorithms enabled the early detection of many diseases such as heart, diabetes and malaria. DT, RF, LR and CNN algorithms were the most widely used at prediction, while accuracy was the most used performance metric. The CNN model proved to be the most adequate at predicting malaria diseases. Furthermore, the RF model showed superiority in accuracy at most times for Diabetes diseases . For Heart Disease prediction, DT showed more superiority in the probability of correct classification of the diseases because of its ability to scale well for large datasets and its susceptibility to avoid overfitting. Finally, the algorithms proved to be the most reliable in predicting heart diseases, malaria and diabetes.

In the future, more advanced machine learning algorithms will be required to improve disease prediction efficiency. Furthermore, learning models should be calibrated more frequently after the training period to improve performance. Furthermore, to minimize over fitting and improve the accuracy of deployed models, datasets should be enlarged on diverse demo-graphics. Finally, to improve the performance of learning models, more relevant feature selection approaches should be applied.

REFERENCES

[1] A. C. Lyngdoh, N. A. Choudhury, and S. Moulik, "Diabetes Disease Prediction Using Machine Learning Algorithms," 2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), 2021, pp. 517-521, DOI: 10.1109/IECBES48179.2021.9398759.

[2] A. S. Alanazi and M. A. Mezher, "Using Machine Learning Algorithms For Prediction Of Diabetes Mellitus," 2020 International Conference on Computing and Information Technology (ICCIT-1441), 2020, pp. 1-3, DOI: 10.1109/ICCIT-144147971.2020.9213708.

[3] N. E. Costea, E. V. Moisi and D. E. Popescu, "Comparison of Machine Learning Algorithms for Prediction of Diabetes," 2021 16th International Conference on Engineering of Modern Electric Systems (EMES), 2021, pp. 1-4, doi: 10.1109/EMES52337.2021.9484116

[4] S. K. J. and G. S., "Prediction of Heart Disease Using Machine Learning Algorithms.," 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1-5, doi: 10.1109/ICIICT1.2019.8741465.

[5] I. Mirza, A. Mahapatra, D. Rego and K. Mascarenhas, "Human Heart Disease Prediction Using Data Mining Techniques," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 2019, pp. 1-5, doi: 10.1109/ICAC347590.2019.9036836.

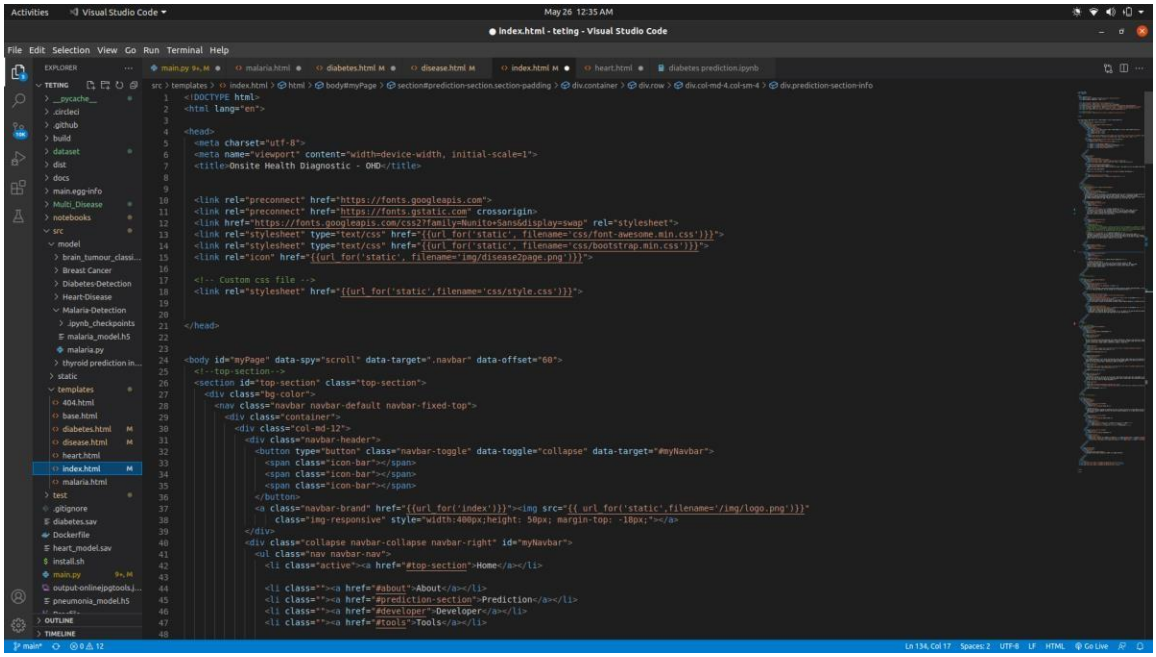
[6] D. Bertsimas, L. Mingardi and B. Stellato, "Machine Learning for Real-Time Heart Disease Prediction," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3627-3637, Sept. 2021, doi: 10.1109/JBHI.2021.3066347.

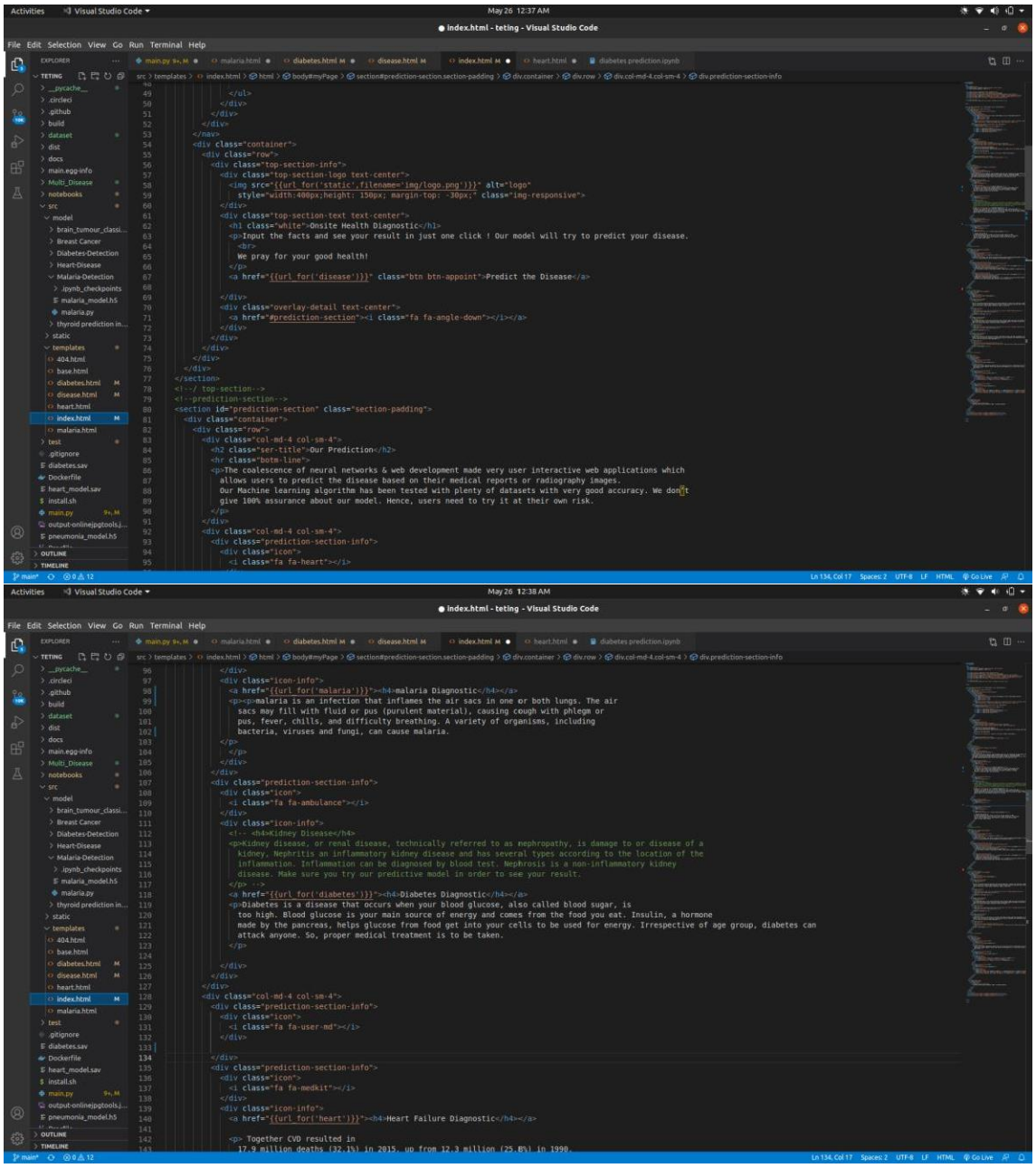
[7] O. Iradukunda, H. Che, J. Uwineza, J. Y. Bayingana, M. S. Bin-Imam and I. Niyonzima, "Malaria Disease Prediction Based on Machine Learning," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1-7, doi: 10.1109/ICSIDP47821.2019.9173011.

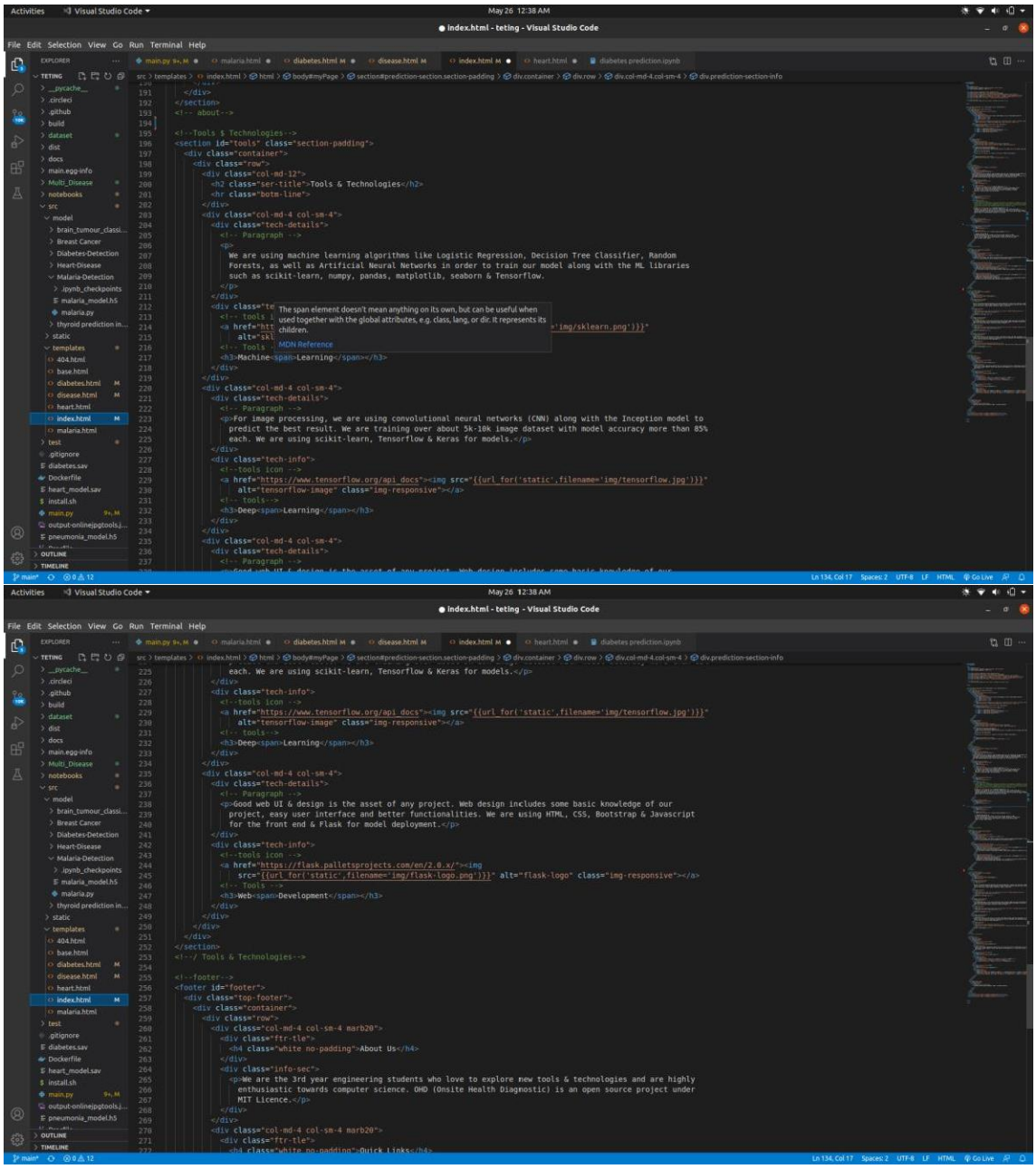
[8] Godson Kalipe, Vikas Gautham, Rajat Kumar Behera, "Predicting malarial outbreak using MachineLearning and Deep Learning approach: A review and analysis International Conference on Information Technology (ICIT),2018, DOI:10.1109/ICIT.2018.00019

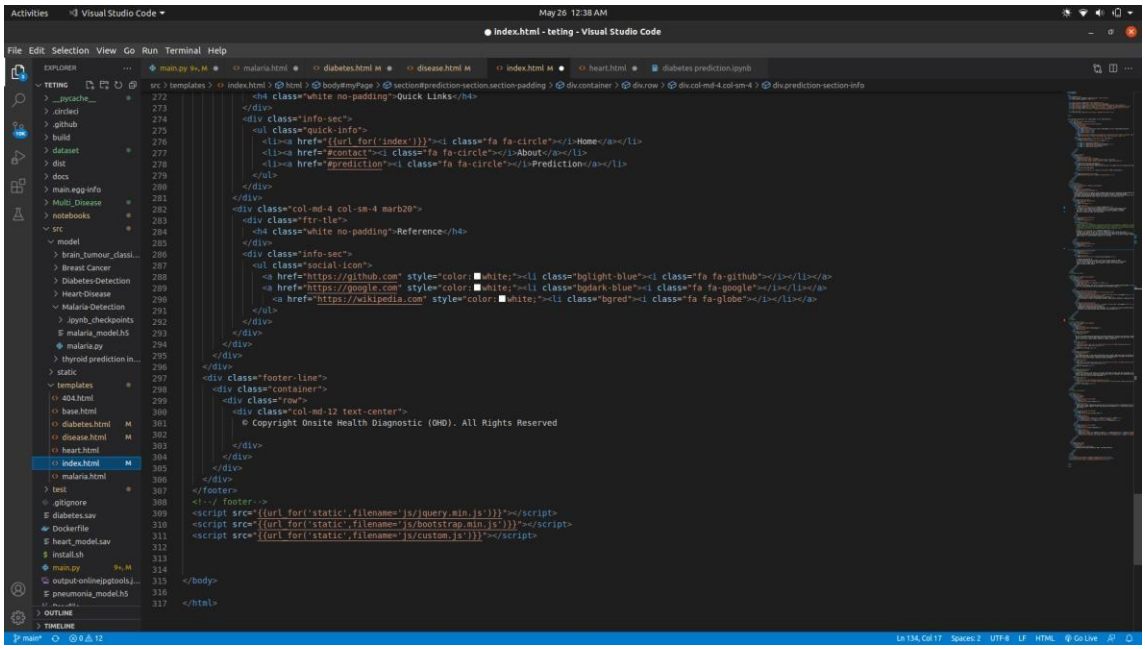
APPENDIX A

INDEX.HTML






Activities
Visual Studio Code
May 26 12:38 AM
index.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > index.html > html > body#myPage > section#prediction-section.section-padding > div.container > div.row > div.col-md-4.col-sm-4 > div.prediction-section-info
<!--Tools $ Technologies-->
<section id="tools" class="section-padding">
<h2 class="ser-title">Tools & Technologies</h2>
<hr class="botm-line">
<h3>MachineLearning</h3>
<h3>DeepLearning</h3>
For image processing, we are using convolutional neural networks (CNN) along with the Inception model to predict the best result. We are training over about 5k-10k image dataset with model accuracy more than 85% each. We are using scikit-learn, Tensorflow & Keras for models.
Good web UI & design is the asset of any project. Web design includes some basic knowledge of our project, easy user interface and better functionalities. We are using HTML, CSS, Bootstrap & Javascript for the front end & Flask for model deployment.
<h3>WebDevelopment</h3>
<!--footer-->
<footer id="footer">
<h4 class="white no-padding">About Us</h4>
We are the 3rd year engineering students who love to explore new tools & technologies and are highly enthusiastic towards computer science. OHD (Onsite Health Diagnostic) is an open source project under MIT Licence.
<h4 class="white no-padding">Quick Links</h4>
Ln 134, Col 17
Spaces: 2
UTF-8
LF
HTML
Go Live


Activities
Visual Studio Code
May 26 12:38 AM
index.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > index.html > html > body#myPage > section#prediction-section.section-padding > div.container > div.row > div.col-md-4.col-sm-4 > div.prediction-section-info
<h4 class="white no-padding">Quick Links</h4>

<ul class="quick-info">
Home
About
Prediction

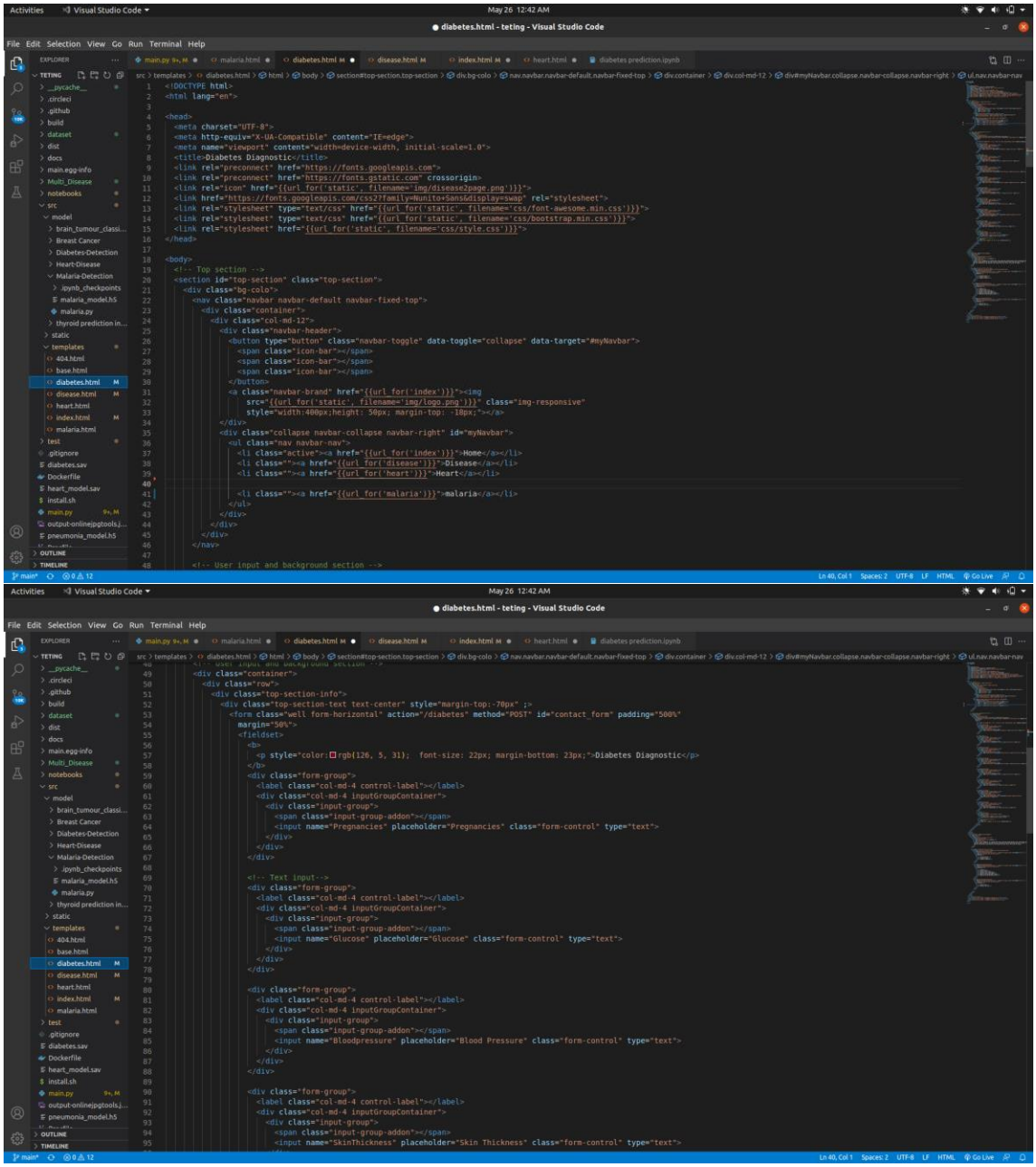
<h4 class="white no-padding">Reference</h4>

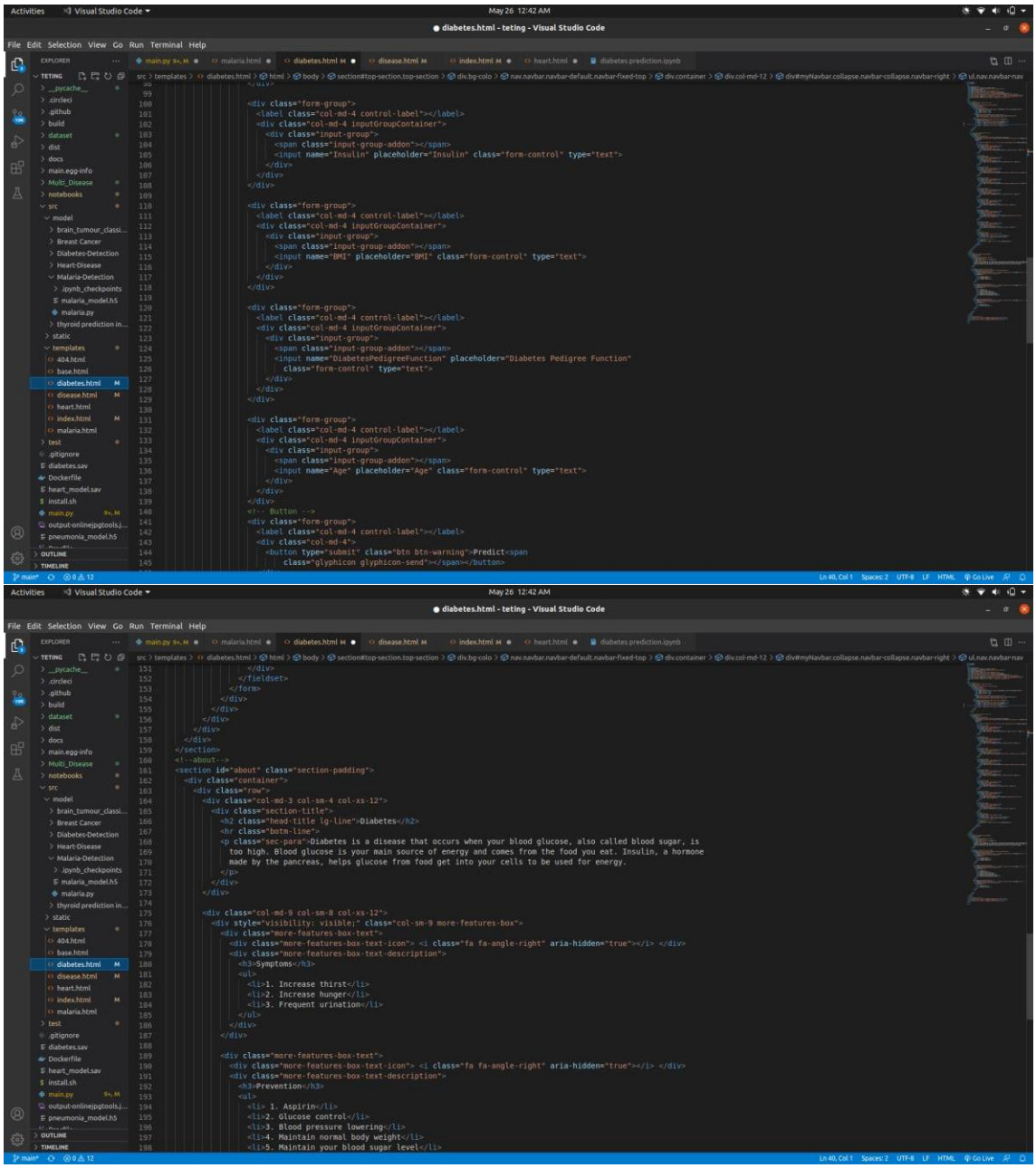
<ul class="social-icon">
<li class="bglight-blue">
<li class="bgdark-blue">
<li class="bgred">

© Copyright Onsite Health Diagnostic (OHD). All Rights Reserved

</footer>
<!-- /.footer-->
<script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<script src="{{url_for('static',filename='js/bootstrap.min.js')}}"></script>
<script src="{{url_for('static',filename='js/custom.js')}}"></script>
</body>
</html>
Ln 134, Col 17
Spaces: 2
UTF-8
LF
HTML
Go Live

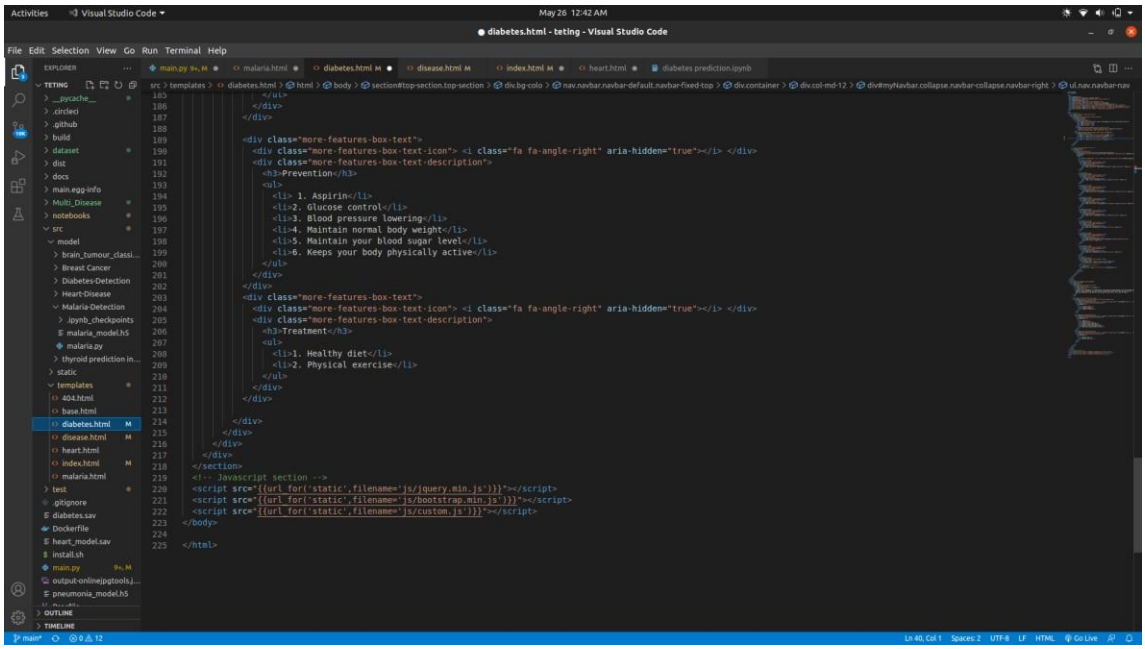
Diabetes.html



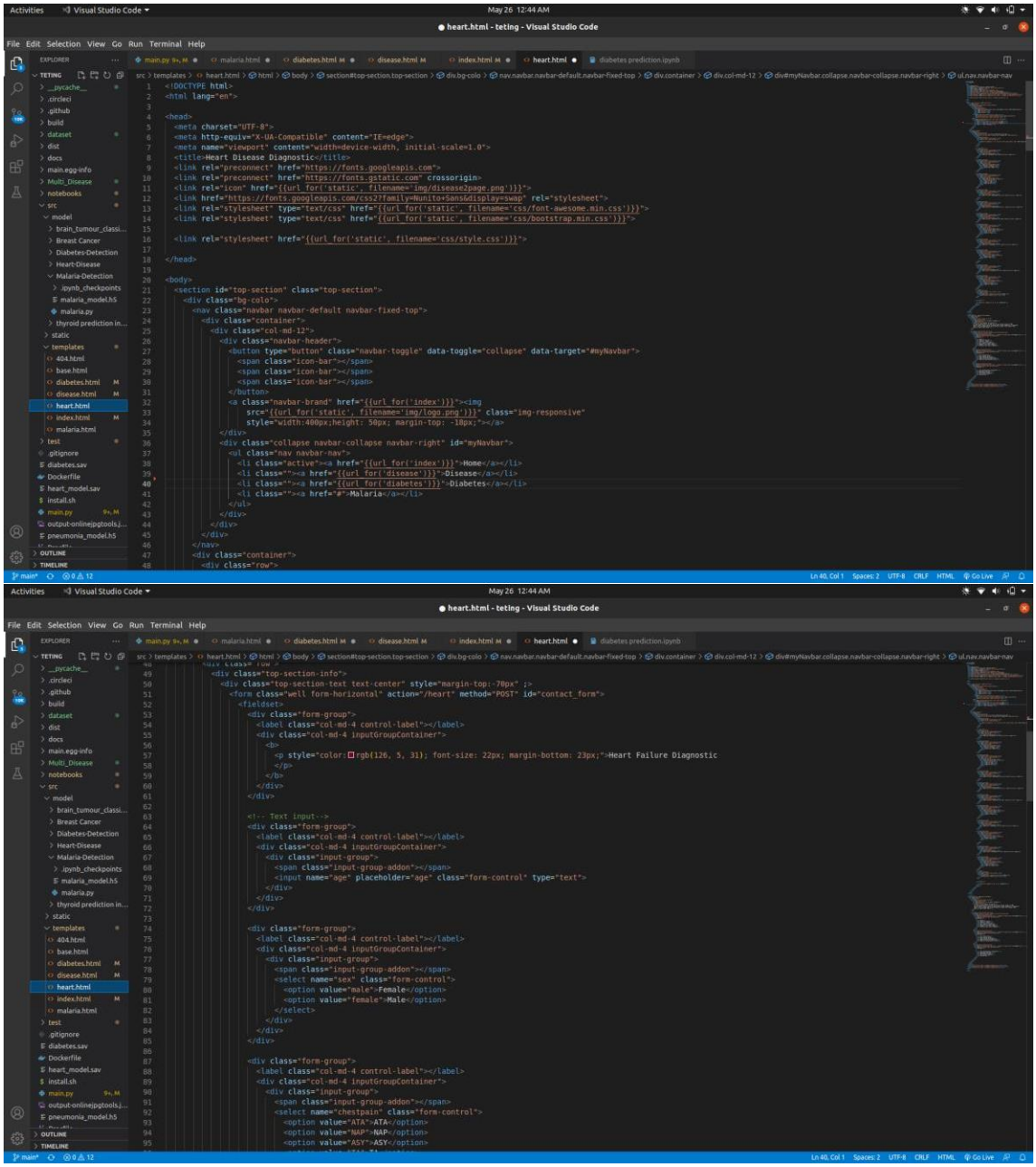

Activities
Visual Studio Code
May 26 12:42 AM
diabetes.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
<label class="col-md-4 control-label"></label>

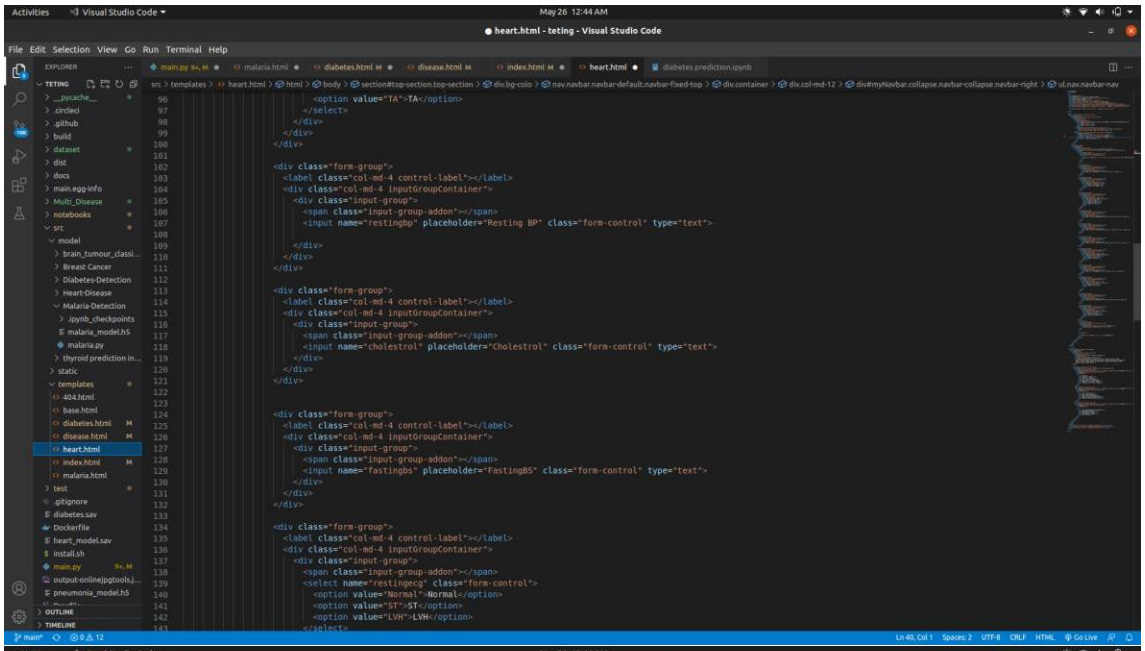
<input name="Insulin" placeholder="Insulin" class="form-control" type="text">
<input name="BMI" placeholder="BMI" class="form-control" type="text">
<input name="DiabetesPedigreeFunction" placeholder="Diabetes Pedigree Function"
class="form-control" type="text">
<input name="Age" placeholder="Age" class="form-control" type="text">
<!-- Button -->
<button type="submit" class="btn btn-warning">Predict<span
class="glyphicon glyphicon-send"></button>
Ln 40, Col 1
Spaces: 2
UTF-8
LF
HTML
Go Live

Activities
Visual Studio Code
May 26 12:42 AM
diabetes.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
<section id="about" class="section-padding">
<h2 class="head-title lg-line">Diabetes</h2>
<p class="sec-para">Diabetes is a disease that occurs when your blood glucose, also called blood sugar, is
too high. Blood glucose is your main source of energy and comes from the food you eat. Insulin, a hormone
made by the pancreas, helps glucose from food get into your cells to be used for energy.
<h3>Symptoms</h3>
1. Increase thirst
2. Increase hunger
3. Frequent urination
<h3>Prevention</h3>
 1. Aspirin
2. Glucose control
3. Blood pressure lowering
4. Maintain normal body weight
5. Maintain your blood sugar level
Ln 40, Col 1
Spaces: 2
UTF-8
LF
HTML
Go Live


Activities
Visual Studio Code
May 26 12:42 AM
diabetes.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > diabetes.html > html > body > section#top-section.top-section > div.bg-colo > nav.navbar.navbar-default.navbar-fixed-top > div.container > div.col-md-12 > div#myNavbar.collapse.navbar-collapse.navbar-right > ul.nav.navbar-nav
<h3>Prevention</h3>
 1. Aspirin
2. Glucose control
3. Blood pressure lowering
4. Maintain normal body weight
5. Maintain your blood sugar level
6. Keeps your body physically active
<h3>Treatment</h3>
1. Healthy diet
2. Physical exercise
<!-- Javascript section -->
<script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<script src="{{url_for('static',filename='js/bootstrap.min.js')}}"></script>
<script src="{{url_for('static',filename='js/custom.js')}}"></script>
</body>
</html>
OUTLINE
TIMELINE
main*
Ln 40, Col 1
Spaces: 2
UTF-8
LF
HTML
Go Live

Heart.html




Activities
Visual Studio Code
May 26 12:44 AM
heart.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py
malaria.html
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
src > templates > heart.html > html > body > section#top-section.top-section > div.bg-colo > nav.navbar.navbar-default.navbar-fixed-top > div.container > div.col-md-12 > div#myNavbar.collapse.navbar-collapse.navbar-right > ul.nav.navbar-nav
<option value="TA">TA</option>
</select>

<label class="col-md-4 control-label"></label>

<input name="restingbp" placeholder="Resting BP" class="form-control" type="text">

<label class="col-md-4 control-label"></label>

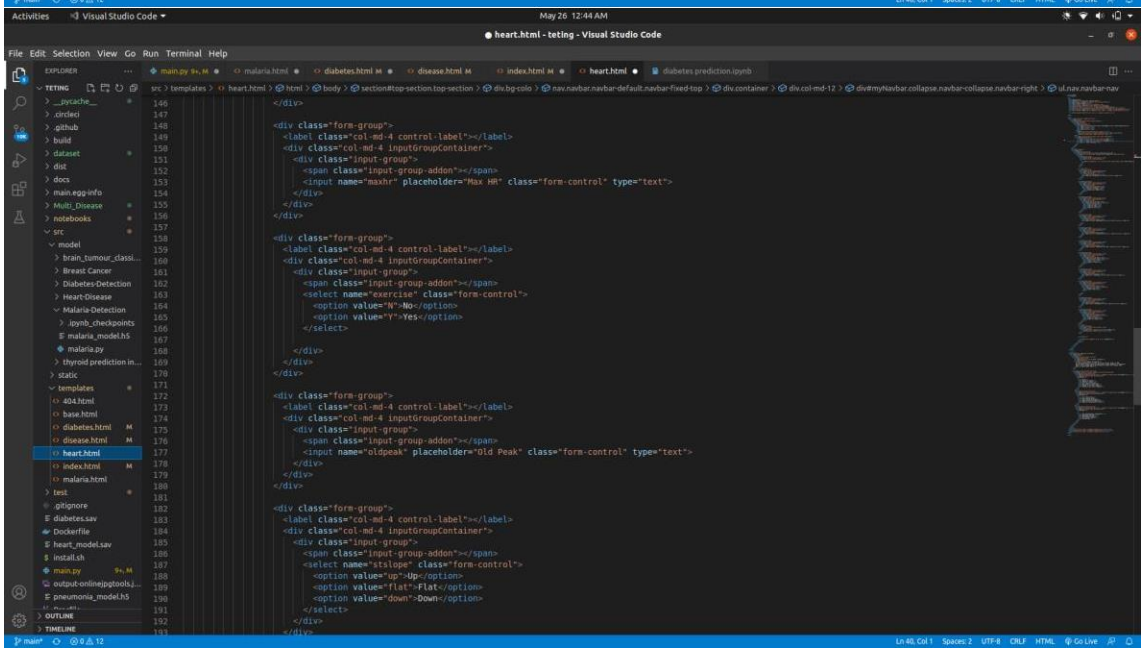
<input name="cholestrol" placeholder="Cholestrol" class="form-control" type="text">

<label class="col-md-4 control-label"></label>

<input name="fastingbs" placeholder="FastingBS" class="form-control" type="text">

<label class="col-md-4 control-label"></label>

<select name="restingecg" class="form-control">
<option value="Normal">Normal</option>
<option value="ST">ST</option>
<option value="LVH">LVH</option>
</select>


Activities
Visual Studio Code
May 26 12:44 AM
heart.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help

<label class="col-md-4 control-label"></label>

<input name="maxhr" placeholder="Max HR" class="form-control" type="text">

<label class="col-md-4 control-label"></label>

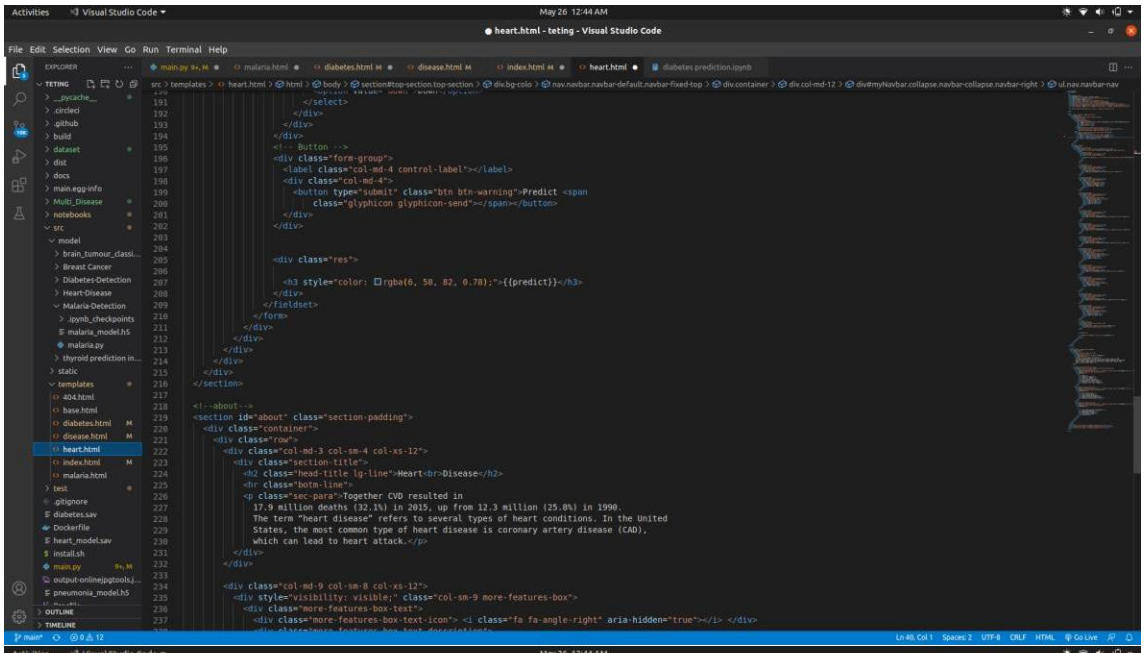
<select name="exercise" class="form-control">
<option value="N">No</option>
<option value="Y">Yes</option>
</select>

<label class="col-md-4 control-label"></label>

<input name="oldpeak" placeholder="Old Peak" class="form-control" type="text">

<label class="col-md-4 control-label"></label>

<select name="stslope" class="form-control">
<option value="up">Up</option>
<option value="flat">Flat</option>
<option value="down">Down</option>
</select>


Activities
Visual Studio Code
May 26 12:44 AM
heart.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py
malaria.html
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
src > templates > heart.html > html > body > section#top-section.top-section > div.bg-colo > nav.navbar.navbar-default.navbar-fixed-top > div.container > div.col-md-12 > div#myNavbar.collapse.navbar-collapse.navbar-right > ul.nav.navbar-nav
<select>
</select>

<!-- Button -->

<label class="col-md-4 control-label"></label>

<button type="submit" class="btn btn-warning">Predict <span
class="glyphicon glyphicon-send"></button>

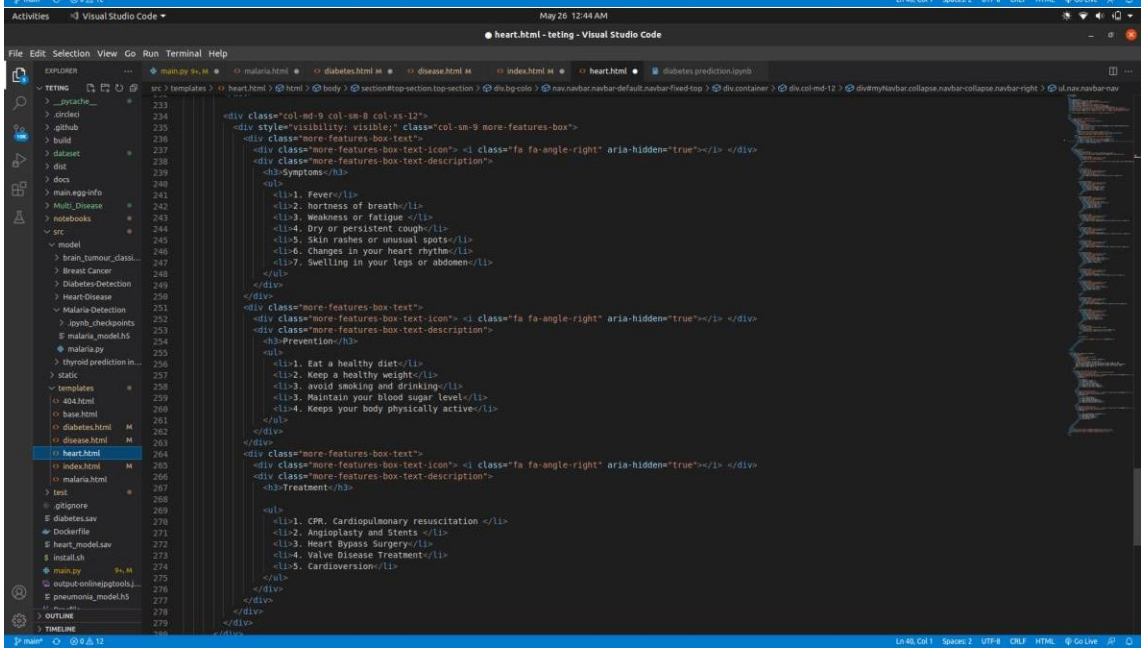
<h3 style="color: rgba(6, 58, 82, 0.78);">{{predict}}</h3>

</fieldset>
</form>

</section>
<!--about-->
<section id="about" class="section-padding">

<h2 class="head-title lg-line">Heart
Disease</h2>
<hr class="botm-line">
Together CVD resulted in
17.9 million deaths (32.1%) in 2015, up from 12.3 million (25.8%) in 1990.
The term "heart disease" refers to several types of heart conditions. In the United
States, the most common type of heart disease is coronary artery disease (CAD),
which can lead to heart attack.

Ln 40, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live


Activities
Visual Studio Code
May 26 12:44 AM
heart.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help

<h3>Symptoms</h3>

1. Fever
2. hortness of breath
3. Weakness or fatigue
4. Dry or persistent cough
5. Skin rashes or unusual spots
6. Changes in your heart rhythm
7. Swelling in your legs or abdomen

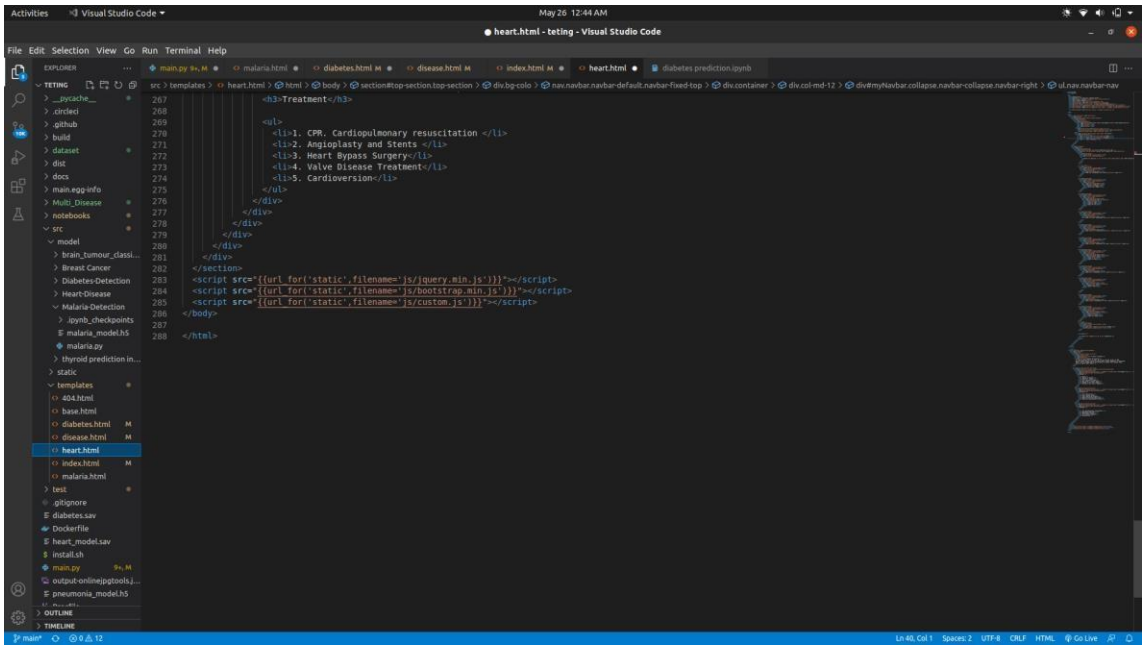
<h3>Prevention</h3>

1. Eat a healthy diet
2. Keep a healthy weight
3. avoid smoking and drinking
3. Maintain your blood sugar level
4. Keeps your body physically active

<h3>Treatment</h3>

1. CPR. Cardiopulmonary resuscitation
2. Angioplasty and Stents
3. Heart Bypass Surgery
4. Valve Disease Treatment
5. Cardioversion

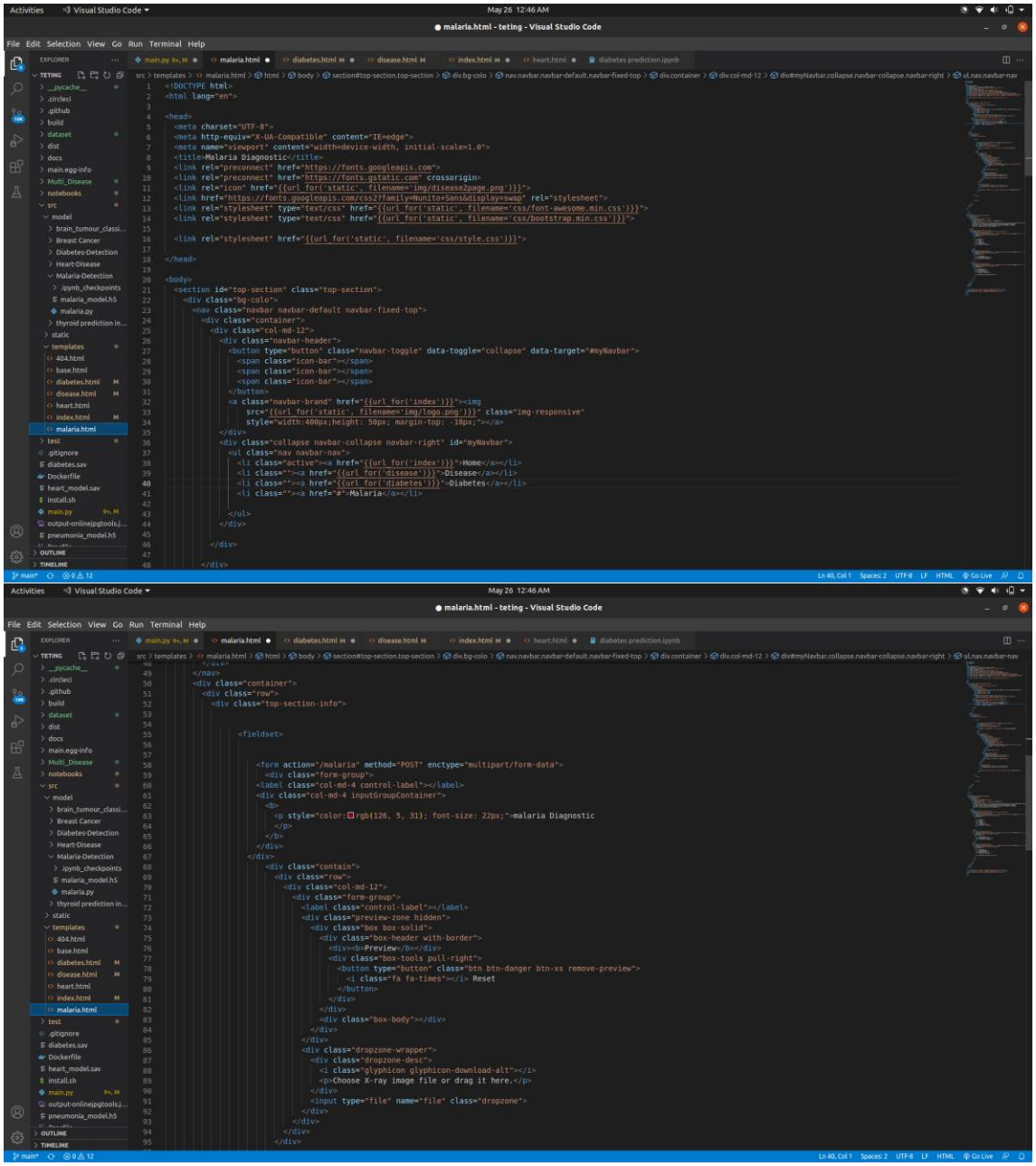
Ln 40, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live

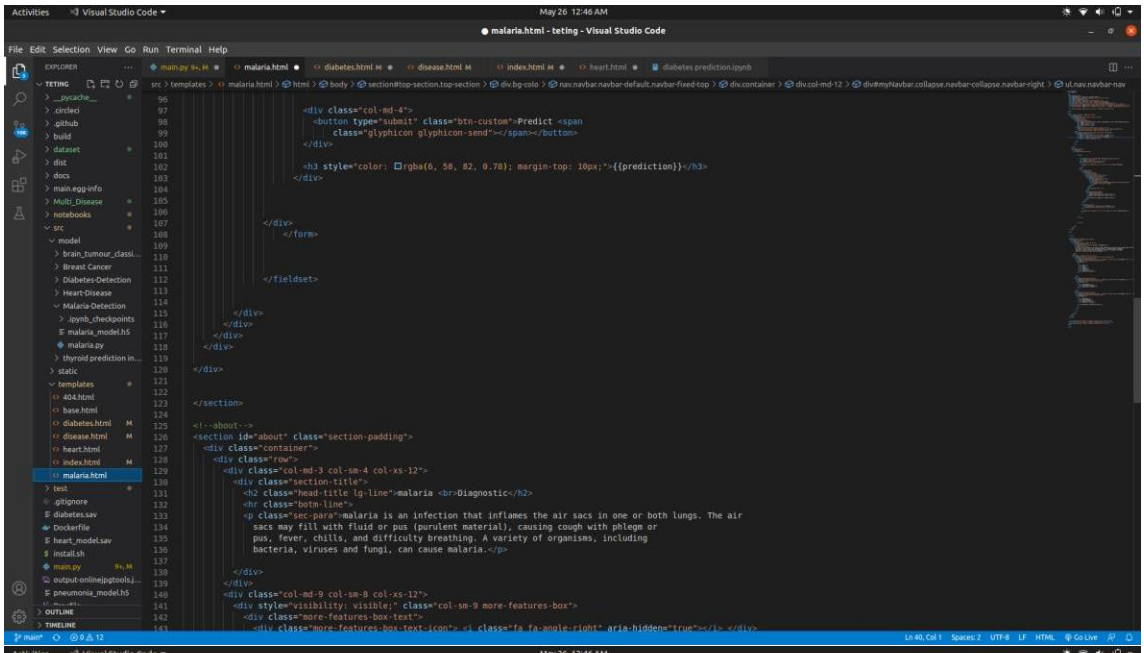

Activities
Visual Studio Code
May 26 12:44 AM
heart.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > heart.html > html > body > section#top-section.top-section > div.bg-colo > nav.navbar.navbar-default.navbar-fixed-top > div.container > div.col-md-12 > div#myNavbar.collapse.navbar-collapse.navbar-right > ul.nav.navbar-nav
<h3>Treatment</h3>

1. CPR. Cardiopulmonary resuscitation
2. Angioplasty and Stents
3. Heart Bypass Surgery
4. Valve Disease Treatment
5. Cardioversion

</section>
<script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<script src="{{url_for('static',filename='js/bootstrap.min.js')}}"></script>
<script src="{{url_for('static',filename='js/custom.js')}}"></script>
</body>
</html>
Ln 40, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live

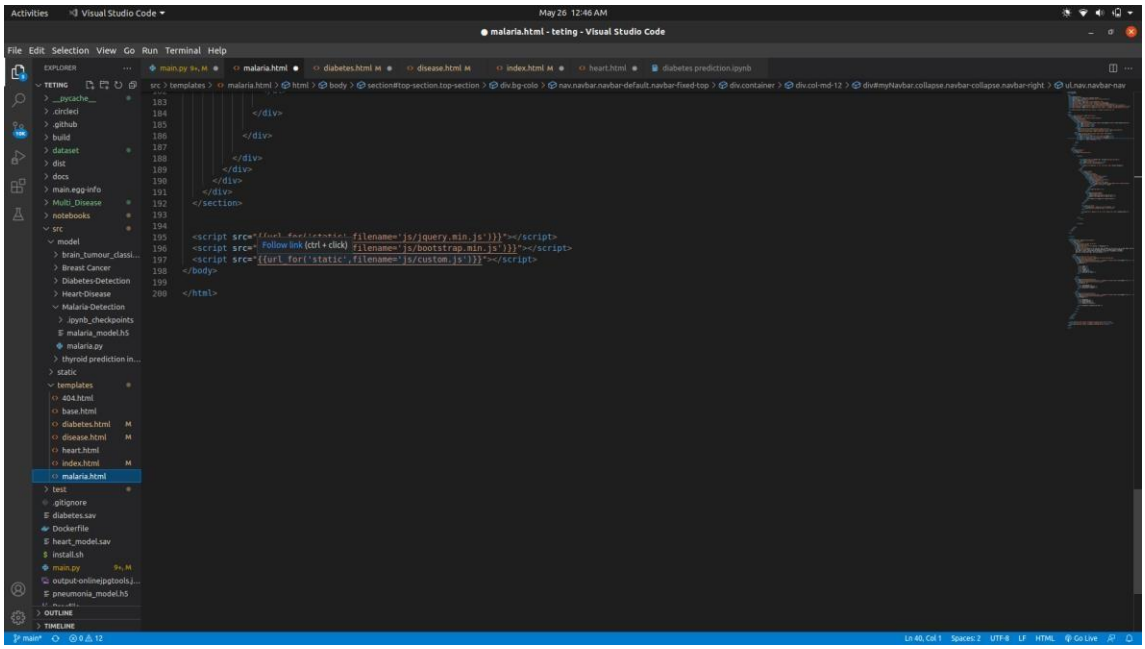
Malaria.html




Activities
Visual Studio Code
May 26 12:46 AM
malaria.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py
malaria.html
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
<button type="submit" class="btn-custom">Predict <span
class="glyphicon glyphicon-send"></button>

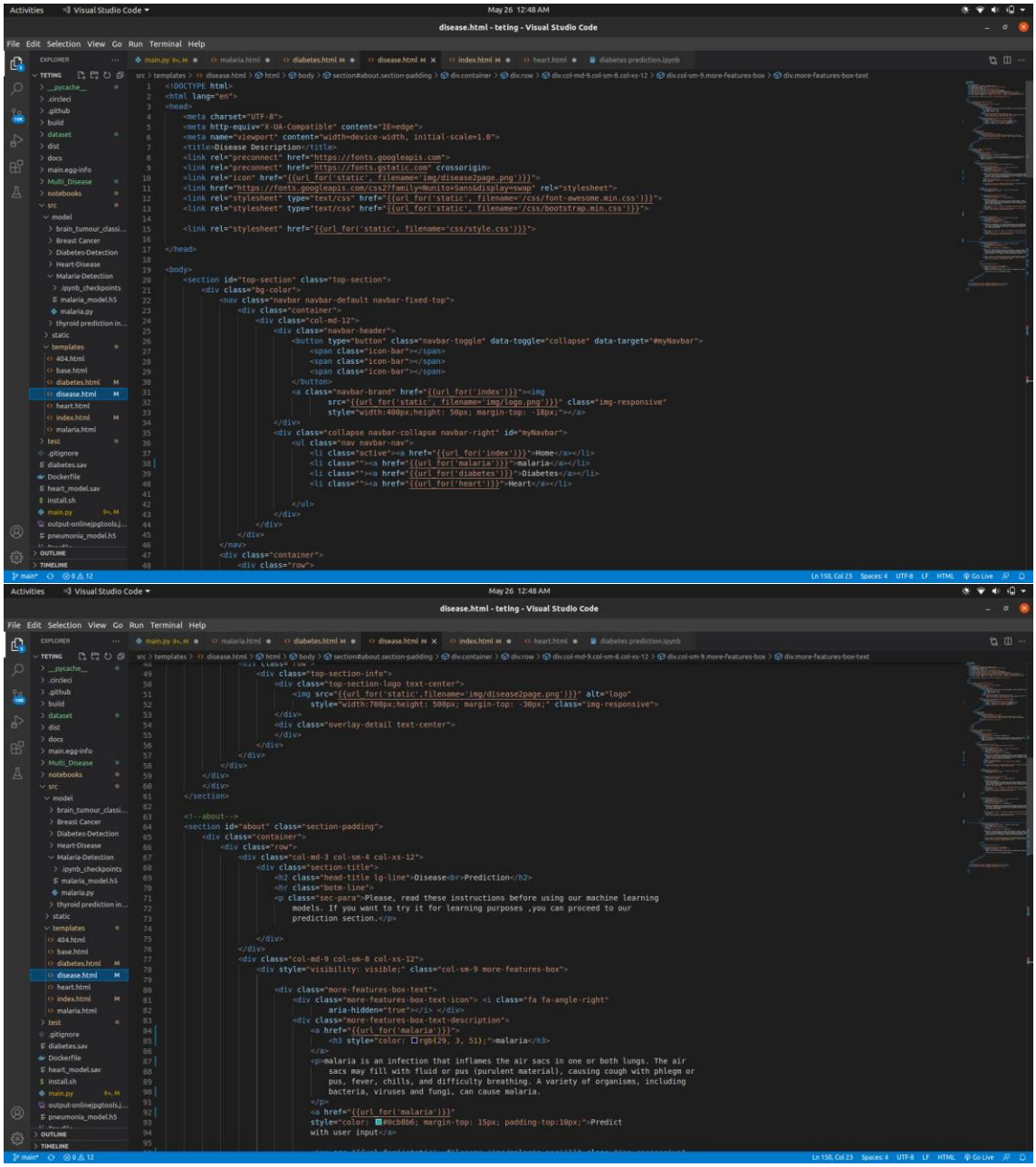
<h3 style="color: rgba(6, 58, 82, 0.78); margin-top: 10px;">{{prediction}}</h3>
</form>
<!--about-->
<section id="about" class="section-padding">
<h2 class="head-title lg-line">malaria
Diagnostic</h2>
<hr class="botm-line">
malaria is an infection that inflames the air sacs in one or both lungs. The air
sacs may fill with fluid or pus (purulent material), causing cough with phlegm or
pus, fever, chills, and difficulty breathing. A variety of organisms, including
bacteria, viruses and fungi, can cause malaria.

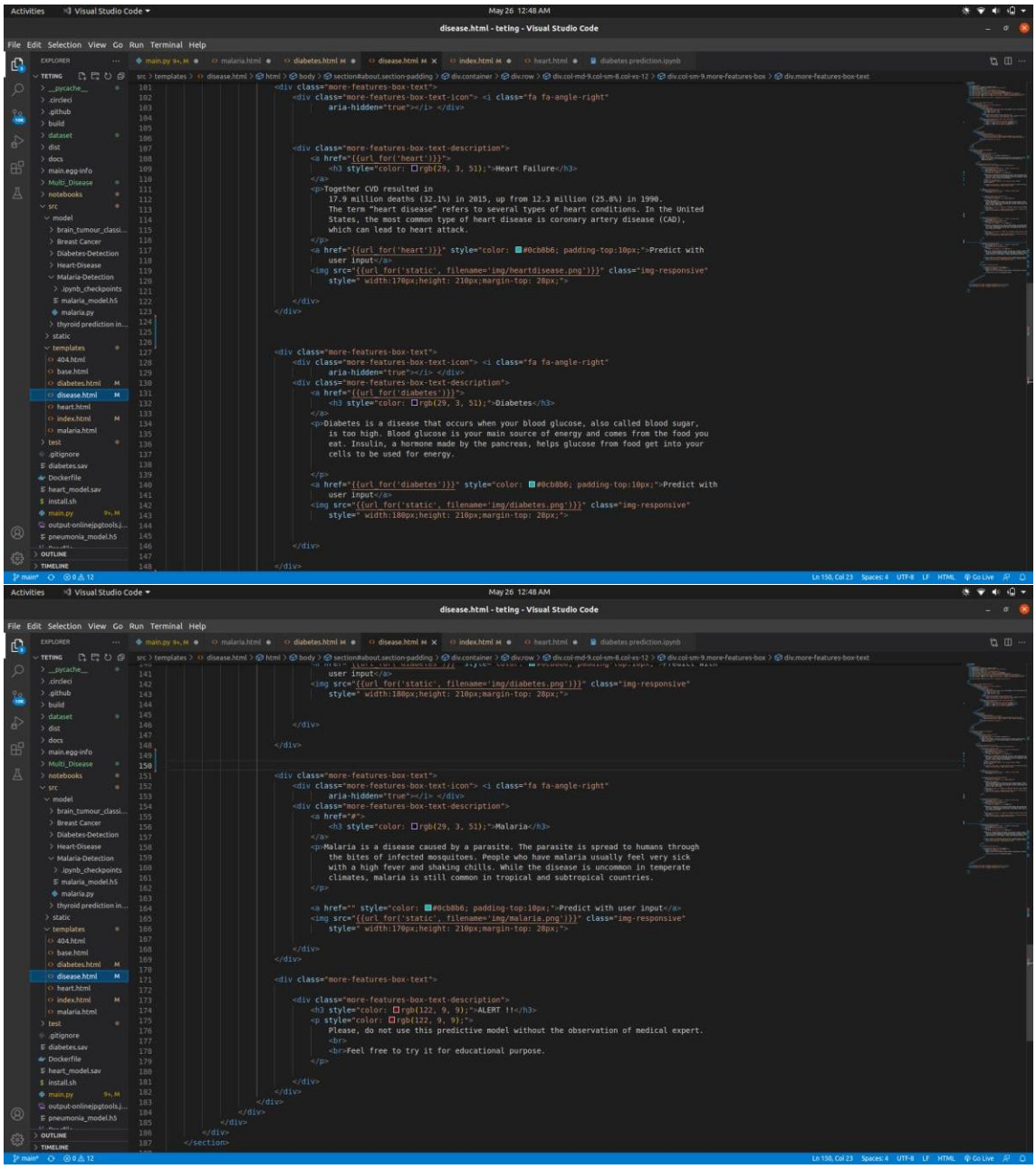
<h3>Symptoms</h3>
1. Cough
2. Fever
3. Fatigue
 4. Sputum
 5. chest pain
6. Shortness of breath
<h3>Prevention</h3>
 1. Vaccination
2. Environmental measure
3. Appropriate treatment
<h3>Treatment</h3>
1. Medication
2. OTC Medication
3. Oxygen therapy
4. Respiratory therapy
5. Antibiotics injected into vein


Activities
Visual Studio Code
May 26 12:46 AM
malaria.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > malaria.html > html > body > section#top-section.top-section > div.bg-colo > nav.navbar.navbar-default.navbar-fixed-top > div.container > div.col-md-12 > div#myNavbar.collapse.navbar-collapse.navbar-right > ul.nav.navbar-nav

</section>
<script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<script src="{{url_for('static',filename='js/bootstrap.min.js')}}"></script>
<script src="{{url_for('static',filename='js/custom.js')}}"></script>
</body>
</html>
Follow link (ctrl + click)
model
brain_tumour_classi...
Breast Cancer
Diabetes-Detection
Heart-Disease
Malaria-Detection
.ipynb_checkpoints
malaria_model.h5
malaria.py
thyroid prediction in...
static
templates
404.html
base.html
diabetes.html
disease.html
heart.html
index.html
malaria.html
test
.gitignore
diabetes.sav
Dockerfile
heart_model.sav
install.sh
main.py
output-onlinejpgtools.j...
pneumonia_model.h5
OUTLINE
TIMELINE
main*
Ln 40, Col 1
Spaces: 2
UTF-8
LF
HTML
Go Live

Disease.html




Activities
Visual Studio Code
May 26 12:48 AM
disease.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > disease.html > html > body > section#about.section-padding > div.container > div.row > div.col-md-9.col-sm-8.col-xs-12 > div.col-sm-9.more-features-box > div.more-features-box-text

 <i class="fa fa-angle-right"
aria-hidden="true"></i>

<h3 style="color: rgb(29, 3, 51);">Heart Failure</h3>

Together CVD resulted in
17.9 million deaths (32.1%) in 2015, up from 12.3 million (25.8%) in 1990.
The term “heart disease” refers to several types of heart conditions. In the United
States, the most common type of heart disease is coronary artery disease (CAD),
which can lead to heart attack.

Predict with
user input
<img src="{{url_for('static', filename='img/heartdisease.png')}}" class="img-responsive"
style=" width:170px;height: 210px;margin-top: 28px;">

 <i class="fa fa-angle-right"
aria-hidden="true"></i>

<h3 style="color: rgb(29, 3, 51);">Diabetes</h3>

Diabetes is a disease that occurs when your blood glucose, also called blood sugar,
is too high. Blood glucose is your main source of energy and comes from the food you
eat. Insulin, a hormone made by the pancreas, helps glucose from food get into your
cells to be used for energy.

Predict with
user input
<img src="{{url_for('static', filename='img/diabetes.png')}}" class="img-responsive"
style=" width:180px;height: 210px;margin-top: 28px;">

Ln 150, Col 23 Spaces: 4 UTF-8 LF HTML Go Live
Activities
Visual Studio Code
May 26 12:48 AM
disease.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
user input
<img src="{{url_for('static', filename='img/diabetes.png')}}" class="img-responsive"
style=" width:180px;height: 210px;margin-top: 28px;">

 <i class="fa fa-angle-right"
aria-hidden="true"></i>

<h3 style="color: rgb(29, 3, 51);">Malaria</h3>

Malaria is a disease caused by a parasite. The parasite is spread to humans through
the bites of infected mosquitoes. People who have malaria usually feel very sick
with a high fever and shaking chills. While the disease is uncommon in temperate
climates, malaria is still common in tropical and subtropical countries.

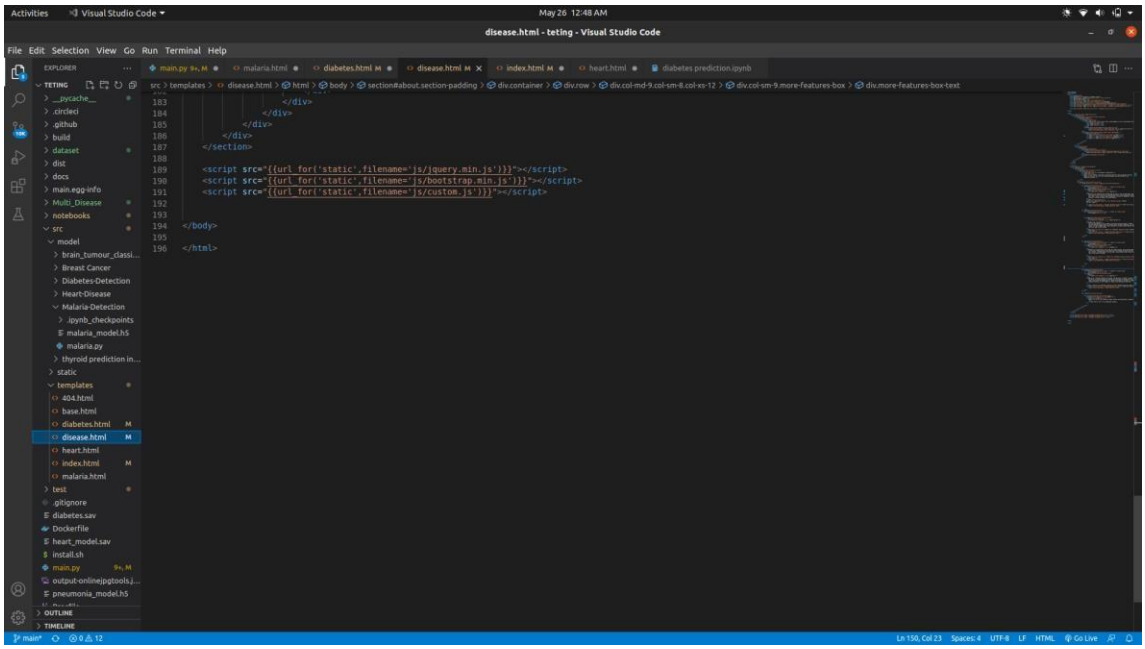
Predict with user input
<img src="{{url_for('static', filename='img/malaria.png')}}" class="img-responsive"
style=" width:170px;height: 210px;margin-top: 28px;">

<h3 style="color: rgb(122, 9, 9);">ALERT !!</h3>

Please, do not use this predictive model without the observation of medical expert.

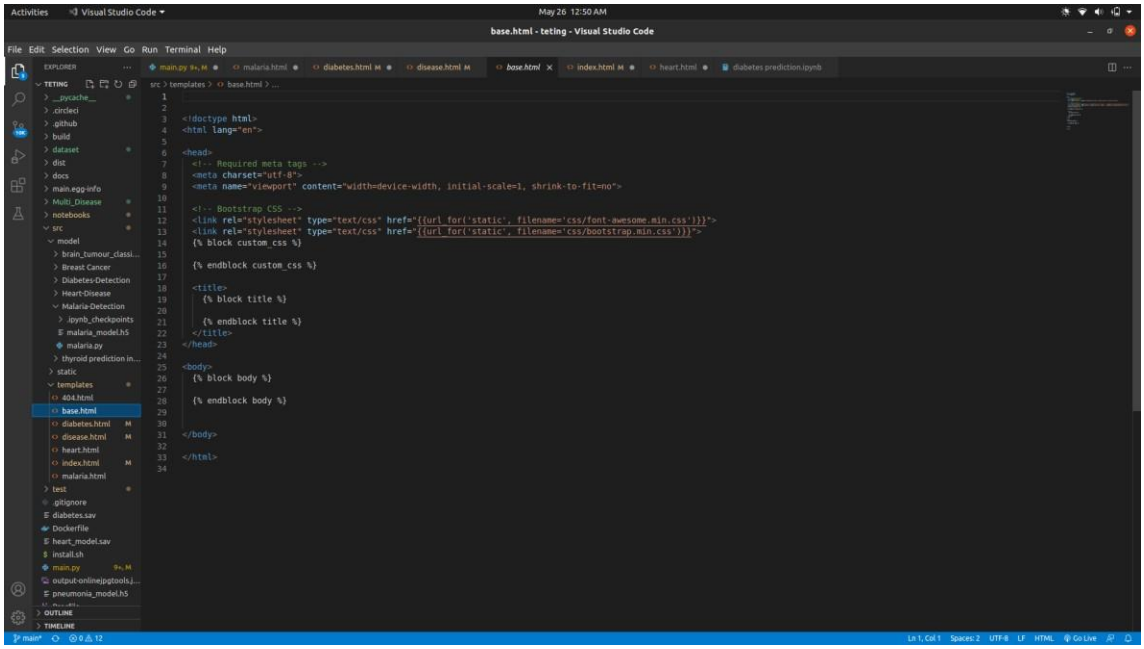
Feel free to try it for educational purpose.

</section>
Ln 150, Col 23 Spaces: 4 UTF-8 LF HTML Go Live

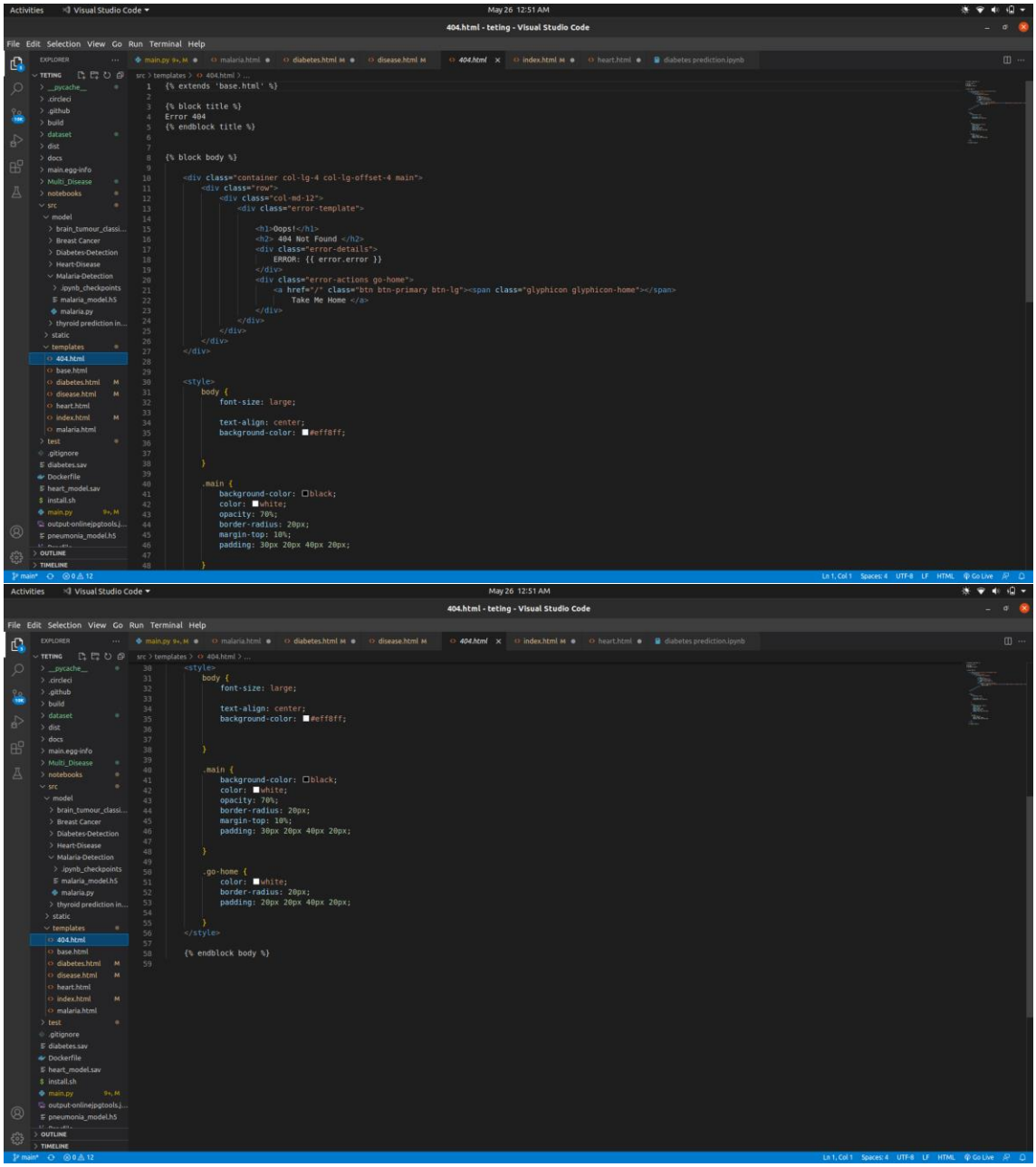

Activities
Visual Studio Code
May 26 12:48 AM
disease.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
src > templates > disease.html > html > body > section#about.section-padding > div.container > div.row > div.col-md-9.col-sm-8.col-xs-12 > div.col-sm-9.more-features-box > div.more-features-box-text

</section>
<script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<script src="{{url_for('static',filename='js/bootstrap.min.js')}}"></script>
<script src="{{url_for('static',filename='js/custom.js')}}"></script>
</body>
</html>
__pycache__
.circleci
.github
build
dataset
dist
docs
main.egg-info
Multi_Disease
notebooks
src
model
brain_tumour_classi...
Breast Cancer
Diabetes-Detection
Heart-Disease
Malaria-Detection
.ipynb_checkpoints
malaria_model.h5
malaria.py
thyroid prediction in...
static
templates
404.html
base.html
diabetes.html M
disease.html M
heart.html
index.html M
malaria.html
test
.gitignore
diabetes.sav
Dockerfile
heart_model.sav
install.sh
main.py 9+, M
output-onlinejpgtools.j...
pneumonia_model.h5
OUTLINE
TIMELINE
main*
Ln 150, Col 23
Spaces: 4
UTF-8
LF
HTML
Go Live

Base.html


Activities
Visual Studio Code
May 26 12:50 AM
base.html - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
src > templates > base.html > ...
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<!-- Bootstrap CSS -->
<link rel="stylesheet" type="text/css" href="{{url_for('static', filename='css/font-awesome.min.css')}}">
<link rel="stylesheet" type="text/css" href="{{url_for('static', filename='css/bootstrap.min.css')}}">
{% block custom_css %}
{% endblock custom_css %}
<title>
{% block title %}
{% endblock title %}
</title>
</head>
<body>
{% block body %}
{% endblock body %}
</body>
</html>
OUTLINE
TIMELINE

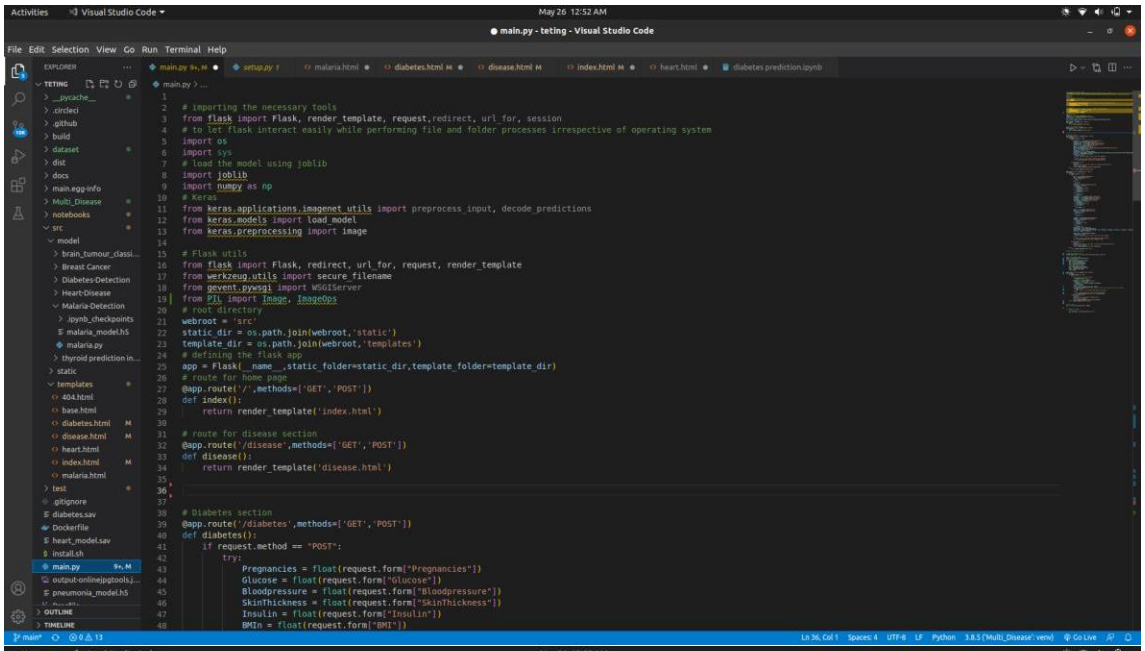
404.html

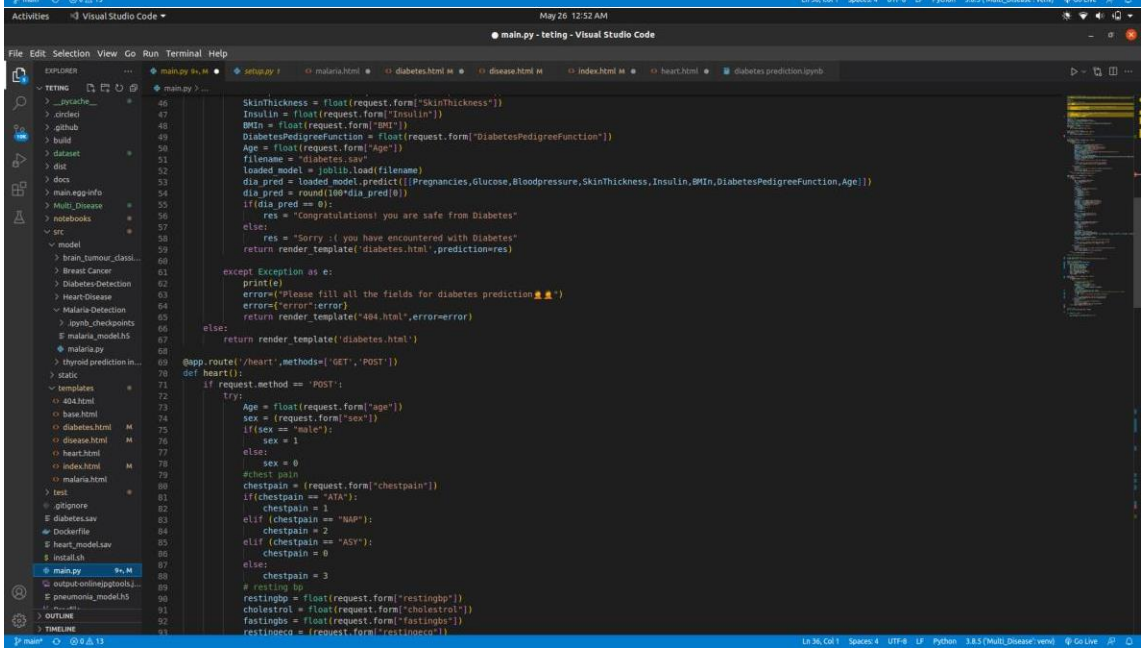


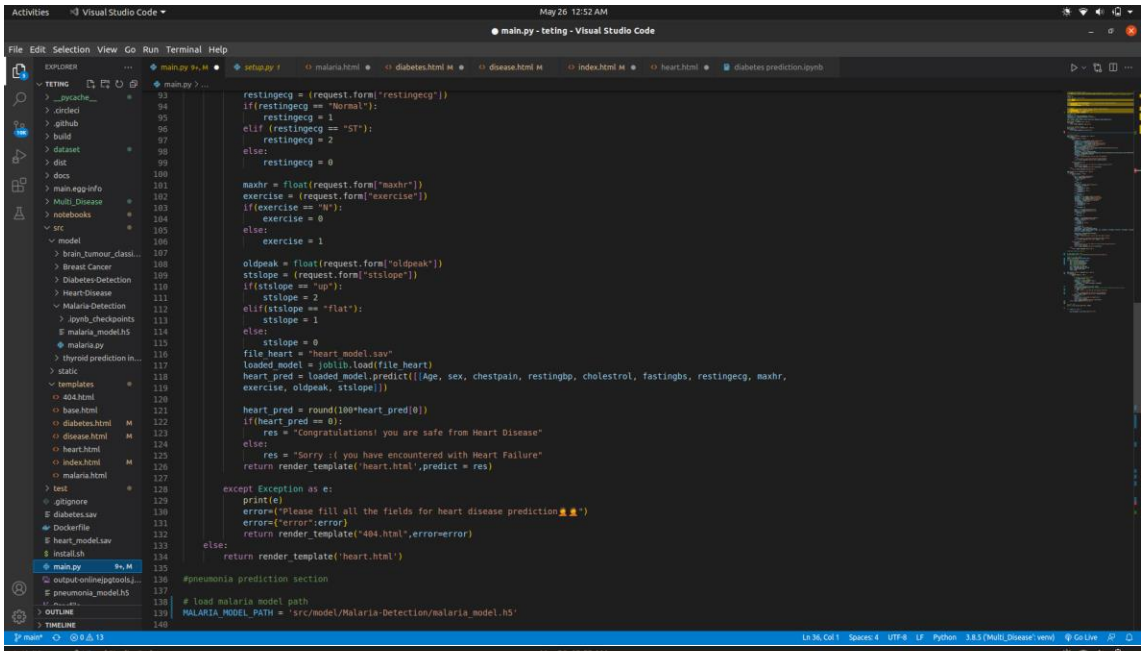
```
{% extends 'base.html' %}

{% block title %}
Error 404
{% endblock title %}


{% block body %}

    <div class="container col-lg-4 col-lg-offset-4 main">
        <div class="row">
            <div class="col-md-12">
                <div class="error-template">

                    <h1>Oops!</h1>
                    <h2> 404 Not Found </h2>
                    <div class="error-details">
                        ERROR: {{ error.error }}
                    </div>
                    <div class="error-actions go-home">
                        <a href="/" class="btn btn-primary btn-lg"><span class="glyphicon glyphicon-home"></span>
                            Take Me Home </a>
                    </div>
                </div>
            </div>
        </div>
    </div>


    <style>
        body {
            font-size: large;

            text-align: center;
            background-color: #eff8ff;


        }

        .main {
            background-color: black;
            color: white;
            opacity: 70%;
            border-radius: 20px;
            margin-top: 10%;
            padding: 30px 20px 40px 20px;
        }
```



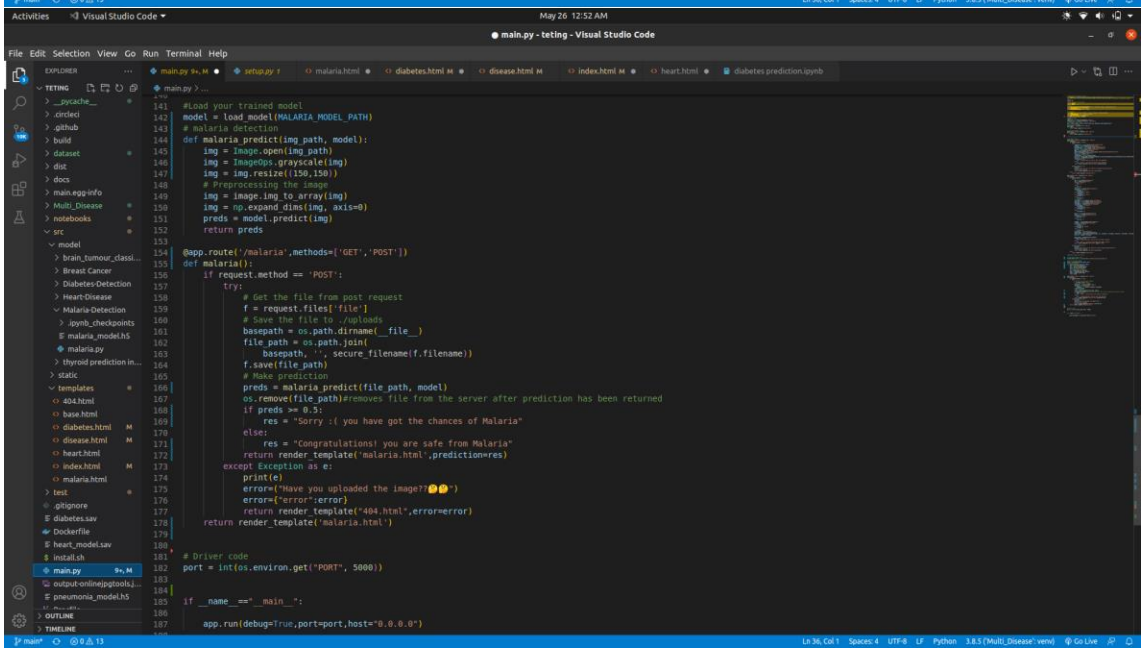
```
    <style>
        body {
            font-size: large;

            text-align: center;
            background-color: #eff8ff;


        }

        .main {
            background-color: black;
            color: white;
            opacity: 70%;
            border-radius: 20px;
            margin-top: 10%;
            padding: 30px 20px 40px 20px;
        }

        .go-home {
            color: white;
            border-radius: 20px;
            padding: 20px 20px 40px 20px;
        }
    </style>

    {% endblock body %}
```


Activities
Visual Studio Code
May 26 12:52 AM
main.py - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py
setup.py
malaria.html
diabetes.html
disease.html
index.html
heart.html
diabetes prediction.ipynb
# importing the necessary tools
from flask import Flask, render_template, request,redirect, url_for, session
# to let flask interact easily while performing file and folder processes irrespective of operating system
import os
import sys
# load the model using joblib
import joblib
import numpy as np
# Keras
from keras.applications.imagenet_utils import preprocess_input, decode_predictions
from keras.models import load_model
from keras.preprocessing import image
# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
from PIL import Image, ImageOps
# root directory
webroot = 'src'
static_dir = os.path.join(webroot,'static')
template_dir = os.path.join(webroot,'templates')
# defining the flask app
app = Flask(__name__,static_folder=static_dir,template_folder=template_dir)
# route for home page
@app.route('/',methods=['GET','POST'])
def index():
return render_template('index.html')
route for disease section
@app.route('/disease',methods=['GET','POST'])
def disease():
return render_template('disease.html')
Diabetes section
@app.route('/diabetes',methods=['GET','POST'])
def diabetes():
if request.method == "POST":
try:
Pregnancies = float(request.form["Pregnancies"])
Glucose = float(request.form["Glucose"])
Bloodpressure = float(request.form["Bloodpressure"])
SkinThickness = float(request.form["SkinThickness"])
Insulin = float(request.form["Insulin"])
BMIn = float(request.form["BMI"])
Ln 36, Col 1
Spaces: 4
UTF-8
LF
Python
3.8.5 ('Multi_Disease': venv)
Go Live

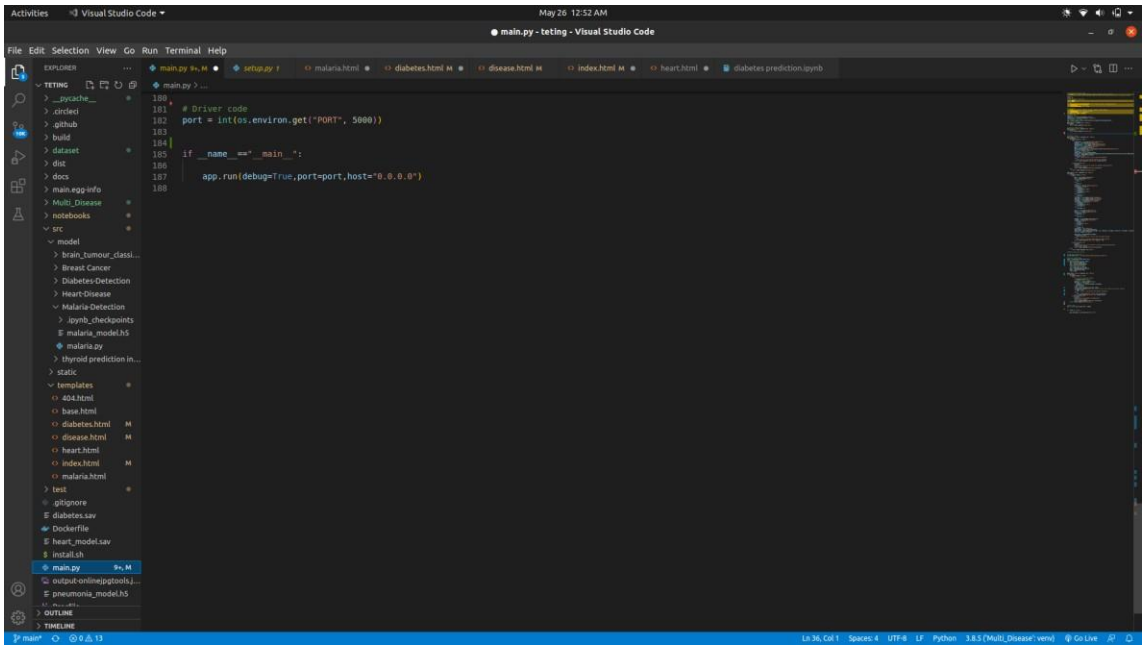

Activities
Visual Studio Code
May 26 12:52 AM
main.py - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
SkinThickness = float(request.form["SkinThickness"])
Insulin = float(request.form["Insulin"])
BMIn = float(request.form["BMI"])
DiabetesPedigreeFunction = float(request.form["DiabetesPedigreeFunction"])
Age = float(request.form["Age"])
filename = "diabetes.sav"
loaded_model = joblib.load(filename)
dia_pred = loaded_model.predict([[Pregnancies,Glucose,Bloodpressure,SkinThickness,Insulin,BMIn,DiabetesPedigreeFunction,Age]])
dia_pred = round(100*dia_pred[0])
if(dia_pred == 0):
res = "Congratulations! you are safe from Diabetes"
else:
res = "Sorry :(you have encountered with Diabetes"
return render_template('diabetes.html',prediction=res)
except Exception as e:
print(e)
error={"Please fill all the fields for diabetes prediction"}
error={"error":error}
return render_template("404.html",error=error)
else:
return render_template('diabetes.html')
@app.route('/heart',methods=['GET','POST'])
def heart():
if request.method == 'POST':
try:
Age = float(request.form["age"])
sex = (request.form["sex"])
if(sex == "male"):
sex = 1
else:
sex = 0
#chest pain
chestpain = (request.form["chestpain"])
if(chestpain == "ATA"):
chestpain = 1
elif (chestpain == "NAP"):
chestpain = 2
elif (chestpain == "ASY"):
chestpain = 0
else:
chestpain = 3
# resting bp
restingbp = float(request.form["restingbp"])
cholestrol = float(request.form["cholestrol"])
fastingbs = float(request.form["fastingbs"])
restingecg = (request.form["restingecg"])
Ln 36, Col 1
Spaces: 4
UTF-8
LF
Python
3.8.5 ('Multi_Disease': venv)
Go Live



```
            restingecg = (request.form["restingecg"])
            if(restingecg == "Normal"):
                restingecg = 1
            elif (restingecg == "ST"):
                restingecg = 2
            else:
                restingecg = 0

            maxhr = float(request.form["maxhr"])
            exercise = (request.form["exercise"])
            if(exercise == "N"):
                exercise = 0
            else:
                exercise = 1

            oldpeak = float(request.form["oldpeak"])
            stslope = (request.form["stslope"])
            if(stslope == "up"):
                stslope = 2
            elif(stslope == "flat"):
                stslope = 1
            else:
                stslope = 0
            file_heart = "heart_model.sav"
            loaded_model = joblib.load(file_heart)
            heart_pred = loaded_model.predict([[Age, sex, chestpain, restingbp, cholestrol, fastingbs, restingecg, maxhr,
            exercise, oldpeak, stslope]])

            heart_pred = round(100*heart_pred[0])
            if(heart_pred == 0):
                res = "Congratulations! you are safe from Heart Disease"
            else:
                res = "Sorry :( you have encountered with Heart Failure"
            return render_template('heart.html',predict = res)

        except Exception as e:
            print(e)
            error={"Please fill all the fields for heart disease prediction🤦‍♂️"}
            error={"error":error}
            return render_template("404.html",error=error)
    else:
        return render_template('heart.html')

#pneumonia prediction section

# load malaria model path
MALARIA_MODEL_PATH = 'src/model/Malaria-Detection/malaria_model.h5'
```



```
#Load your trained model
model = load_model(MALARIA_MODEL_PATH)
# malaria detection
def malaria_predict(img_path, model):
    img = Image.open(img_path)
    img = ImageOps.grayscale(img)
    img = img.resize((150,150))
    # Preprocessing the image
    img = image.img_to_array(img)
    img = np.expand_dims(img, axis=0)
    preds = model.predict(img)
    return preds

@app.route('/malaria',methods=['GET','POST'])
def malaria():
    if request.method == 'POST':
        try:
            # Get the file from post request
            f = request.files['file']
            # Save the file to ./uploads
            basepath = os.path.dirname(__file__)
            file_path = os.path.join(
                basepath, '', secure_filename(f.filename))
            f.save(file_path)
            # Make prediction
            preds = malaria_predict(file_path, model)
            os.remove(file_path)#removes file from the server after prediction has been returned
            if preds >= 0.5:
                res = "Sorry :( you have got the chances of Malaria"
            else:
                res = "Congratulations! you are safe from Malaria"
            return render_template('malaria.html',prediction=res)
        except Exception as e:
            print(e)
            error={"Have you uploaded the image?🤔🤔"}
            error={"error":error}
            return render_template("404.html",error=error)
    return render_template('malaria.html')


# Driver code
port = int(os.environ.get("PORT", 5000))

if __name__=="__main__":

    app.run(debug=True,port=port,host="0.0.0.0")
```


Activities
Visual Studio Code
May 26 12:52 AM
main.py - teting - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
TETING
main.py 9+, M
setup.py 1
malaria.html
diabetes.html M
disease.html M
index.html M
heart.html
diabetes prediction.ipynb
main.py > ...
180
181 # Driver code
182 port = int(os.environ.get("PORT", 5000))
183
184
185 if __name__=="__main__":
186
187 app.run(debug=True,port=port,host="0.0.0.0")
188
__pycache__
.circleci
.github
build
dataset
dist
docs
main.egg-info
Multi_Disease
notebooks
src
model
brain_tumour_classi...
Breast Cancer
Diabetes-Detection
Heart-Disease
Malaria-Detection
.ipynb_checkpoints
malaria_model.h5
malaria.py
thyroid prediction in...
static
templates
404.html
base.html
diabetes.html M
disease.html M
heart.html
index.html M
malaria.html
test
.gitignore
diabetes.sav
Dockerfile
heart_model.sav
install.sh
main.py 9+, M
output-onlinejpgtools.j...
pneumonia_model.h5
OUTLINE
TIMELINE
main*
0 13
Ln 36, Col 1
Spaces: 4
UTF-8
LF
Python
3.8.5 ('Multi_Disease': venv)
Go Live