“Smart health prediction system”

Developed using SVM and random forest algorithms

Author: Roshan Rai

Email Id - roshanraidv@gmail.com

ABSTRACT
Smart health prediction system

The world in which we currently live is one in which science and technology are enlarging accessibility, proximity, and more informative. These days, science and technology have an impact on every sector, and as a result, we can observe growth and development in every sector. The medical sector is one of the most significant and delicate topics in science and technology, aside from all other fields. We can see that numerous medical researchers and experts are attempting to digitize medical services as a result extremely complex health issues have become possible thanks to technological advancements Online health consultants, digital eyes diagnoses, security and patient care with IoT medical devices, Pinpoint rogue devices etc. are the wisdom of the technology.

However, such a technological advancements, many individuals around the world are still unable to access fully dependable and correct health care, as a result they have to loss their life before their time. Even though many hospitals are well-equipped to give patients the best medical care possible, some of them nevertheless fall short in some areas. People are still far from receiving appropriate medical care because of a lack of trained medical personnel, inadequate equipment, a shortage of hospital space, poor doctor-patient communication, and poorly kept medical records. As a result, this project considers the smart health prediction system, which is based on data mining, to be the ideal method for facilitating an improved healthcare system. This project believes that, if a model is built with correct data and a suitable algorithm, it can be used to anticipate ailments, allowing users to become aware of them before they manifest physically. The goal of the project is to create an intelligent system for disease prediction based on user-provided symptoms. In order to build the system, SVM and random forest are employed as a combined model in this project.

Acknowledgements

To begin, I'd want to express my appreciation to my institution for offering an outstanding platform and adequate environment. I want to thank Dr. Tatiana Simmonds for giving opportunity to work on this topic "Smart health prediction system" .Additionally, I'd want to convey my appreciation to Dr. Konstantinos Skindilias, the supervisor of my project for guiding me from the beginning to the end of the project. His guidance has been really fruitful as a result I am able to submit my project at given time frame. Through this research, I have gained a thorough understanding of many aspects of how AI may benefit health care as well as the potential power of data.

In a similar vein, I'd want to express my gratitude to the friends and medical professionals I contacted regarding system effectiveness.

Contents

Tables of figure

Tables of Tables

CHAPTER 1 – INTRODUCTION

1.1. Project description

The project ‘Smart health prediction system’ is an intelligent prediction system developed based on the data mining. Data mining technique is used in this project to find the patterns and correlation from a large data sets in order to produce significant information that can be used to address issues and predict the course of diseases based just on their initial symptoms. The dataset is gathered from kaggle.com, and various machine learning approaches are evaluated and deployed in order to develop an effective model that can predict diseases accurately. This report contains precise details regarding the system's development. After conducting extensive research on the problem domain, requirements, comparable systems, methodology, and algorithms, the appropriate approach and algorithms are selected.

The project provided an online, intelligent, smart health prediction system that allows users to input their symptoms and makes possible disease predictions based on those symptoms. This project gives users the ability to anticipate changes in their health early on, which may serve as early warning signs for chronic diseases, improve patient diagnosis as well as bring in more patients for doctors.

1.2. Current scenario

The majority of the complex diseases in the field of medicine have been solved with the use of smart technology like artificial intelligence and machine learning. Health services are now more accessible, effective, efficient, and accessible. The analysis, presentation, and comprehension of complex medical and healthcare data can now be emulated by artificial intelligence in the healthcare industry.

However, as the human population is expanding at such a rapid rate, the realm of health care is facing an increasing number of challenges of a wide variety that are becoming more widespread. All of the current health issues cannot be remedied by the available health care provider. It may have occurred more than once for someone who needs emergency medical care but is unable to get it for any reason. People have to wait a significant amount of time before they can schedule an appointment with a doctor since there are not enough medical services. People are often need to wait in a long queue in order to receive basic medical services, which poses a risk to their health in addition to wasting valuable time. This problem can be handled by implementing an intelligent health disease prediction system, which will address the deficiencies of the existing framework.

Users will be able to receive online consulting and help from the project's developers. This project has been opened in order to develop more features that will provide users with the ability to obtain prompt advice on their health difficulties through the usage of an online intelligent health care system. Disorders and symptoms that are relevant to the system as a whole are entered into the system together with other symptoms. Users have the ability to communicate problems and symptoms with the system. After that, it analyses the user's symptoms to see if there are any illnesses that could be associated to those symptoms. A medical practitioner (the Doctor) is able to access the system, read patient data and reports, and make decisions based on the information they find there. When the doctor examines the particulars of the patient's search, they do so with the presumption that this will allow them to figure out what it is that the patient was seeking for. Administrators are able to update the database with fresh disease information by using the type of disease as well as the symptoms associated with it.

1.3. Problems areas

The purpose of the project is to develop software that is of a generic nature and may be deployed by any healthcare company. On the other hand, the success or failure of a project is dependent on a variety of different elements. For the project, a database that is both comprehensive and dependable of diseases and the symptoms they cause is required. In order to build a reliable prediction model, it is necessary to select the appropriate method, in addition to gathering the necessary data. In a manner comparable to this, the potential for difficulty in a project is increased when there is insufficient trustworthy data. In addition, failure of the project is a possibility if the wrong algorithm is used. Because the quality of the project is dependent on the data, it is essential to obtain data from trustworthy sources as well as the medical business in order to manage the risk aspect. After that, the data are examined and analysed by a trained medical practitioner to ensure that they are accurate. Following an adequate amount of research into the many different algorithms and the development of a prototype system utilising the algorithms, the algorithms that provide the greatest degree of precision will be selected and utilised in the project.

1.4. Aim

The project aims to build an efficient health diseases prediction system based on the symptoms selected by the users.

1.5. Objective

To begin, there is a certain goal that needs to be accomplished before a project can be considered successful. To accomplish the primary purpose of the project, having clear objectives is of the

utmost importance. The following is a list of the objectives of the project that should be accomplished to meet the primary goal: -

- Study the feasibility of the project and gather the requirement for the project.
- Concise research on a project that is comparable, looking through statistical data, official records, and other sources.
- In order to effectively manage both the time and the tasks, prepare a Gantt chart and a WBS.
- Choosing the Appropriate Form of Technology (programming languages, library, and platform).
- One of the following tasks is to collect the data that will be used in the preparation of the dataset.
- Conducting research on a variety of algorithms and choosing algorithms that are acceptable.
- Build the model ready for predictive analysis.

1.6. Structure of the report

1.6.1. Background

The background section of the report includes information about the end-users, literature reviews, considered algorithms, end-users, legal, social, ethical, professional aspects, technical aspects, approached taken, and comparison of similar projects. The detailed explanation of the background section will be mention in CHAPTER 2- BACKGROUND.

1.6.2. Development

The Development section of the report includes information about the different considered methodology, selected methodologies, code development process. The detailed explanation of the development section will be mention in CHAPTER 3 – Development.

1.6.3. Testing and analysis

The Testing and analysis section of the report includes information about the different test cases, testing types, and critical analysis. The detailed information about the testing and analysis section will be mention in CHAPTER 4: TESTING AND ANALYSIS.

1.6.4. Conclusion

The conclusion section of the report includes critical analysis, future work and personal evaluation, Also, this section includes the limitation and advantages of the project. The detailed explanation of the conclusion section will be mention in CHAPTER 5: CONCLUSION.

CHAPTER 2 – BACKGROUND

2.1. Literature review

Aditi Gavhane, Gouthami Kokkula, Isha Pandya, Kailas Devadkar have proposed a "Heart disease prediction system using machine learning" (Aditi Gavhane, 2019). They had collected relevant data pertaining of element to train the data as per the proposed algorithm to predict the patient heart disease. In this project sensor are used to measure the factors like heart beat rate and pressure that are easily available in watches, and cell phones. The parameter like age, sex, blood pressure, heart beat rate, diabetes, hyper cholesterol and body mass etc. are taken into the consideration and dataset is prepared for training. Similarly, author had used neural network algorithm multi-layer perceptron (MLP) to train and test the dataset in this project. The system will give the result as yes or no. If the system predict that the person has a heart diseases then the result will be obtained as yes and vice versa (Aditi Gavhane, 2019).

Similarly, Vikramaditya R. Jakkula, Diane J. Cook, Gaurav Jain have proposed a "Prediction Models for a smart Home Based Health Care System" (Vikramaditya R. Jakkula, 2007). In this paper, machine learning technique weka is used for predicting trends of health data over time and parallel Box Jenkins seasonal ARIMA model is also used for forecasting health data ranges over time. Training set containing 40 days of health parameter data collected from the inhabitant in the MavHome apartment is used. Firstly, support vector machine algorithm found in weka is used for predicting whether health trends are increasing, decreasing or constant. Secondly, automatic forecasting technique like phicast is used which allows Box- Jenkins method for forecasting the range of health data values. The data is divided into two groups, the first groups is used for test set whereas second group is used for calculating the accuracy of the constructed model and finally 3 fold cross validation is used to analysis the outcomes due to the size of the dataset (Vikramaditya R. Jakkula, 2007).

Likewise, s. Mohan, C. Thirumalai and G. Srivastava have proposed an "Effective heart disease prediction using Hybrid machine leaning technique" (S. Mohan, 2019) in which computational approach was used to find the factors of heart disease on the UCL Cleveland dataset. From the dataset important information is extracted like female have less chance for heart diseases compared to male. HRFLM which is a computational approach is used with ANN which enables back propagation associated with 13 features. The system is developed using R studio. ML process for developing this system starts from a pre-processing data followed by the features selection based on the DT entropy. Respectively, features selection and reduction process is performed where 13

attributes of the data set and two attributes pertaining to age and sex are used to identify the personal information on the patient. Clustering of datasets is done in classification of modelling using different algorithms (decision tress, language model, Support vector machine, random forest, naïve Bayes and the performance (accuracy, precision, error) of the classifier's are evaluated for error optimization.

2.1.1. Similar project Comparisons.

Similar project	Algorithms	Technology used	Parameter	Accuracy	GUI
Heart disease prediction system using machine learning	Neural network algorithm multi-layer perceptron (MLP)	Sensor is used for data collection	Age, sex, blood pressure, heart rate, diabetes, hyper cholesterol, body mass index(obesity)	91%	No
Prediction Models for a smart Home Based Health Care System	Support vector machine, ARIMA	Argus and X-Io networks (sensor) used for data collection.	Not mention	76%	No
Effective heart disease prediction using Hybrid machine leaning technique	Random forest and support vector machine.	Hybrid HRFLM approach	Age, Sex, Cp, Trestbps, Chol, FBS, Resting, Thali, Exang, Oldpeak, Slope, Ca, Thal ,Num	88%	No

Table 1 Comparison tables of similar project.

2.2. About the end-users

The system's primary audience is the general public, who like to ensure any potential ailments they may contract in the future. By selecting their symptoms, users can get an idea of their health status before making an appointment with a doctor or going to the emergency room. The user will be made aware that he or she has contracted one of a number of diseases depending on their symptoms, and if a registered doctor is located in close proximity to the user, the system will provide information about whom they should see.

Similarly, medical professional like doctors, researcher, and the medical organization are also the end-users of the system. Doctors are associated with the system through which patient can get consult as well as help from doctor's. In the system, doctors are organized into subsets based on the specialties in which they are trained. It's possible that a healthcare organization will be the system's ultimate end-user, in which case they'll be responsible for its implementation and day-to-day operation, as they'll have full administrative privileges and access to all of its functionality. The people who will benefit the most from this system are those who are concerned about their health and medical field.

2.3. Analysis of the system

2.3.1. Legal, Social, Ethical and Professional issues

Making a more accessible health care system through technology might not be going to harm anyone. Despite the abundance of hospitals and medical facilities, some individuals still live in remote areas without access to superior healthcare. Individuals who care about their health may benefit from this systems. If the government legitimately confirms the system's effectiveness, implementing such a system might not be illegal. Health care systems are under stress as a result of a combination of chronic illness, increased patient demand, and resource limitations. In addition, as the use of digital health devices increases, more data is being gathered across all healthcare settings, which has had some impact on legal difficulties. In the proposed system, data would be fully utilized, enabling healthcare professionals to concentrate on the underlying causes of illness and monitor the effectiveness of medications and preventative measures. The legitimate organization will confirm all information required for the system's growth, in accordance with all legal standards. The several legislative frameworks governing AI and data in healthcare provide guidance on how to interpret existing law, describe novel methods to satisfy the requirements, and foster innovation and competition (frontiers, 2022).

The suggested strategy addressed governance and patient empowerment issues as well as issues of trust and comprehension. People may have doubts about such a system because it is difficult to believe that a machine can diagnose diseases by simply providing symptoms. Most users are unfamiliar with the system and have limited confidence in its capability to predict illnesses. Concerns about the effectiveness of the proposed system should be added. Offerings include pertinent training and convincing descriptions of the benefits and insights of the proposed system (Deloitte, 2020).

Similar ethical issues impact the healthcare industry as a whole. In actuality, the problem is problematic in standard medical care. According to the principle of autonomy, clinicians would abide by a patient's request to stop therapy in order to maintain their quality of life, for instance. However, when patients are unconscious or incapable of making the best choices, this presents a challenging ethical dilemma. As a result, this system will deal with any ethical problems. The system will be able to anticipate diseases based on symptoms and be created with all legal, social, and ethical considerations. 2019 (Victor Chang).

Professionally, this system could be beneficial like honey to a bee. The people who are far from medical facilities might save some time and get a medical consultant from the medical experts. There are a lot of rural areas where people are unaware of medical services; for those individuals, medical professional skills may provide and, using this platform, educate them on how to prevent diseases. Medical professionals with competence can examine a variety of patient records and get an overview of the diseases that most commonly infect people.

2.4. Approaches Taken

2.4.1. Explanation of the proposed solution for solving the problem

Data gathering, algorithm selection, model construction, and developing model for prediction is essential methods for this project. In this project, SVM machine learning method and the random forest machine learning algorithm is implemented to create the model. Each and every step of the development process is accomplished using the DSDM methodology. A Gantt chart is made to schedule the tasks so that they can be completed on time. Below is an overview of some of the most important approaches that should be taken to accomplish this project:

2.4.2. Data collection and pre-processing.

Well, data for this project is collected from the kaggle.com. The dataset includes about 134 different kinds of symptoms (itching, skin rash, nodal skin eruptions, stomach pain, vomiting,

anxiety etc.). The dataset consists of 4921 rows and 18 features. The dataset contain the target variable (which is a diseases) like fungal infection, allergy, GERD, Chronic Cholestasis, AIDS, Diabetes etc.

2.4.3. Selecting algorithms.

In this step, appropriate algorithm is selected according to the reliability, dimension and features of the dataset. Algorithms are used for recognizing patterns and building the model. In this project SVM and random forest algorithm is selected.

2.4.4. Building model and prediction

In this step, the model is built by feeding the dataset according to the nature of the algorithm which is used for prediction. The model is iteratively built in this project till the satisfactory accuracy is obtained.

2.5. Algorithms

2.5.1. K – Nearest Neighbors Algorithm

The k- nearest neighbors is a supervised learning classifier and it is a non-parametric algorithm. It classifies or predict the elements by grouping of its individual data points (IBM, 2021). K-Nearest neighbors' algorithms are used in basically search, regression and classification problems by assuming that the similar points can be found near one another in a close proximity. The KNN algorithms works by depending upon the similarity, distance, proximity and closeness.

For prediction problems, the average value of K nearest neighbors plays a significant role to make a prediction depending upon the continuous value about the classification where classification is used as a discrete values as an output. K-NN algorithms stores all the available data and classify new data point by keeping the new data point into the category that is most similar to the available categories (java T point , 2021). The Euclidean distance between the new data point and the available data/ categories data point is defined. Since it is an instance-based/ memory based learning method, it focused on storing all its training data. As a result, it is able to stores all training data during the operation of computation when prediction is made. The main concept of K-NN is to label class by identifying the nearest neighbours. When nearest neighbour is determine, it will makes easy to determine which data points are closest to a given input data.

As mention in the above, the algorithm works on the base of distances between the input data point and the available data point. There are different types of distance measuring methods that can be used in this algorithms, some of them are:-

- Euclidean distance

  This method is commonly used for determining the distance between the input data and the available data which has a limited to real-valued vectors.

  Equation:-

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (Y_i - Xi)^2}$$

- Manhattan distance

  This method is a distance metric which can be used to measure absolute distance between the input data points and the available data points.

  Equation:

$$Manhattan\ Distance = (\sum_{i=1}^{m} |X_i - Y_i|)$$

Finally, after deep research and analysis, I found that K-NN algorithms become significantly slower as the volume of data increases which makes impractical choice in environments where predictions need to be made rapidly. Moreover, it is difficult to find the precise parameter value of K where the diseases prediction can be done accurately. The dataset collected for the health diseases prediction is noisy as well as it have outliers where multiple measured needs to be theorized for the accurate creation of the classification/ clusters. But the K-NN algorithms only takes the nearest neighbor for classification which is not enough to produce high accuracy while performing prediction (Shahadat Uddin, 2022).

Hence, The K-NN algorithms is not used in this project.

2.5.2. Decision Trees

Decision Trees (DTs) algorithm is a non – parametric supervised learning method which can be used for classification and regression. It determines the set of decision from the data features to create a model for predicting targeted variable which means it create a training model to predict the respective target variable by learning sorts of decision from training data. (scit-learn , 2021). It is called decision trees as it uses a tree-like model of decisions allowing to visualize as well as to explicit the represented data with corresponding to decision making.

Decision trees is based on the sum of product (SOP) representation. Either it act as disjunction or conjunction. Every attributes similar from the root node to the leaf node with a same class is known as conjunction whereas attributes from the root nodes ending to different leaf nodes of that class is called disjunction. As it uses hierarchical tree-like structure as a constant approximation, it has a root nodes, decision nodes, leaf nodes, sub-tree and pruning.

- Root Node

  This represents the entire attributes and elements which is distributed into different homogeneous sets or into two or more sub-nodes.
- Decision nodes

  This represent the distribution of attributes to their similar from sub-nodes to further sub-nodes.
- Leaf Node

  This node is also called as terminal node as it doesn't split into other sub-nodes.
- Sub-tree

  This is the node which is a subsection of the sub-node or entire branches.
- Pruning

  This is an opposite process of splitting where sub-nodes of a decision node is removed to stop overfitting.

  (Chauhan, 2022)

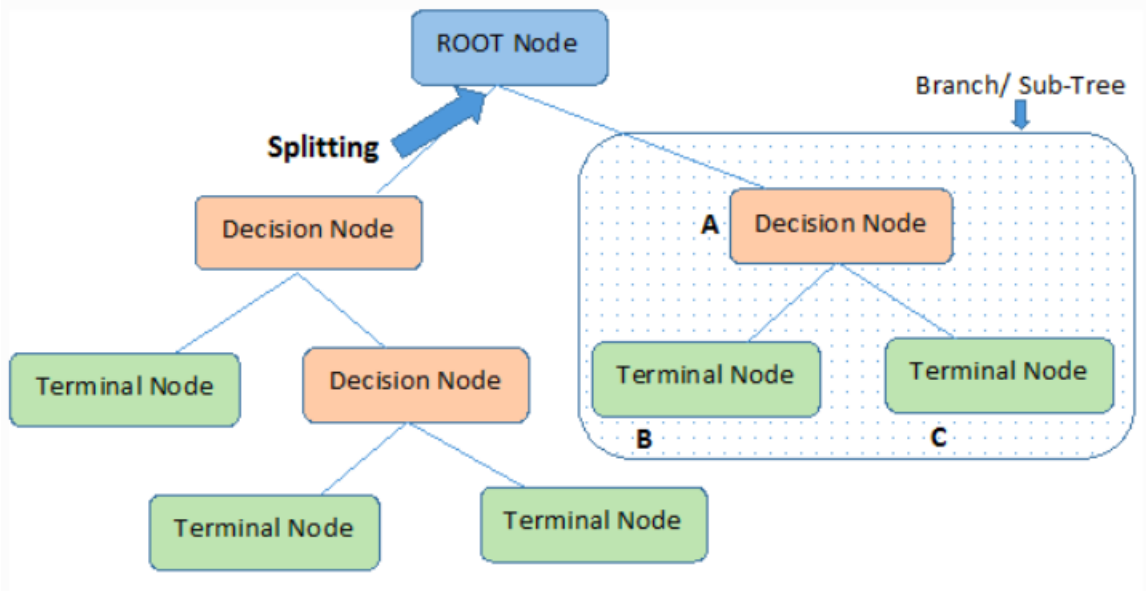

Figure 1 Architecture of decision tree

In this method the process of prediction of a class label start from the root node. The value of the root

attribute is compared with the internal node/record attribute. Decision trees classify the input variable between the terminal nodes by sorting them down from the root node. The operation in the decision trees are recursive in nature and is repeated for every terminal nodes / subtree rooted at the new nodes. Each node of the decision tree is able to perform as a test case for every attributes, and recursively each edge respectively corresponds to the test case. The tree accuracy depends on how decision is made for strategic split. Multiple algorithms may be used for making a decision whether to split a node into two or more sub-nodes. The sub-nodes increases the homogeneity with respect to the targeted variable.

2.5.3. Naive Bayes classifier

Naïve Bayes is a supervised and probabilistic machine learning algorithm that is based on the Bayes theorem. This algorithm can only be used for classification problems (Gandhi, 2018). Naive Bayes can be used for both small-size samples as well as large-size samples. It segregates the independence of attributes by assuming that the particular features in a class are unrelated to the other features. Bayes' theorem:- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

Where variable A and B states the event, P (A) is the probability of event A, P (B) is the probability of event B and P (A|B) is a posteriori probability of A and B. There are different types of Naïve Bayes classifiers, some of them are Gaussian Naïve Bayes, Multinomial Naïve Bayes, and Complement Naïve Bayes, Bernoulli Naïve Bayes, Categorical Naïve Bayes and Out-of-core naïve Bayes model fitting.

Besides this algorithm being simple, efficient, and easy to build, this algorithm is not used in this project (health prediction system) because it assumes all features are independent which is not accurate as most of the features of the sample depend on each other practically. Since the output made by this algorithm is on the probability basic it may not be accurate. Moreover, in the health diseases dataset, complex attribute dependencies can be present where this algorithm cannot give accurate output (M. A. Jabbar, 2016).

2.6. Selected Algorithm

2.6.1. Support Vector Machine (SVM)

SVM is a supervised machine learning algorithm which is a discriminative classifier with a hyperplane. It can be used for regression, classification and for outlier detection which mapped the data points to the different category and those category are separated by hyperplane making a wider gap as much as possible. SVM represent the different classes of data in a hyperplane in an N-

dimensional space which depends on the number of features. The hyperplane is the finest line that is used for classifying the data points into the different categories. The hyperplane operates in an iterative manners in a SVM for minimizing the error. The hyperplane line of SVM classifies the data point into a two different classes where on one side, it represent the data points as a category while on the other side of hyperplane line, the newly input data point is classifies into a different category (McGregor, 2020). SVM can be used for all types of dataset like separable, inseparable, linear planes, and non-linear planes.

The main objective of SVM is to segregate the given data points into a best possible way. The nearest point between the given data points and the category which is obtained after the classification is known as the margin. The efficiency of the SVM algorithm depends on the maximum possible margin between the two classes/ support vector that can satisfy the hyperplane. The hyperplane select the maximum possible margin that can satisfy the hyperplane so that it can classify the given data point in a best possible way. Finest hyperplane for linear and separable variable is selected from the nearest data points where maximum classification has been occur. But for non-linear and non-linear planes, the SVM uses the kernel that can transfer the low dimensional input into a higher-dimensional space which makes data easier to classify into the different category points.

Similarly, SVM consists of SVM Kernels which helps for classifying inseparable data points into a separable data points for accurate segregation by adding a multiple dimensions layer. SVM has different kinds of Kernel, They are linear kernel, polynomial kernel, radial basis function kernel and Hyperbolic Tangent Kernel.

- Linear Kernel

  Linear kernel classify the non-linear dimensional input into a higher dimensional data point for classification.

  Equation:- $k(x, y) = x + c$

  The linear kernel is responsible for transferring the lower dimensional data point to the higher dimensional data point relatively associated with short training duration.
- Polynomial Kernel

  This kernel is responsible for normalizing training data sets. This kernel works well when the sample of data points is fewer and is used for determining curved or nonlinear input space.

  Equation:- $(x,Y) = (ax^T Y + c)$

- Radial basis function

  This function of the kernel can deals with the both huge and small dataset. This kernel is commonly used and has high performance then other three kernel function. This kernel can mapped the data points in infinite dimension

  Equation:-

  (l, m) = exp (11- $ml^2/2\sigma^2$)

- Hyperbolic Tangent Kernel

  This kernel works when the features space has low dimension. This kernel can works on both small as well as large sample.

  Equation:- $(x,\Upsilon) = \mathrm{h}(ax^T \Upsilon + c)$

(L. Mohan, 2020)

Reason behind selecting SVM

SVM technique provide a high performance rate for solving classification problems like diseases prediction using common variable and is a discriminative power for classification in the cases where the dataset size is small but the number of features is larger (High dimension space). SVM has the same method like other algorithms where the steps like data pre-processing, measurement, variable selection, modelling building, performance evaluation, cross –validation test etc. In model building for diseases prediction, SVM mapped the training set of vector-label pairs to a different category for classification and for the training set of vector pairs which are not linear, SVM uses kernel function by transforming the non-linear vector pairs to a higher dimension for segregation using a satisfactory hyperplane for a linear data points (Youn-jung Son, 2010).

2.6.2. Random forest

Random forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions. It consists of a large number of individual decision trees that operate as segregation elements for individual models. Multiple associated decision trees produce the class prediction and the maximum number of similar class predictions is taken as the model for prediction, similar majority output produced by the individual decision trees is considered as the final predicted value (Yiu, 2019). As it has number of tree, it generate the precision output as well as it take priority over the limitation of the decision tree by minimizing the overfitting the dataset.

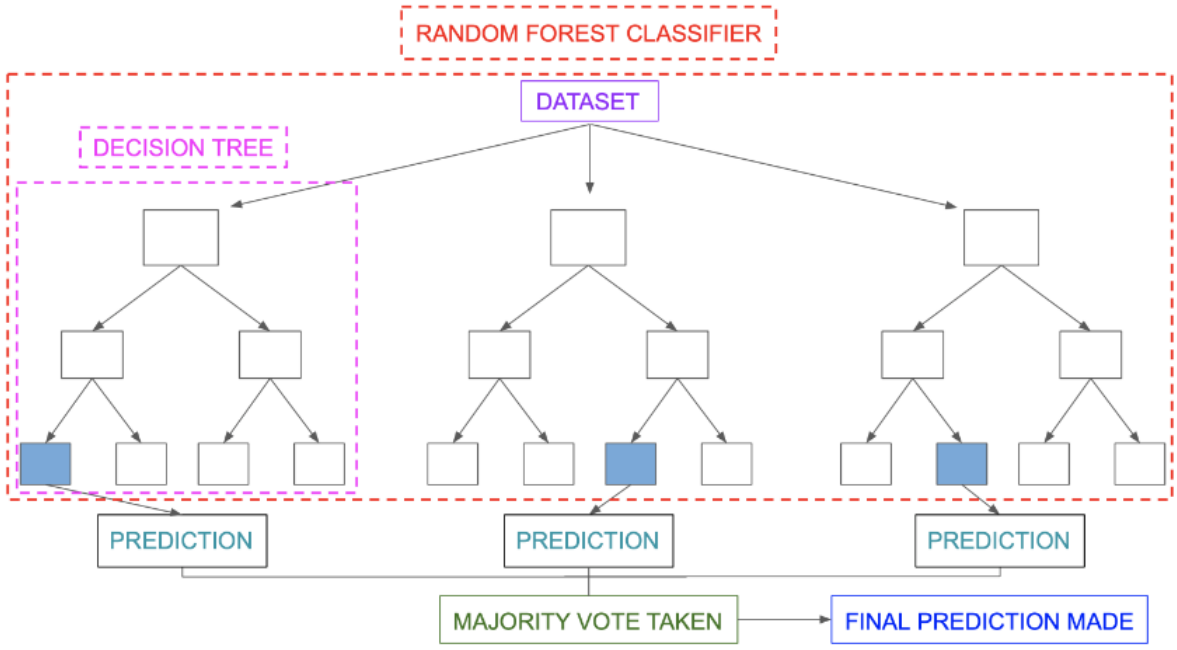

Figure 2 Architecture of Random forest

As shown in the above figure, random forest generate a multiple decision tree and considered the majority selection of prediction as the final result. Firstly, in the random forest, three main hyper-parameters (node size, number of trees, number of features sampled) are set and the dataset is transferred to each individual decision tree for training (IBM, 2020). Hyper-parameter enables random forest either to increase the predictive power of the model or to make the model performance faster (Donges, 2021). During the splitting of nodes, a random subset of the features is taken into consideration which contains multiple features as well as dimensions. As these algorithms are based on the decision tree, it contains decision nodes, leaf nodes, and root nodes similar to the decision tree. Each node has its specific features.

- Root node

  In this node, the dataset is transferred for training to each individual decision tree.
- Decision node

  In this node, each individual decision tree generates respective output.
- Leaf node

  This node is the final output produced by the decision tree on the basis of a majority similar prediction.

Moreover, the number of decision trees required in the random forest depends upon the size of the dataset.

Reason behind selecting this algorithm

From the above research, it is known that the random forest algorithm is a robust, reliable, and efficient algorithm for classification problems like “disease prediction”. It consists of a default hyper-parameter that gives an accurate prediction. It can handle higher dimensional data as well as reduce the risk of overfitting. The dataset of diseases may consist larger in size as well as there might be a number of features, dependencies, and different dimensional to which this algorithm can give the best result.

CHAPTER 3 – DEVELOPMENT

3.1. Considered Software development Methodologies

Software development methodologies are the order of steps in the software development process that determine which steps should proceed after which process is in sequential order. These methodologies are also known as software development life cycles. This step is very important and must be done successfully in order to finish any project. It evaluates all of the potential factors that contribute to the success or failure of a project. There are many different methodologies for developing software, but in order to complete a project within the allotted time frame, the appropriate algorithm needs to be chosen. This choice needs to be made in accordance with the size and complexity of the project. Selecting appropriate methodologies will help to minimize costs, and resources and increase efficiency.

Prototype methodology, Rapid application development methodology, and Rational Unified Process Methodology are some of the various kinds of methodologies that have been taken into consideration for this project and DSDM methodology.

3.1.1. Prototype Methodology

Building, testing, and refining the prototype of a system are the steps involved in the prototype methodology. These steps are repeated until an acceptable prototype is created. This methodology functions most effectively in contexts, where the requirements of the project are unknown. This methodology begins with the gathering of requirements using a variety of sources, such as interviews, surveys, and other similar methods. After the requirements have been gathered, a streamlined design of the system is developed. Following the completion of the design analysis, the prototype is then put through its paces. In a similar vein, the process of re-designing, re-building the prototype, re-testing, and reworking the system is iterated between the developer and the client

until the system is built to the satisfaction of the client. This process continues until the system is built to the client's specifications (Martin, 2022).

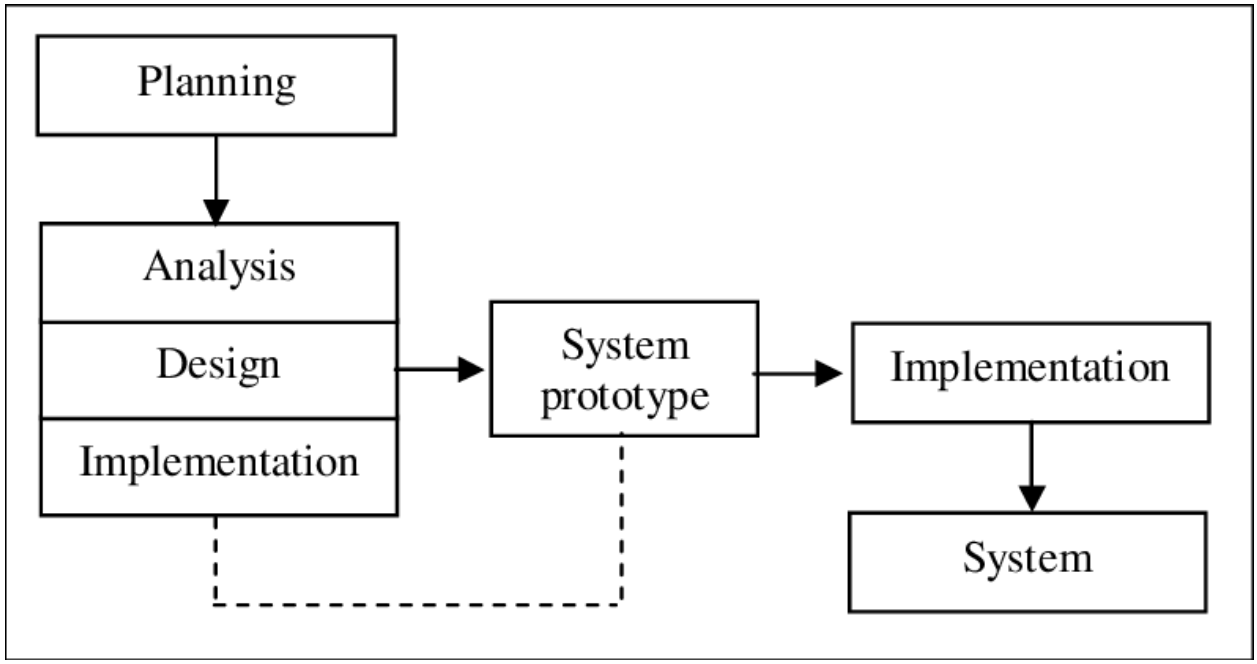

Figure 3 Prototype methodology

The main reason behind the consideration of this methodology:

- The customer is given significant input into the design process. As a consequence of this, there is a good possibility that the system will be developed in accordance with the anticipations of the customer.
- There will be a reduced likelihood of the software being rejected.
- Because there is less need for specialized training, the end users can become very familiar with the system before it is fully developed.
- The developer will be able to achieve better software development solutions with the assistance of immediate user feedback.

Limitations of the Prototyping methodology are:

- It is difficult for software developers to incorporate all of the changes that are requested by customers when prototyping because the process is slow and time-consuming.
- Because the needs of the customers are constantly evolving, it is likely that a significant financial investment will be necessary to re-design and reconstruct the system.
- There is a possibility that customers will not be willing to take part in the iteration cycle for a period of time that is significantly longer.

3.1.2. Rapid Application development

RAD (Rapid Application Development) entails building the system with minimal planning in favor of rapid prototyping to make it easier to incorporate the changes within the development process. To ensure timely delivery, integrated functional models are developed as a prototype into the final product in parallel. Business modelling is the first step in this framework. In this phase, an in-depth analysis of the company's operations is conducted to uncover crucial data (tutorialspoint , 2022). Rapid Application Development Methodology includes the following stages: The phases of Rapid Application Development Methodology are:

- Data Modelling.
- Process Modelling.
- Application Modelling.
- Application Generation.
- Testing.
- Turnover.

The main reason behind the consideration of this methodology:

- It is reliable and changeable as well as useful in lowering project risk and cost.
- The system can be built quickly with fewer resources.
- Reduced reliance on hand-written code attributable to the widespread use of code generators and code reuse.

Limitation of this methodology:

- This model isn't appropriate when there are high technical risks as well as it can’t be used for smaller projects.
- Following this methodology for developing a system requires highly skilled developers.
- Time-boxing, in which features are delayed to later versions to complete a release in a short period, resulted in fewer features being included.

3.1.3. Rational Unified Process Methodology

As an agile software development methodology, RUP (Rational Unified Process Methodology) completes projects in four phases: modeling, analysis and design, implementation, testing, and application. This system's process is iterative because its fundamental steps are repeated repeatedly

throughout the undertaking. Up until the system satisfies the client's requirements and objectives, various project components can be changed, and the fundamental process can be repeated (toolshero, 2022). The phase of the Rational Unified Process Methodology are:

- Business Modeling
- Requirement analysis and design
- Implementation
- Test
- Deployment

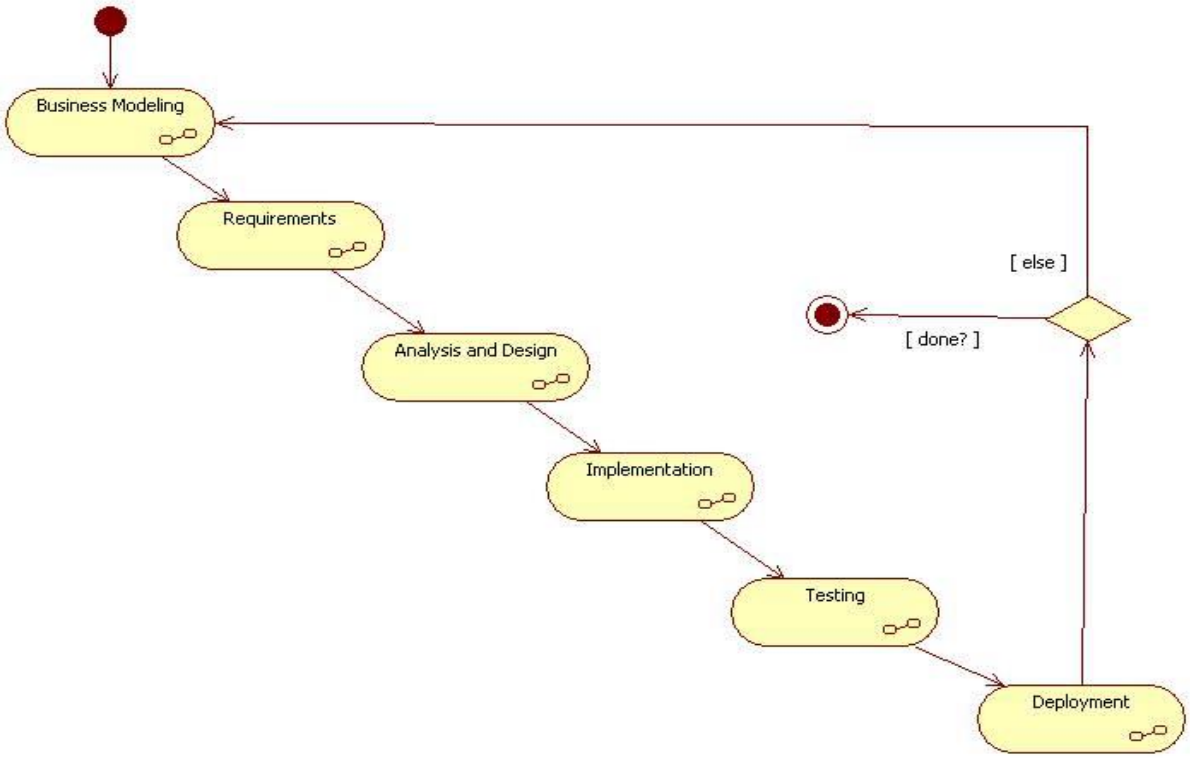

Figure 4 Rational unified process

The primary reasons for taking into account this methodology are as follows:

- Since it recycles its components, it reduces the amount of time needed for the development process.
- Because it uses an iterative process, this methodology is superior for producing high-quality software within a predetermined amount of time.
- This method includes the creation of comprehensive documentation.

The Rational Unified Process Methodology has a few drawbacks, the most significant of which are as follows:

- The software development process is overly complicated and disorganized.
- For this methodology to be used, software developers need to be highly skilled in their respective fields.
- It has a fair amount of overhead and isn't quite as flexible and adaptive as agile, which is a significant drawback.

3.1.4. DSDM

The Dynamic Systems Development Method, also known as DSDM, is an approach that is iterative and incremental, and it places an emphasis on the active participation of the users. It places an emphasis on working together and coordinating efforts between the users and the developer. The users' expectations regarding all facets of the system are taken into account during the development of the DSDM prototype of the system (Boog, 2022). The delivery of the system in a relatively short amount of time is the primary focus of this methodology. The Distributed Systems Development Method (DSDM) places an emphasis on delivering the product on a regular basis during each iteration by utilizing a variety of tools, techniques, and practices that are designed to support the overall DSDM process. These are:

- Time-boxing.
- MoSCow prioritization.
- Facilitated workshops.
- Iterative development.
- Modelling and prototyping techniques.

Time-boxing, Pre-project Phase, Feasibility Study, Business Study, Functional Model Iteration, Design and Build the Iteration, and Implementation are the phases involved in DSDM. Each stage was essential to completing the project within the allotted time (Gupta, 2022).

Time-boxing:

For obvious reasons, we need to finish the project quickly, so the time boxing method is utilized. One or more deliverables are established as goals to be completed within a specified time frame (the "time box") in DSDM. A time box is a period of time, typically 3–4 weeks, during which a specific task must be completed in order to enable the iterative development of a product.

Pre-project Phase:

Before beginning the project in earnest, there is an interim known as the pre-project phase. During this stage, the decisions regarding the ideas and the overall plan for the beginning of the project are made. The plan is developed in preparation for the completion of the project, and it takes into account all of the requirements and resources.

Feasibility Study:

Research is carried out at this stage, as it is essential to the progress of the project. In this stage, we set out to develop a comprehensive plan for assessing the needs, challenges, and potential options. Market, legal, social, and ethical implications are initially examined to determine the system's viability.

Business Study:

In this phase, the business aspect of the project is research studied and identified.

Functional Model Iteration:

The development of a prototype system and a working model are both parts of this phase of the process. At the outset, remarkable features and functionality that are capable of catering to the requirements of the customer are analysed and taken into consideration. The most important key functionality that is required for the system has been identified, and development on it has begun on a priority basis.

Design and build the Iteration:

During this phase, the prototype is developed and then shown off to the client in order to make sure that the final product lives up to the client's anticipations in terms of its features. Iterative progress has been made with the designs and development features until they meet the requirements specified by the client.

Implementation:

All of the functionality is set up together at this point because it is the final phase. Each component of the newly developed system is set up through its paces with a variety of testing procedures to guarantee that it functions appropriately. The users are given instructions on how to operate the

systems properly. Following the completion of the user training, the system is rolled out to the consumer base. It is necessary to perform reviewing activities from the user in order to compare the requirements and the accuracy of the project.

3.2. Selected Methodology

3.2.1. DSDM

Different methodologies are mentioned in the above section with their advantage and disadvantage. DSDM methodology with a time-boxing technique is selected as the project contains machine learning which needs to accomplish within a given time frame. Its stages contain the strategic alignment with the complication of the project effectively which is required for completing the project. The phases involved in DSDM span the entire lifecycle of a project which provides the full roadmap for delivering on time. Throughout these phases, it allows a project to go through a controlled start to a point where the understanding of the project is good enough to start building the solution iteratively and incrementally. The work performed to finalize this project on the basis of the DSDM stage is discussed below:

3.2.1.1. Stage

1. Feasibility study

Several systems that are quite similar to one another are analyzed, and an in-depth study is carried out in order to determine whether or not the system meets the aim of the project. There are a variety of publications and research papers that make up the system's development process, and these are the ones that are studied. Within the realm of medicine, a survey is carried out to learn more about the challenges faced and potential answers. Examining a variety of journals, books, websites, and conducting interviews are some of the methods that are used to carry out in-depth research on the subject of "health diseases prediction," with the goal of identifying the issues and developing potential solutions.

The dataset including information about disease symptoms used in the experiment was obtained from the website kaggle.com. It is possible to make predictions using a variety of methods, including decision trees, naive Bayes, random forests, K-NN neighbors, and support vector machines, all of which are the subject of ongoing research. In particular, a Random forest and a support vector machine have been chosen for use in this research. An estimation of the costs is carried out. In addition, research is conducted on the necessary resources as well as the technical

aspects, such as the programming language, the framework, etc. In addition, the technical abilities necessary for the successful completion of this project are recognized and acquired.

2. Business Study

During this stage, research of the commercial implications of "Diseases prediction" is carried out. In particular, during this phase, an investigation into how this system might benefit or be stimulated from a commercial standpoint in the coming days is carried out. As a result, the survey is carried out prior to the creation of the system during this phase. In addition, the comprehension of the effectiveness of the system in relation to the rating and review of a comparable system is included in this step. In the first step of the development process, an examination of the influence that the system will have in the future is carried out. The scope of the project as well as any possible risks that may emerge from the developed system are examined, and potential remedies are studied and taken into consideration.

3. Functional Model Iteration

Patients, doctors, medical organizations, and medical researchers are all potential end users of this project, a prototype of the system is developed in order to confirm that it can meet the requirements of these end users. In order to meet the precise requirements of the user and to create the prototype system for disease prediction based on symptoms, certain elements of the system were counted. Following receipt of the feedback from the end-users, it has been determined that there is a requirement for an effective project that provides a more accurate prediction as well as the requirement for new features.

In a similar manner, the functional model is constructed and tested through its paces multiple times. The remarks made by the end-users are taken into account, through which users' expectations can be mapped with the aim of the project by the process of iteratively building and testing the functional model. During this phase, the development tasks such as identifying required dependencies, loading a dataset, and partitioning the dataset for training are carried out.

4. Design and build iteration

In this phase, the function of the system is design and developed based on the priority. The wireframe for the system is designed and the based on the wireframe user interface for the system

is develop using the bootstrap and HTML. This is the phase where the essential features like login features, registration features, displaying features etc. are developed. The backend development is done in the visual studio code using the python programing language. Moreover, the model is developed in this phase which predict the diseases.

5. Implementation

The system developed is completed and the detailed documentation about the project and development process is documented clearly. The users are trained in how to use the system.

3.3. System development

The development is performed in both Jupyter notebook and VisualStudio code. VisualStudio code is used for developing some web features as well to integrate the system with the web using the Django framework. However, the code that is developed in the jupyter notebook is used for exploring dataset, documentation purpose, visualizing dataset and checking the accuracy of the model. The development process and the library used in this project are mention below with screenshot.

3.4.1. Development in Jupyter Notebook

The SVM and the random forest algorithm are used for developing model for prediction in this project. Before implementing the algorithms, there are important steps that needs to be performed for building accurate model such as to check the dimensional space, cross-validation, co-relation, dependencies etc. These steps are difficult to perform and test in visual studio code because visual studio code can’t provide efficient graph and output as jupyter notebook, so, the jupyter notebook is used.

3.4.1.1. Importing dependencies

The required dependencies are identified and imported them into the jupyter for developing the system.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from sklearn.model_selection import train_test_split
        from sklearn.svm import SVC
        from sklearn.metrics import f1_score, accuracy_score, confusion_matrix
        from sklearn.ensemble import RandomForestClassifier
        %matplotlib inline
        from matplotlib.pylab import rcParams
        rcParams['figure.figsize'] = 15, 6
```

Figure 5 libraries

S.N	Modules	Reason for Importing
1	Pandas	As an array-based data storage structure, Pandas is a Python library which read as well processed dataset.
2.	Numpy	Numpy is used for performing mathematical functions. A significant number of mathematical, algebraic, and transformation functions can be found inside this collection. As a consequence of this, it is designed to carry out a variety of mathematical operations on arrays.
3.	Matplotlib	Since it is a plotting package for Python, it may be used for creating graphical representations of numerical data.
4.	Sklearn	It's a free and open-source machine learning package that works for both supervised and unsupervised training. Model fitting, data pre-processing, model selection, model evaluation, etc. are just a few

		of the many tasks that it can handle (scikit-learn, 2022).
6.	sklearn.svm	Sklearn.svm is used for training dataset building model for prediction.
7.	sklearn.metrics	Sklearn.metrics is used in this project to implements several loss, score, add utility functions to measure classification performance.
8.	Sklearn.ensemble	In order to increase generalizability/robustness compared to a single estimator, this project uses Sklearn.ensemble to integrate the prediction of numerous base estimators built with a specific learning algorithm.

Table 2 library description

3.4.1.2. Loading dataset.

After importing the dependencies, the dataset is loaded in the program using the pandas which are in the CSV format. The dataset is loaded and store in the variable called dataset using the pd.read_CSV format.

```
In [5]: df = pd.read_csv('diseases_dataset.csv')
        df.head()
```

Out[5]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	scurring	skin_peeli
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	

5 rows × 134 columns

Figure 6 importing dataset

3.4.1.3. Pre-processing dataset

This is the important steps where data cleaning, data integration. Data selection and data transformation process are performed. In this step any missing value and irrelevant value are eliminated and the dataset is reshaped into the new data frame, the empty columns are filled with the value 0.

```
: df.isna().sum()
  df.isnull().sum()
  cols = df.columns
  data = df[cols].values.flatten()
  s = pd.Series(data)
  s = s.str.strip()
  s = s.values.reshape(df.shape)
  df = pd.DataFrame(s, columns=df.columns)
  df = df.fillna(0)
  df.describe
```

```
: <bound method NDFrame.describe of       itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  \
  0           0          0                     0                    0
  1           0          0                     0                    0
  2           0          0                     0                    0
  3           0          0                     0                    0
  4           0          0                     0                    0
  ...       ...        ...                   ...                  ...
  4915        0          0                     0                    0
  4916        0          0                     0                    0
  4917        0          0                     0                    0
  4918        0          0                     0                    0
  4919        0          0                     0                    0
```

Figure 7 checking null value, reshaping data into a new data frame

The test is performed to check the if the dataset is balance or not, below screenshot and graph shows that all the data of the dataset are balance as they are of have equal number attributes.

```
In [20]: disease_counts = df["prognosis"].value_counts()
         temp_df = pd.DataFrame({
                     "Disease": disease_counts.index,
                     "Counts": disease_counts.values
                 })
         disease_counts
```

```
Out[20]: Fungal infection                              120
         Hepatitis C                                   120
         Hepatitis E                                   120
         Alcoholic hepatitis                           120
         Tuberculosis                                  120
         Common Cold                                   120
         Pneumonia                                     120
         Dimorphic hemmorhoids(piles)                  120
         Heart attack                                  120
         Varicose veins                                120
         Hypothyroidism                                120
         Hyperthyroidism                               120
         Hypoglycemia                                  120
         Osteoarthristis                               120
         Arthritis                                     120
         (vertigo) Paroymsal  Positional Vertigo       120
         Acne                                          120
         Urinary tract infection                       120
         Psoriasis                                     120
         Hepatitis D                                   120
         Hepatitis B                                   120
         Allergy                                       120
         hepatitis A                                   120
         GERD                                          120
         Chronic cholestasis                           120
         Drug Reaction                                 120
         Peptic ulcer diseae                           120
         AIDS                                          120
```

Figure 8 Checking whether the dataset are balance or not

```
plt.figure(figsize = (18,8))
sns.barplot(x = "Disease", y = "Counts", data = temp_df)
plt.xticks(rotation=90)
plt.show()
```

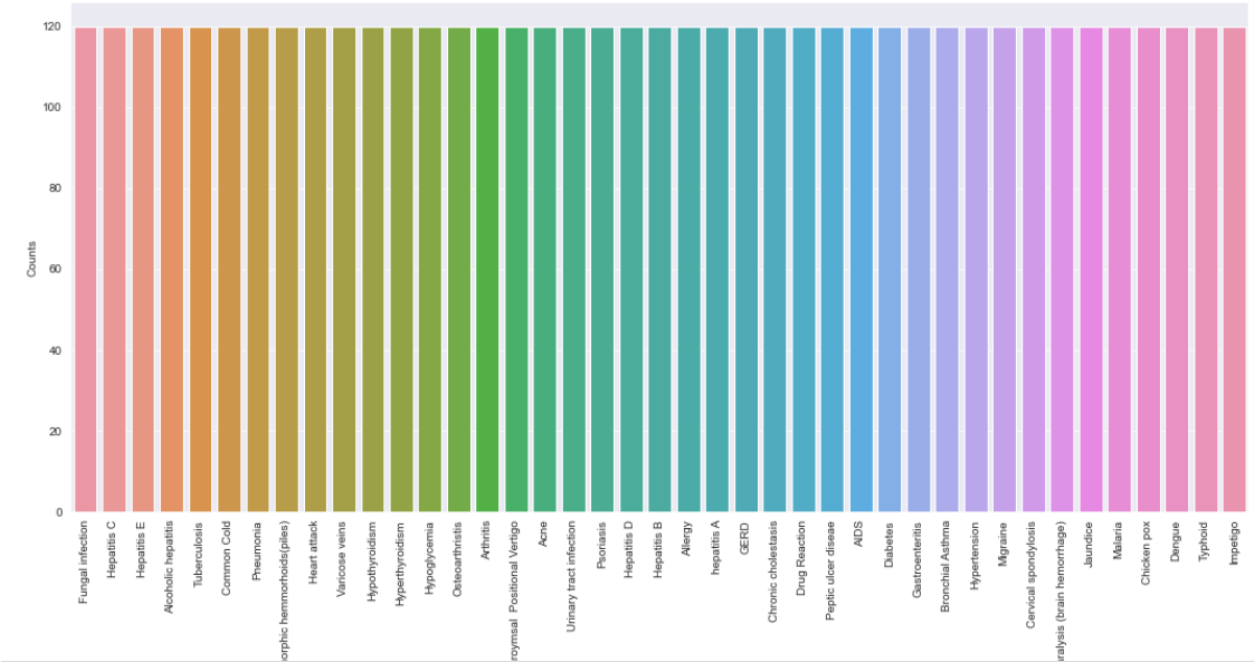

Figure 9 Graph shows all the variable in the dataset are in a balance form.

The target variable is encoded/transform into the numerical value so that it can be easy to train dataset for preparing the model.

```
encoder = LabelEncoder()
df["prognosis"] = encoder.fit_transform(df["prognosis"])
X = df.iloc[:,:-1]
y = df.iloc[:, -1]
X
```

...	blackheads	scurring	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	blister	red_sore_around_nose	yellow_crust_ooze	prognosis
...	0	0	0	0	0	0	0	0	0	15
...	0	0	0	0	0	0	0	0	0	15
...	0	0	0	0	0	0	0	0	0	15
...	0	0	0	0	0	0	0	0	0	15
...	0	0	0	0	0	0	0	0	0	15
...	...	...	...	...	...	...	...	...	...	...
...	0	0	0	0	0	0	0	0	0	0
...	0	0	0	0	0	0	0	0	0	2
...	0	0	0	0	0	0	0	0	0	38
...	0	0	0	0	0	0	0	0	0	35
...	0	0	0	0	0	0	0	0	0	27

Figure 10 encoded target value into numerical

3.4.1.4. Splitting dataset

The dataset is splitted for training. 80% data of dataset is splitted for training purpose whereas remaining 20% of dataset is left for testing.

```
X_train, X_test, y_train, y_test = train_test_split(
           X,y,train_size= 0.80, shuffle= True, random_state=24)
```

Figure 11 Splitting dataset

3.4.1.5. Building model and performing prediction

In this steps, the model is built. The dataset is trained using the SVM and random forest. Similarly, train dataset is fitted in a both model which can be used for prediction. Here, we can see model is built and using the model the prediction is performed.

```
final_svm_model = SVC()
final_rf_model = RandomForestClassifier()
final_svm_model.fit(x_train, y_train)
final_rf_model.fit(x_train, y_train)

svm_preds = final_svm_model.predict(x_test)
rf_preds = final_rf_model.predict(x_test)

print(svm_preds)
print(rf_preds)
```

```
(4182, 17) (738, 17) (4182,) (738,)
['Pneumonia' 'Impetigo' 'Urinary tract infection' 'Fungal infection'
 'GERD' 'Peptic ulcer diseae' 'Gastroenteritis' 'AIDS' 'Common Cold'
 'Dengue' 'GERD' 'Diabetes' 'Gastroenteritis'
 '(vertigo) Paroymsal  Positional Vertigo' 'Urinary tract infection'
 'Bronchial Asthma' 'Paralysis (brain hemorrhage)' 'Migraine'
 'Chronic cholestasis' 'Hepatitis D' 'Dimorphic hemmorhoids(piles)' 'Acne'
 'Common Cold' 'Cervical spondylosis' 'Hypertension' 'Dengue'
 'Fungal infection' 'AIDS' 'Paralysis (brain hemorrhage)'
 '(vertigo) Paroymsal  Positional Vertigo' 'Chicken pox'
```

Figure 12 Building model

3.4.1.6. Checking Accuracy

After building the model, the accuracy is calculated of individual model as well of combined model. Accuracy determines how much predicted value is near to its true value or recognized standard. The accuracy is calculated using the confusion matrix.

1. SVM model accuracy

Figure shows the heatmap and accuracy obtained by the SVM model. The accuracy obtained by the SVM model is 93.49 and the F1- score is 93.21.

```
conf_mat = confusion_matrix(y_test, svm_preds)
df_cm = pd.DataFrame(conf_mat, index=df['Disease'].unique(), columns=df['Disease'].unique())
print('F1-score% =', f1_score(y_test, svm_preds, average='macro')*100, '|', 'Accuracy% obtained by SVM =', accuracy_score(y_test
sns.heatmap(df_cm)
```

```
F1-score% = 93.21202458356615 | Accuracy% obtained by SVM = 93.4959349593496

<AxesSubplot:>
```

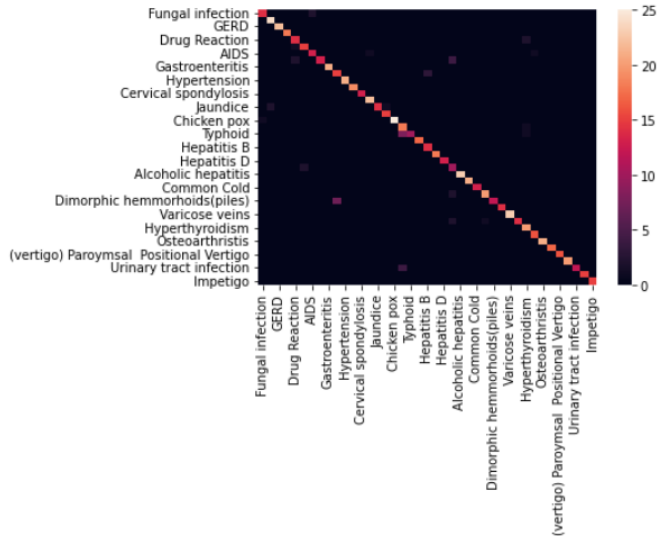

Figure 13 SVM accuracy

2. Random forest model accuracy

Figure shows the heatmap and accuracy obtained by the random forest. The accuracy obtained by the random forest model is 99.45 and the F1-score % is 93.2120

```
In [25]: conf_mat = confusion_matrix(y_test, rf_preds)
         df_cm = pd.DataFrame(conf_mat, index=df['Disease'].unique(), columns=df['Disease'].unique())
         print('F1-score% =', f1_score(y_test, svm_preds, average='macro')*100, '|', 'Accuracy% =', accuracy_score(y_test, rf_preds)*100)
         sns.heatmap(df_cm)

         F1-score% = 93.21202458356615 | Accuracy% = 99.45799457994579

Out[25]: <AxesSubplot:>
```

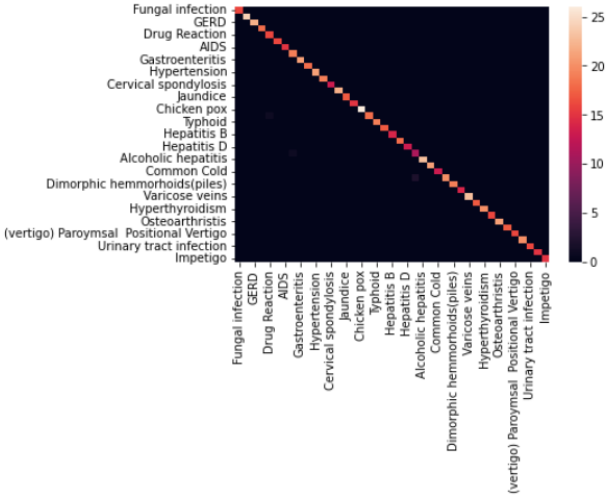

Figure 14 heatmap of random forest model

3. Combined model accuracy (SVM + random forest)

Figure shows the heatmap and accuracy obtained by the combined model. The accuracy obtained by the combined model is 95.39.

```
final_preds = [mode([i,j])[0][0] for i,j
                    in zip(svm_preds, rf_preds)]

print(f"Accuracy on Test dataset by the combined model\
    : {accuracy_score(y_test, final_preds)*100}")
```

```
Accuracy on Test dataset by the combined model    : 95.39295392953929
```

Figure 15 Accuracy obtained by the combined model

```
sns.heatmap(cf_matrix, annot = True)
plt.title("Confusion Matrix for Combined Model on Test Dataset")
plt.show()
```

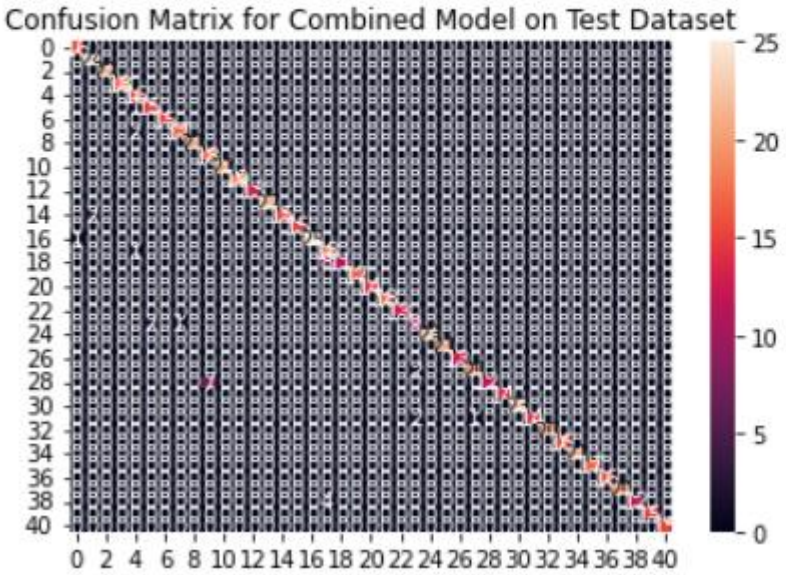

Figure 16 Heat map produce by the combined model

3.4.5. Performance evaluation of the model

Below table shows clearly that individually random forest performed better than SVM. However when the two algorithm is combined the model performance is appreciable.

Algorithm	Dataset	Accuracy
SVM	Training size = 80% Testing size = 20%	93.49%
Random forest	Training size = 80% Testing size = 20%	99.45%
Combined model	Training size = 80% Testing size = 20%	95.39%

Table 3 Performance evaluation of the model

CHAPTER 4 – TESTING AND ANALYSIS

4.1. Test Plan

The most important aspect of the project is testing if the system's built-in functionality functions or not. Unit testing and system testing, the two testing techniques described below, are carried out here.

4.1.1. Unit Testing, Test plan

In the software development process known as unit testing, the smallest testable unit of code, or unit, is examined for correct operation on its own (Tech Target, 2021) . A test environment is used to run the system code functions, and the result is compared to what is anticipated. Below is a description of the test case that was run in this project as part of the unit testing.

S.N	Test Case	Outcome
1	UI execution Test	Successful
2	Loading dataset Test	Successful
3	balancing test	Successful
4	Splitting the dataset and training the model	Successful
5	Combining models test	successful

Table 4 Unit testing, Test plan

Test Type	Unit testing
Objective	To determining the correct functioning of the script responsible for lunching the UI
Performed action	The program is executed from the visual studio using command python manage.py runserver
Expected outcome	The web browser should display the UI
Obtained outcome	The web browser is displayed with the user interface
Test Result	Successful

Table 5 UI execution test

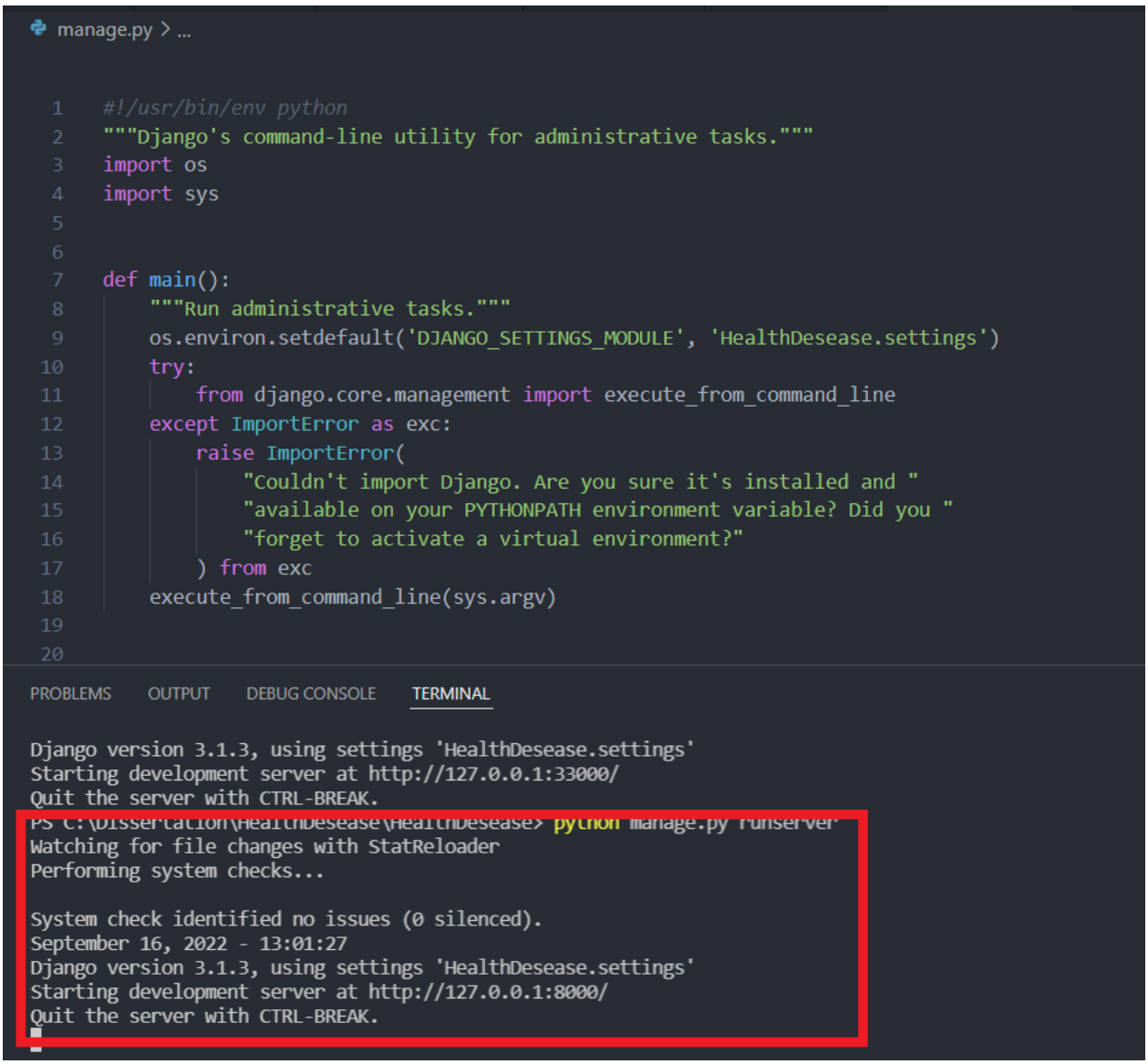

Figure 17 UI execute successfully

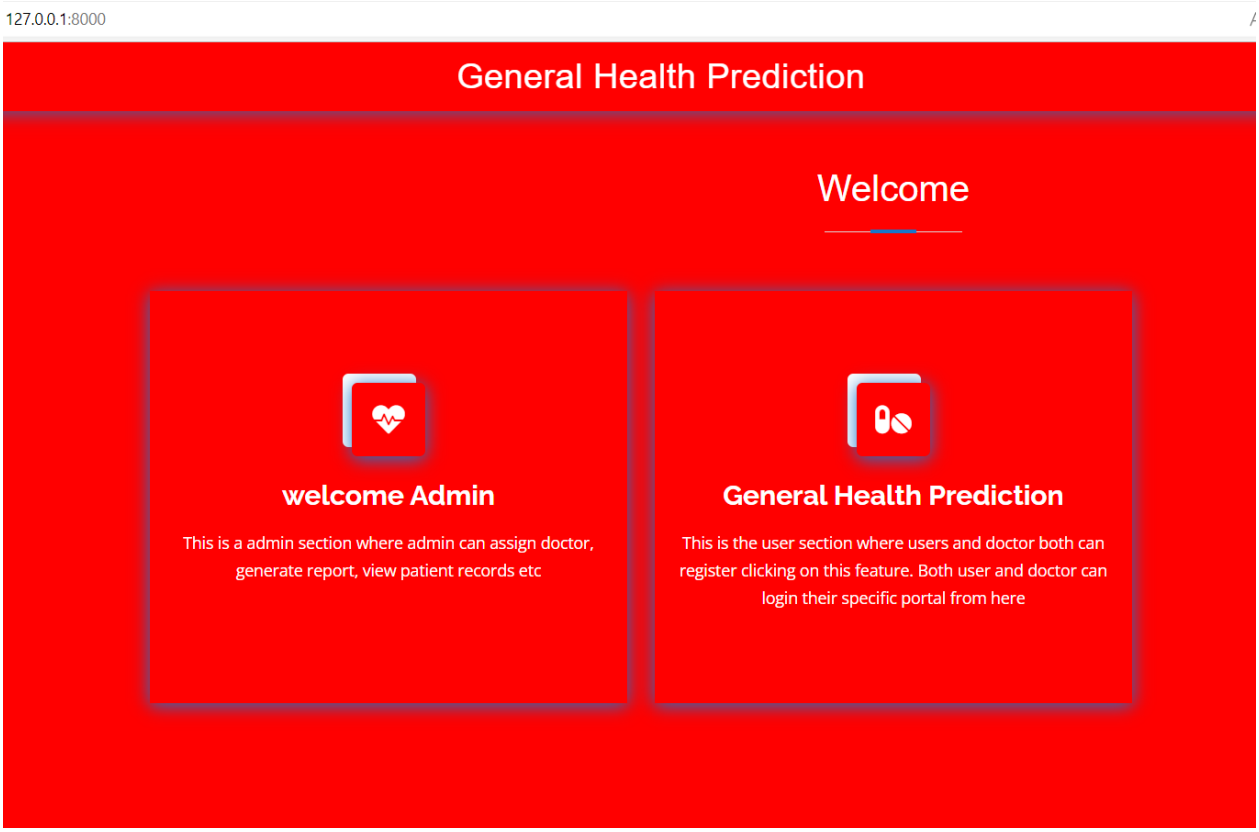

Figure 18 User interface of the system

1. Loading dataset test

Test Type	Unit testing
Objective	To determining the correct functioning of the method used for loading dataset.
Performed action	The health diseases dataset is loaded
Expected outcome	The dataset should be loaded into the program
Obtained outcome	The dataset is loaded into the program successfully.
Test Result	Successful

Table 6 loading dataset test

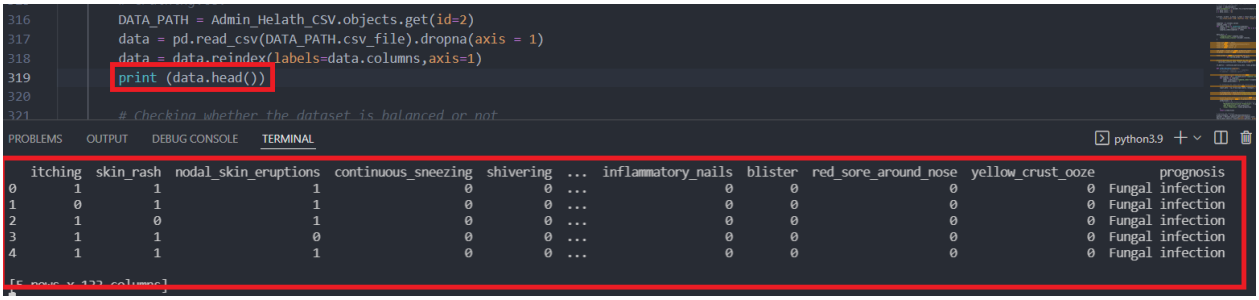

Figure 19 Loading dataset

2. Balancing dataset test

Test Type	Unit testing
Objective	To determining whether all the features in the dataset are balance or not.
Performed action	The index of the variable of the dataset is count.
Expected outcome	The program should display all the variables of the dataset with the balance value in a terminal.
Obtained outcome	All the variables are displayed with the balance value in a terminal after the program is executed.
Test Result	Successful

Table 7 balancing dataset test

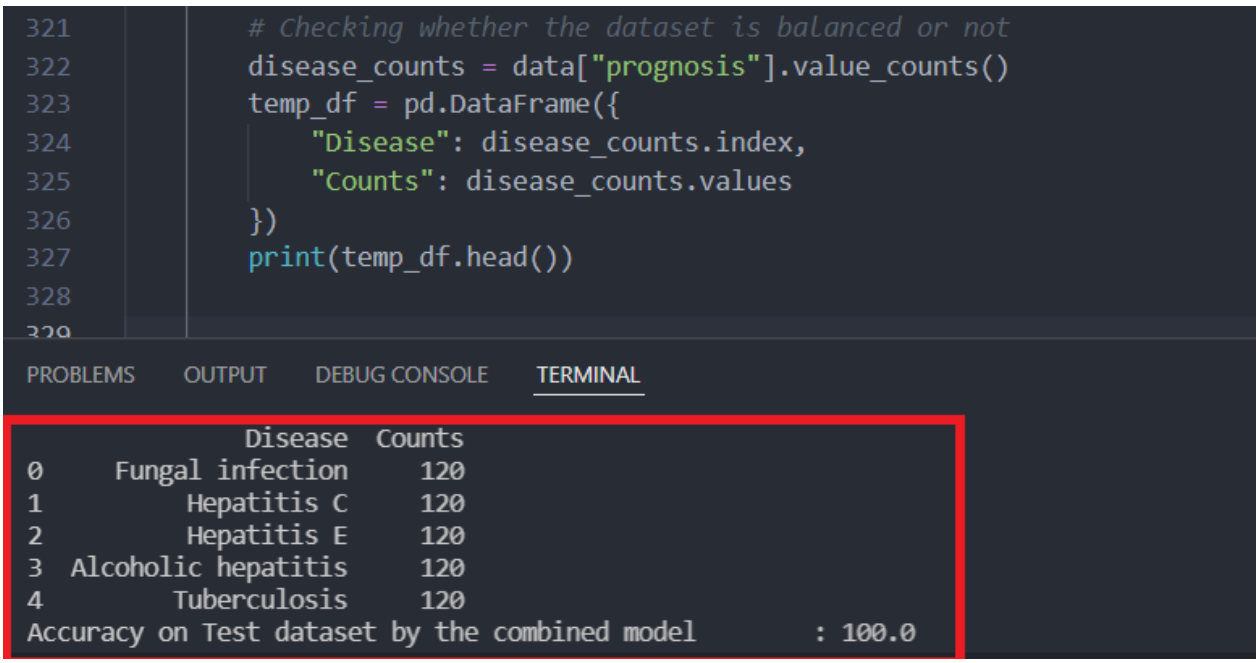

Figure 20 checking if the dataset is balanced or not

3. Splitting the data and training the model

Test Type	Unit testing
Objective	To determining whether the data is split and the model is build or not.
Performed action	The program is executed
Expected outcome	The program should be run without any error and it should display the variable that are trained using the model.
Obtained outcome	The program is executed successfully without error and the trained variable should be displayed in the terminal
Test Result	Successful

Table 8 splitting the data and training the model - unit test

```
337         #spliting model
338         X_train, X_test, y_train, y_test = train_test_split(
339             X,y,train_size= 0.80, shuffle= True, random_state=24)
340         #training model
341         final_svm_model = SVC()
342         final_rf_model = RandomForestClassifier()
343         final_svm_model.fit(X, y)
344         final_rf_model.fit(X, y)
345         # building model
346         svm_preds = final_svm_model.predict(X_test)
347         rf_preds = final_rf_model.predict(X_test)
348         print(svm_preds)
349         print (rf_preds)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[ 7 26 11 21 40 12 14 30  0 15 17 12 20 28  7 18  2 35 28  8  0 18  8  9
  9  8 10 38 13 17 15 34 36 23 15 38  7 38  8 23  6 10 33 29 11  6 24 33
  1 29  7  5 31 26 23 26  0 18 14 28 12 22  0  6  5 23 20 26 18 37  5 14
  2 23 32  2 15 32 37  0  1  4 32 38  6  1 25 38 30 19 32 14 11 39  7 15
 40 19 13 31 19  0 11 15 27  6 18 39  7 27  6 21 35 38  6 22 11 40 19 10
 12 26 10 26 34  6 35 20  8 14 17 39  6 10 11 37 30 12  8  2  5  5 14  2
 13  9 30  1 30 24 36 25 37 34 13 39 11 13  4  9  3 29 35  9  7 36  6  4
 36 15 30 13 31  1  9 10 17 32 16 38 32 20  6 28 19  1 14 35 35 12  1 13
  4 16 19 38 31 25 16 25 31  4 21 16 31 21 24 33 35 23  9 40 11 36 10 23
  7 26 27  7  0 13 26 39 12 27 16 12 36  6 17 30  7  6 14 15  6 15 16 20
 34  9 18 12 13 31 29 39 17 29 20 32  0 23 12 38 26 12  4  5 15 14 21 19
```

Figure 21 Train variable displayed in a array -unit testing

4. Combining model test

Test Type	Unit testing
Objective	To determining whether the two model SVM and random forest is combined together or not.
Performed action	The program is executed
Expected outcome	The program should be run without any error and both model should be combined into a single variable.
Obtained outcome	The program is executed successfully without error and the model must be combined into a single variable.
Test Result	Successful

Table 9 Combined model test

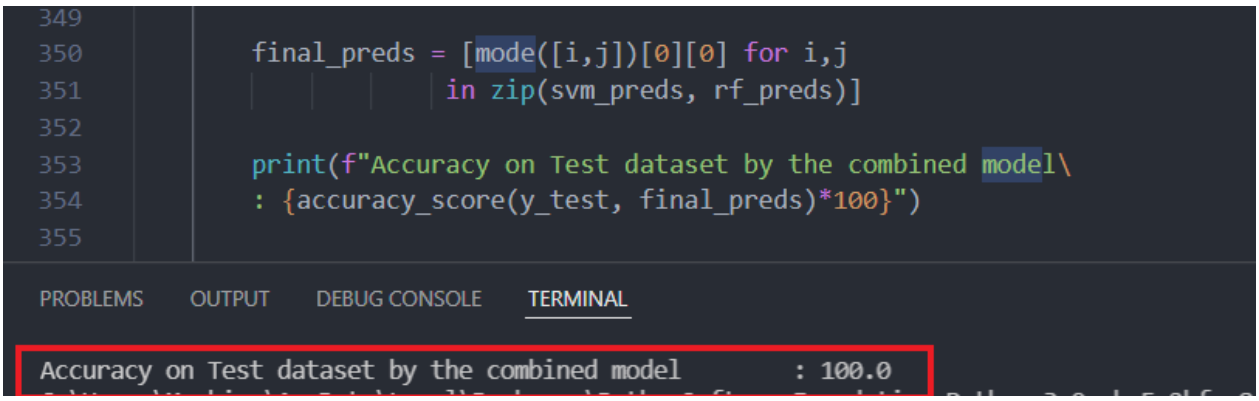

Figure 22 Combined model test displaying successfully

4.1.2. System Testing, Test Plan

System testing refers to evaluating the system as a whole, including integrated functions or components, to ensure the system performs as intended (softwaretesting , 2022). The system's individual parts have all been tested and integrated.

Below is a description of the test case's specific details as it relates to system testing.

S. N	Test Case	Outcome
1	Browser compatible Test	Successful
2	User registration and login test	Successful
3.	Doctor registration and login test	Successful
4.	Admin login test	Successful
5.	Updating profile test	Successful
6.	Delete functionality test	Successful
7.	Admin assigning doctor role test	Successful
8.	Performing prediction test	Successful
9.	Logout test	Successful

Table 10 System Testing, test plan

1. Browser compatible test

Test Type	System testing

Objective	To determine the browser compatible
Performed action	The project is executed in a different browser
Expected outcome	The project should be executed in a different browser without any errors.
Obtained outcome	The project is executed in a different browser without any error
Test result	Successful.

Table 11 Browser compatible test

Executing in Microsoft edge browser.

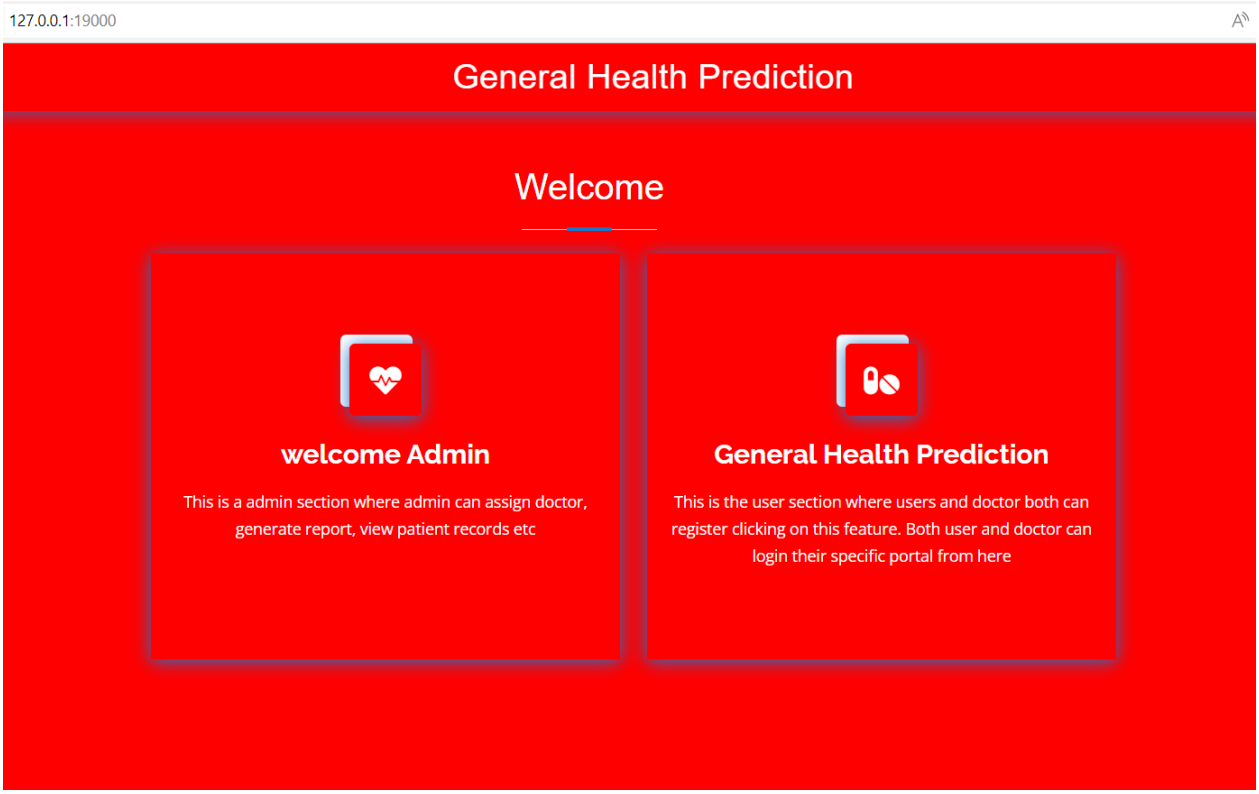

Figure 23 successfully executed in Microsoft edge

Executing in google chromo browser.

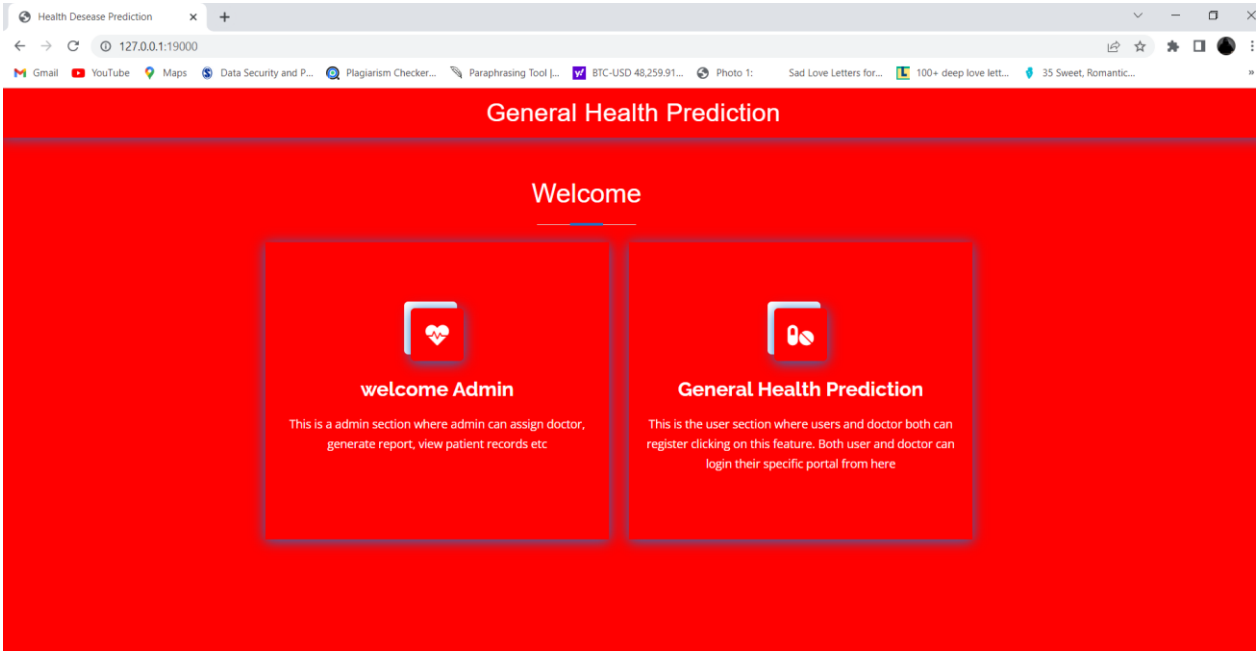

Figure 24 successfully executed chromo browser

2. User registration and logging test

Test Type	System testing
Objective	To determine the successful registration and logging of user
Performed action	Registration form is filled with required details and the button registration is pressed as well login with registered username and password.
Expected outcome	Users should be registered in the system successfully and the should be able to login.
Obtained outcome	User is successfully registered and login system.
Test result	Successful

Table 12User registration test

After providing the information as per the registration form and after clicking the registered button, user should be registered in the system.

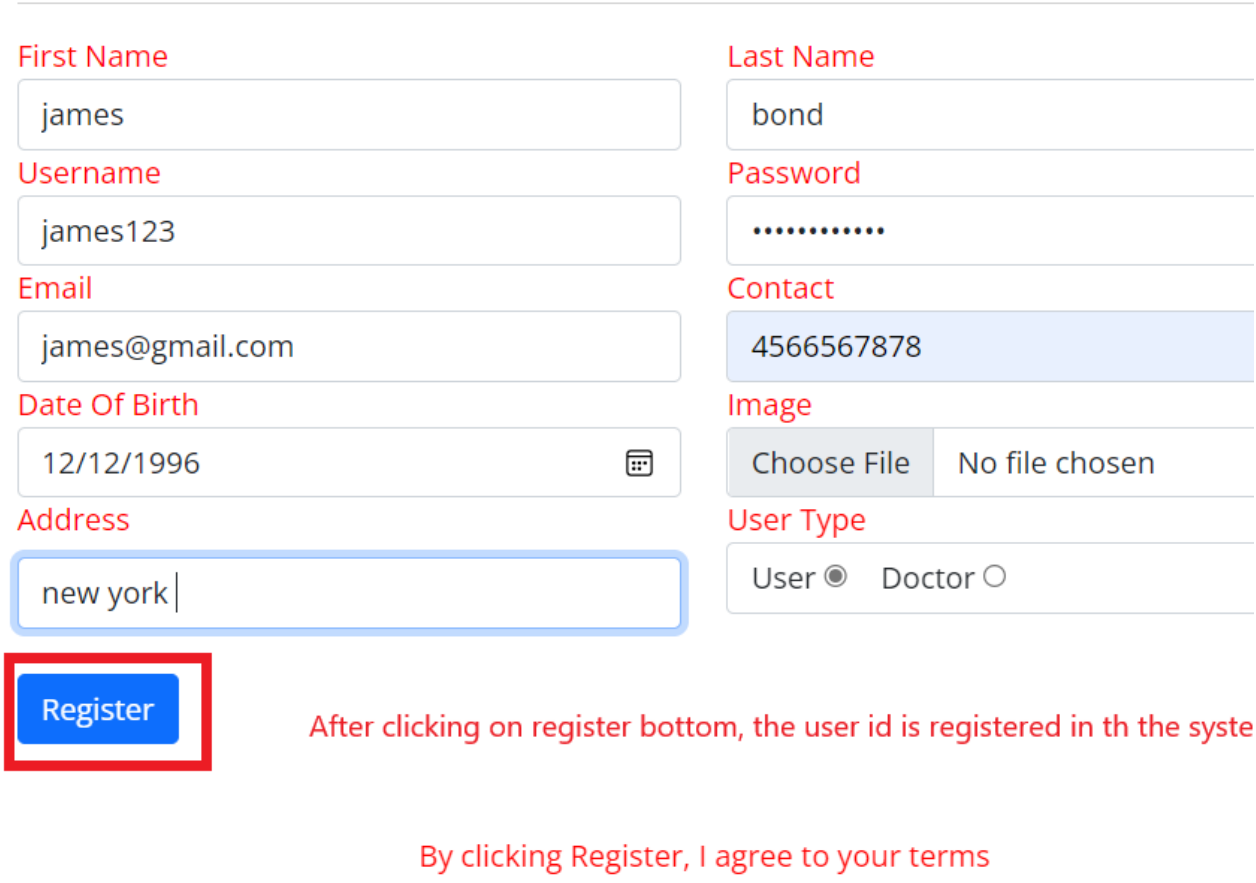

Figure 25 registration form

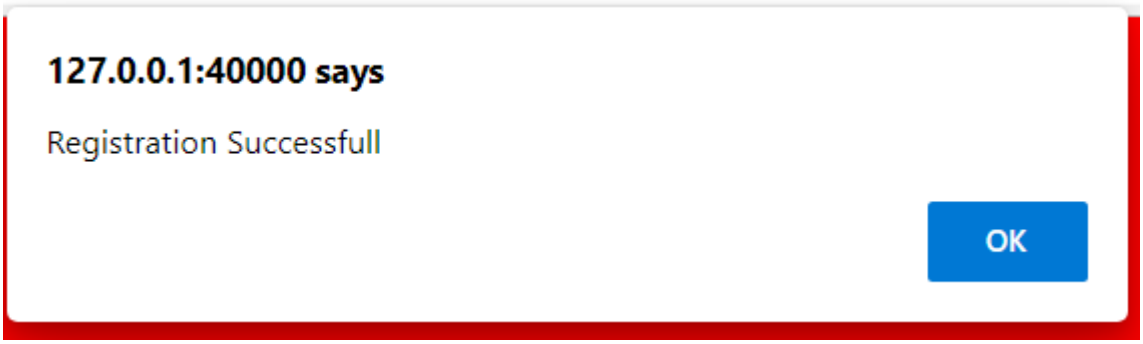

Figure 26 registered successfully

After registration, users should be able to login in the system.

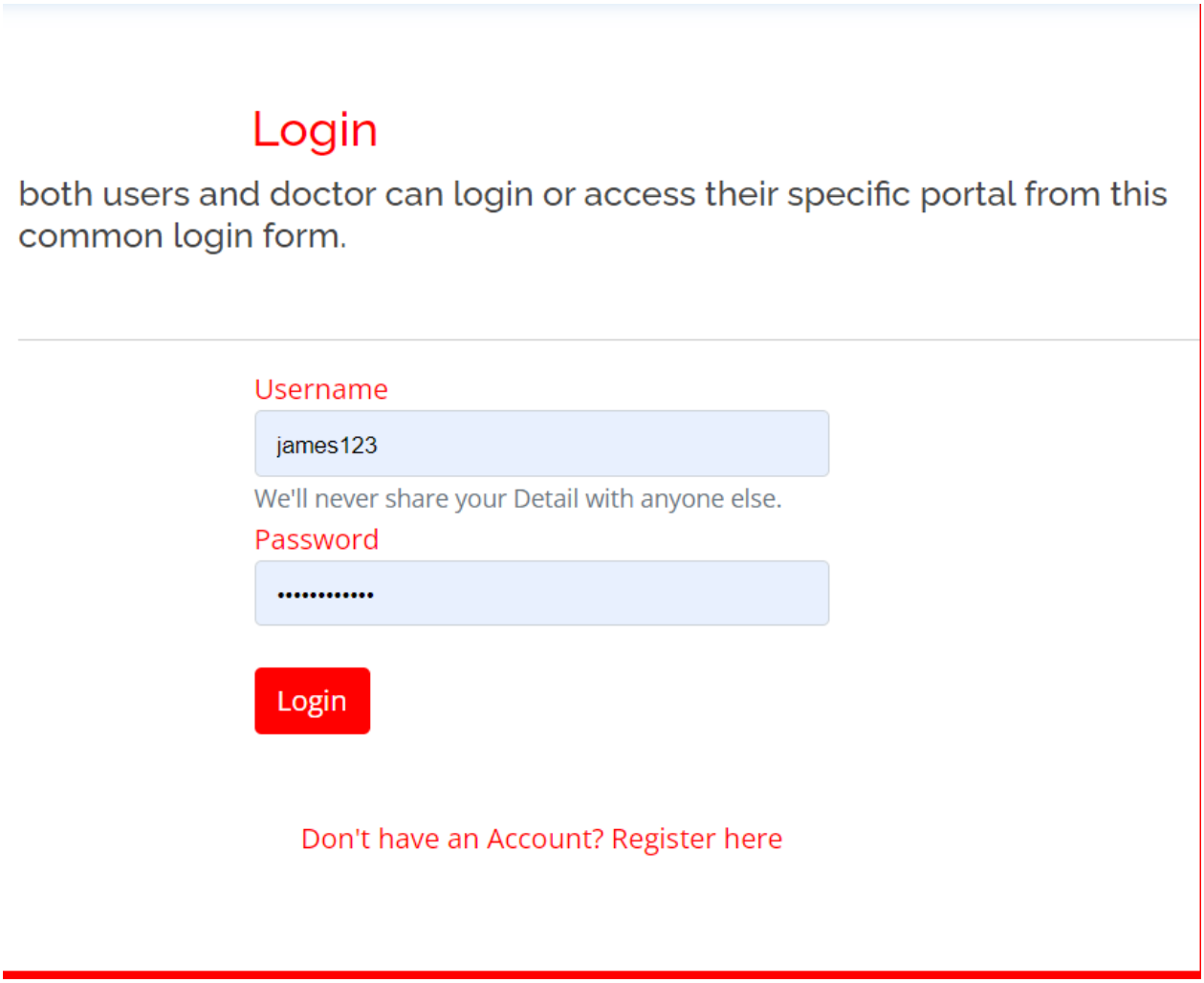

Figure 27 User's login form test

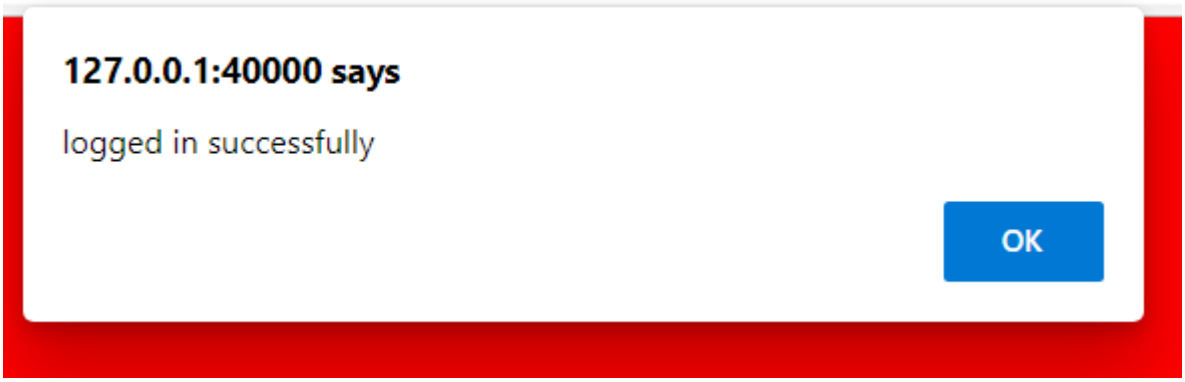

Figure 28 Login successfully test

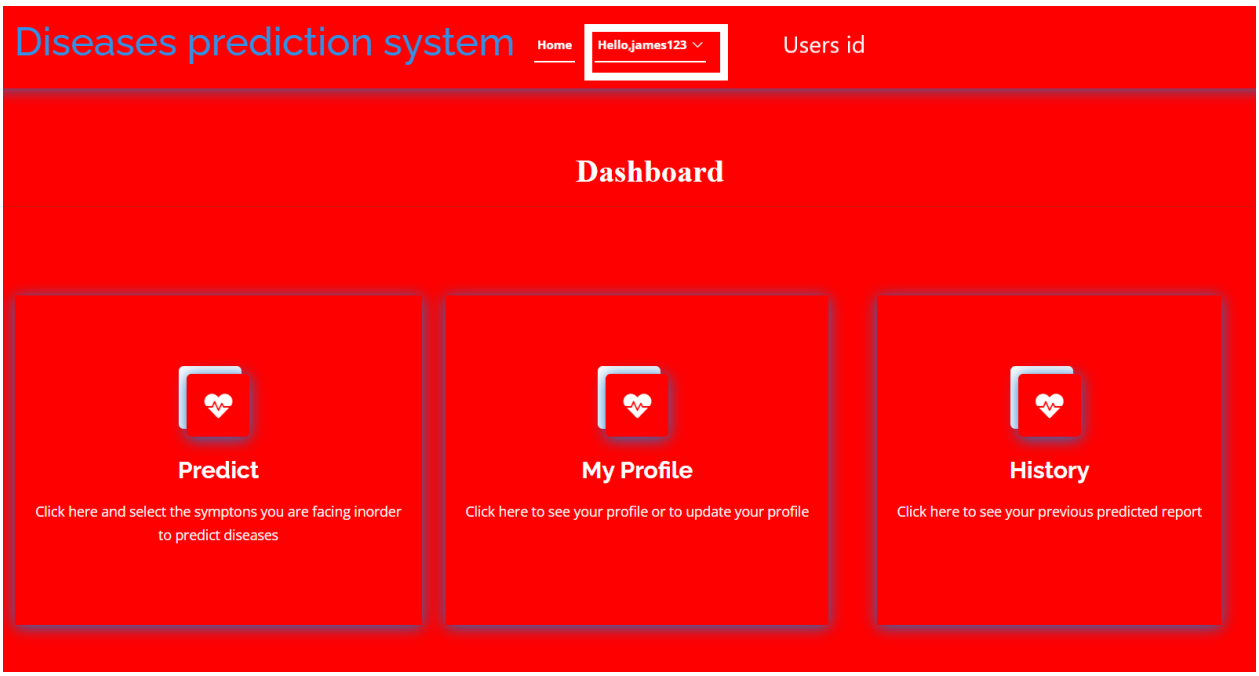

Figure 29 User's portal after registration

3. Doctor registration and login test

Test Type	System testing
Objective	To determine the successful registration and logging after registration of doctor.
Performed action	Registration form is filled with required details and the button registration is pressed as well login with registered username and password.
Expected outcome	Doctors should be able to registered and login the system by registered username and password
Obtained outcome	Doctor is registered and login the system.
Test result	Successful.

Table 13 Doctor registration and login test

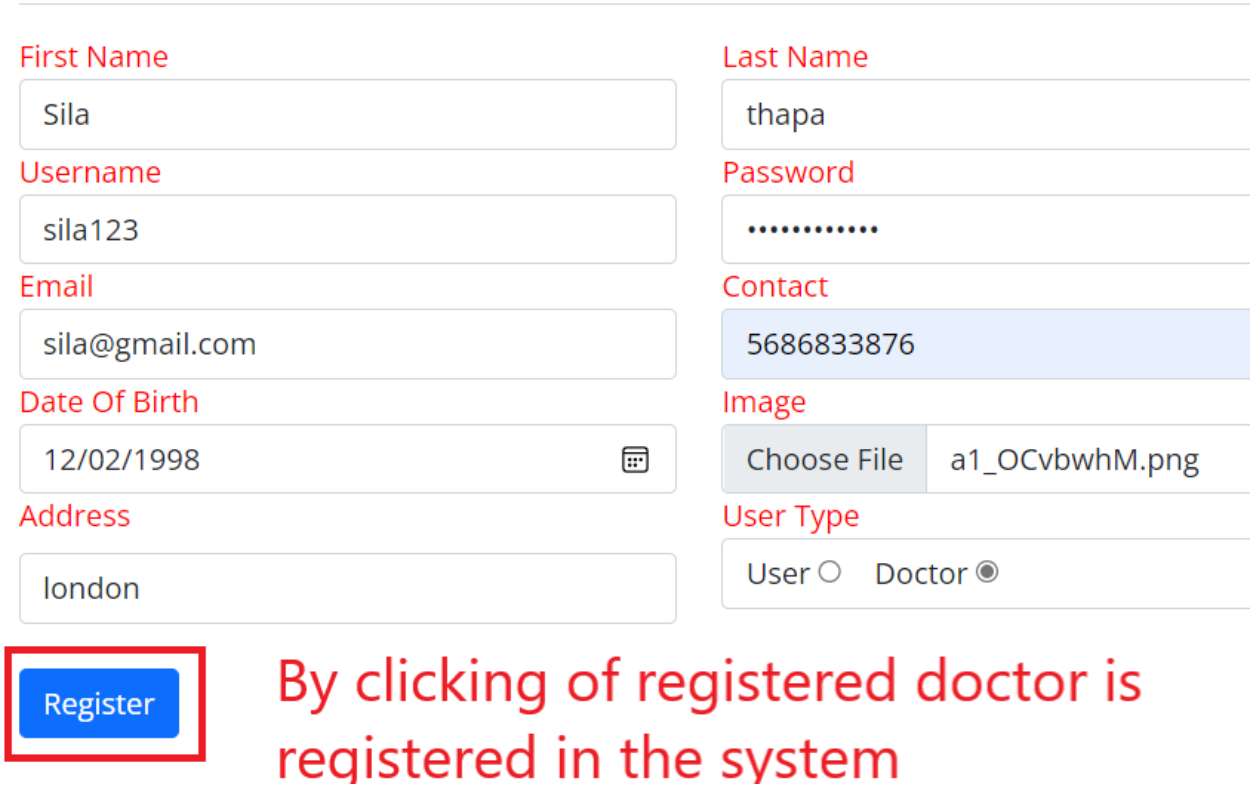

Figure 30 doctor registration form test

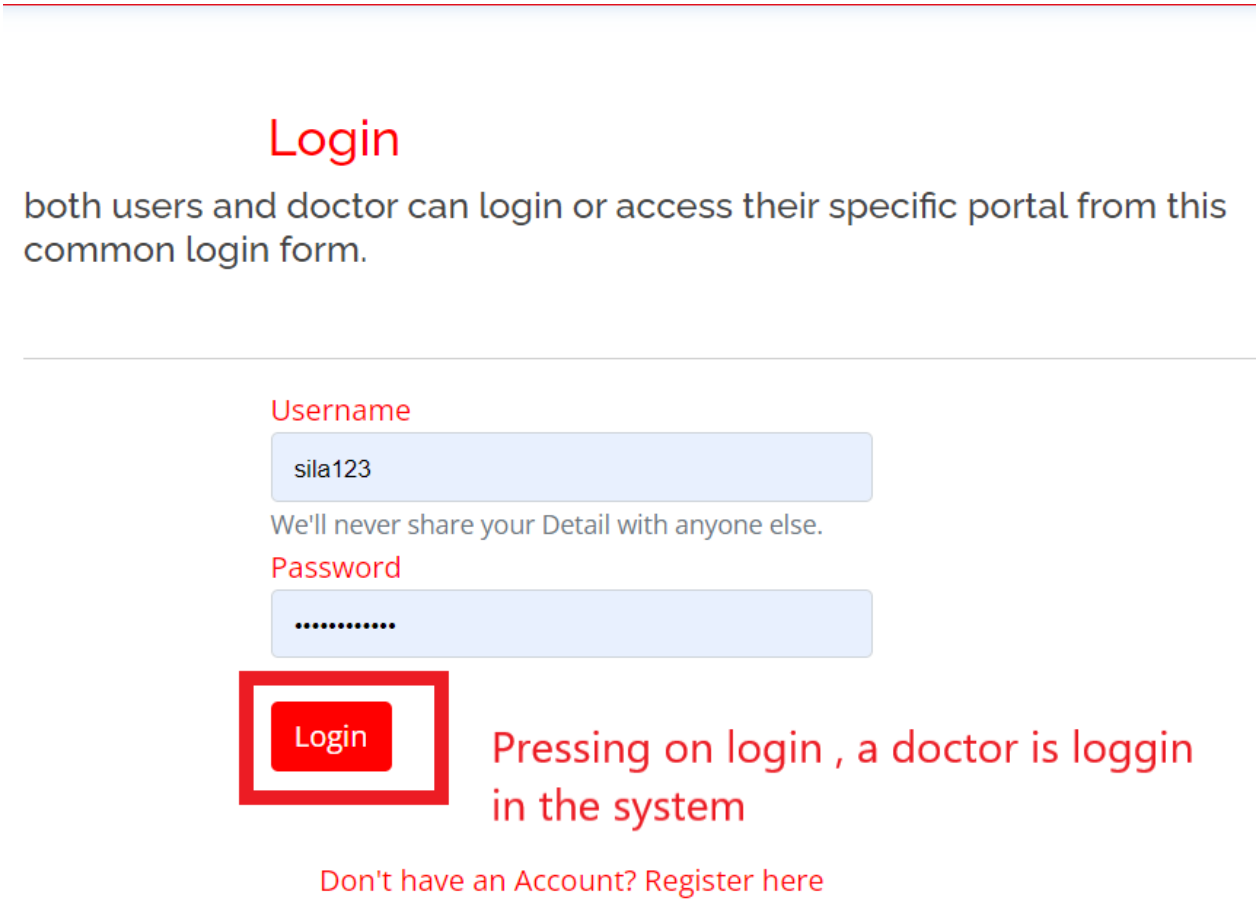

Figure 31 doctor login form test

Figure 32 doctor portal

4. Admin login test

Test Type	System testing
Objective	
Performed action	
Expected outcome	
Obtained outcome	
Test result	

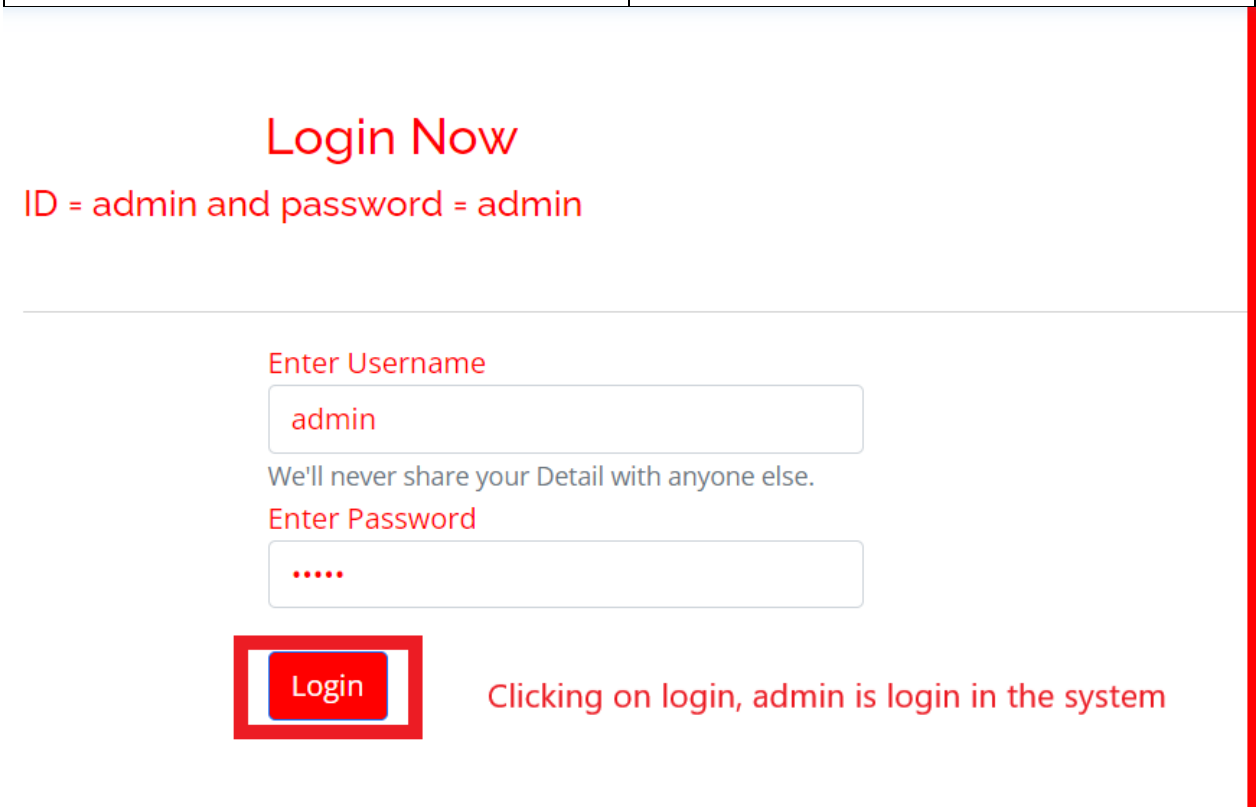

Figure 33 Admin login test

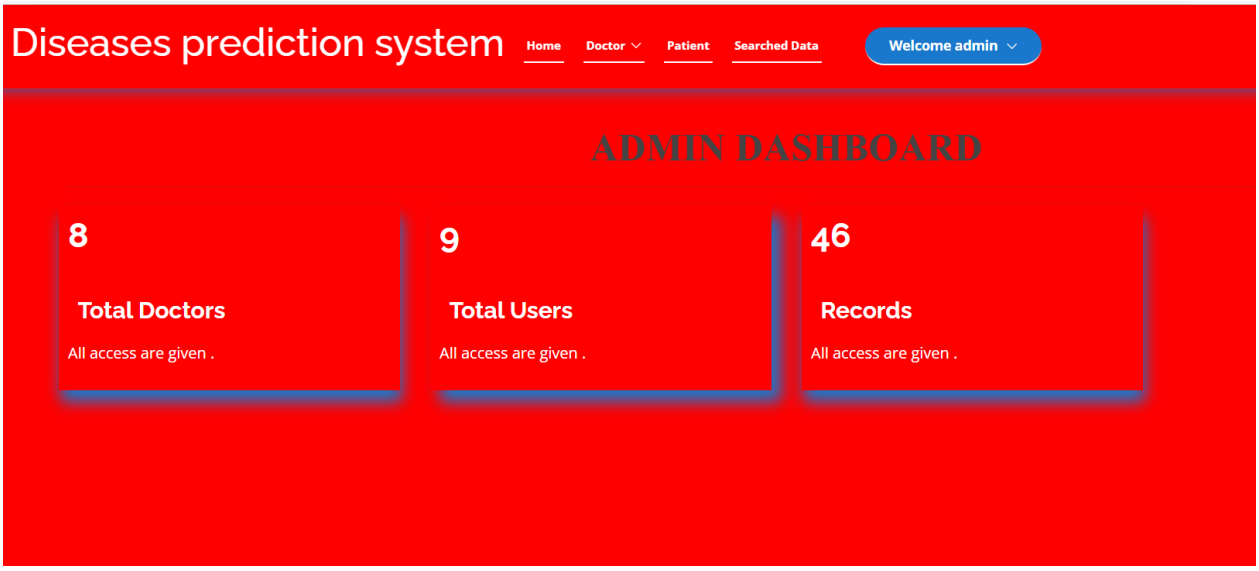

Figure 34 admin dashboard

5. Profile updating test

Test Type	System testing
Objective	To determining whether the update profile features of the system works or not.
Performed action	The profile of doctor and users is update to check by changing details.
Expected outcome	Profile of doctor's and users should be updated after changing the details.
Obtained outcome	Profile of doctor's and the users is changed after updating the details.
Test result	Successful

Table 14 Profile updating test

Before updating profile.

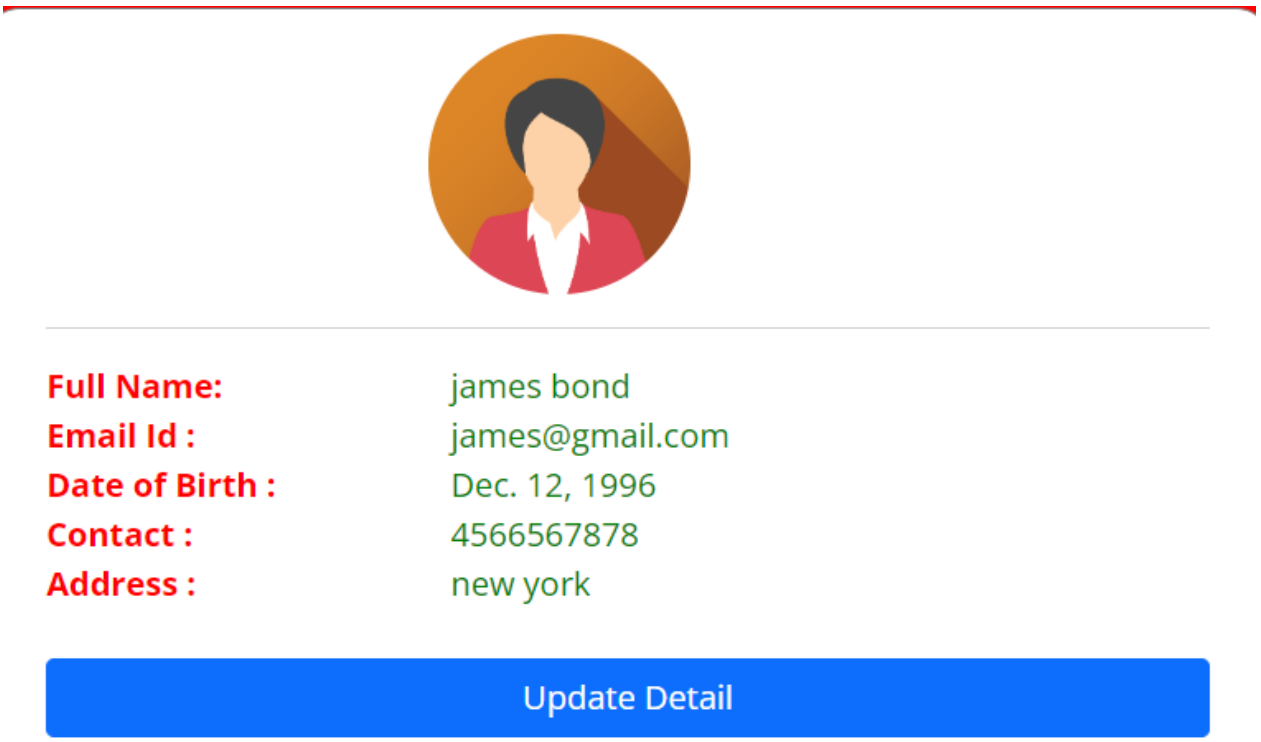

Figure 35 before updating profile

Updating the address detail

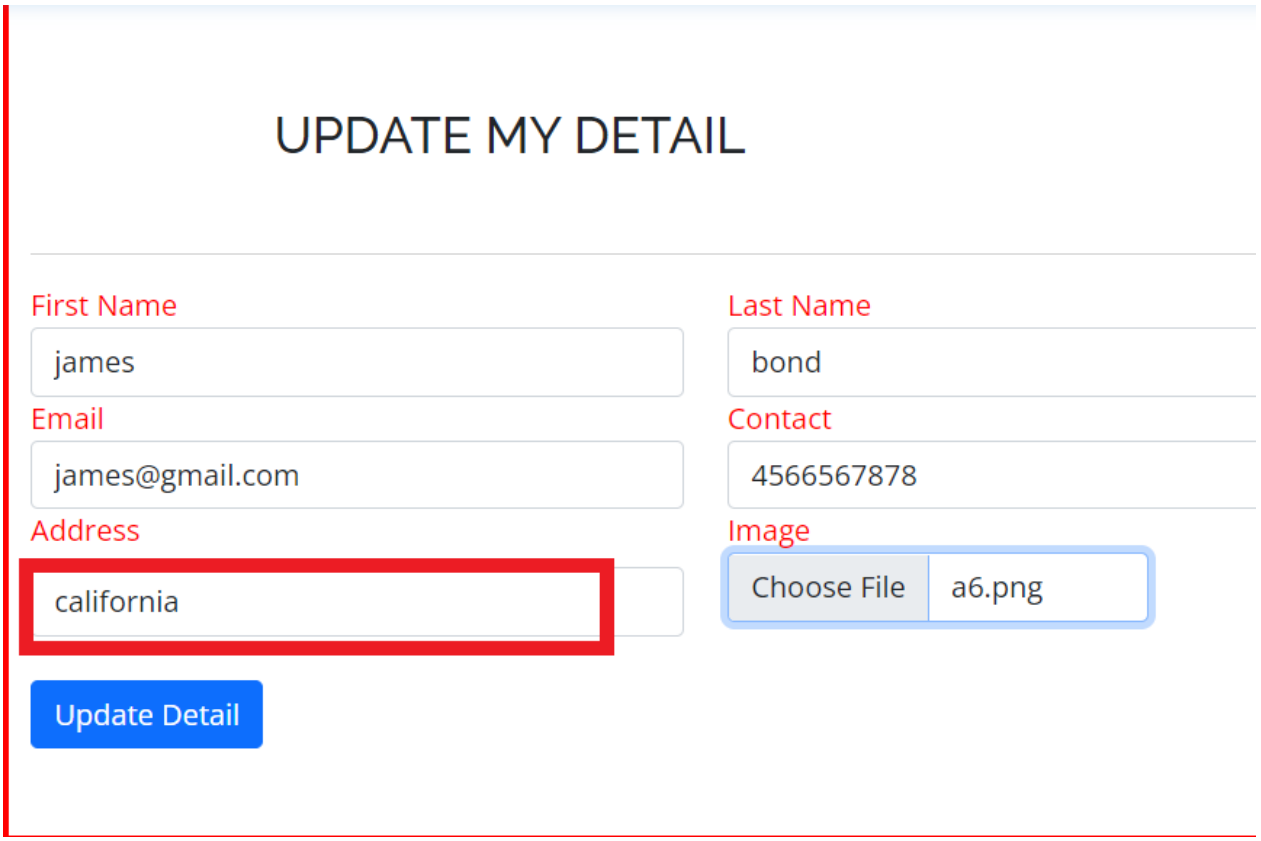

Figure 36 updating address details

After updating address detail.

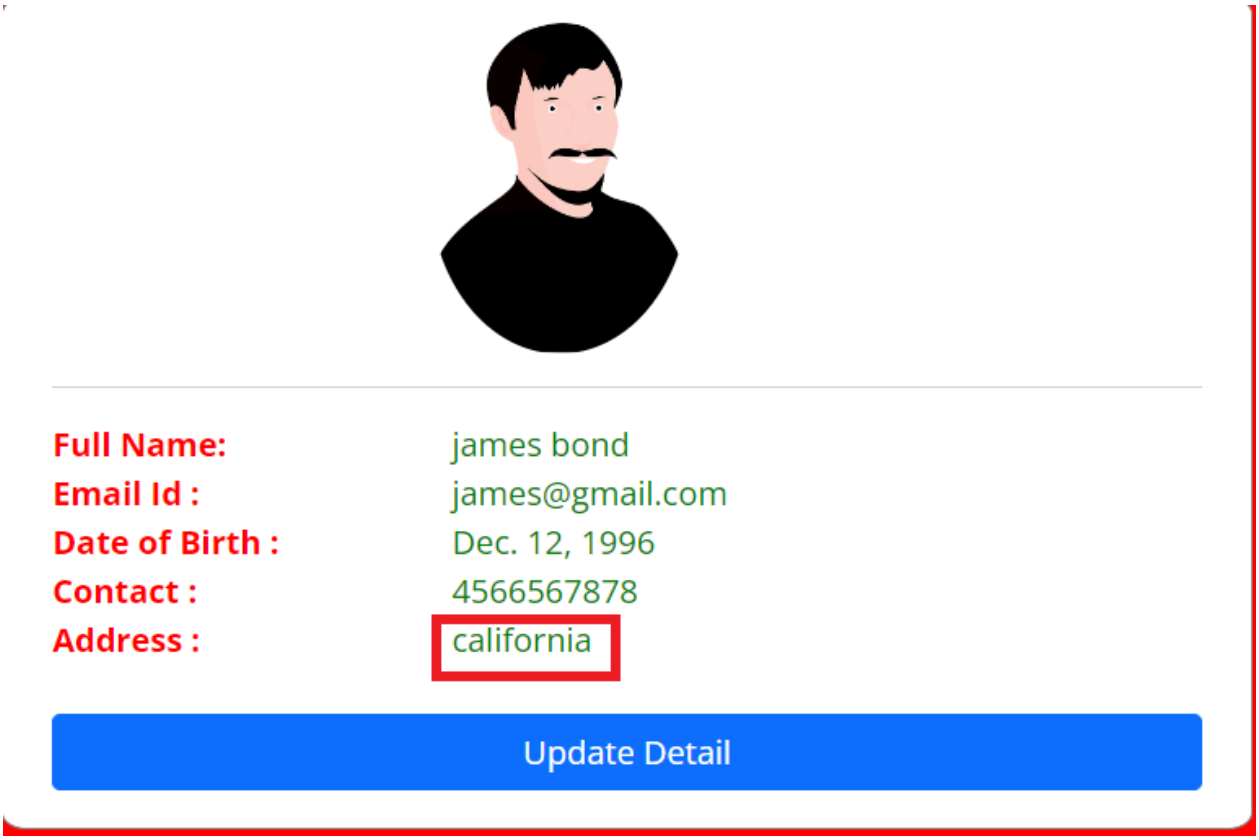

Figure 37 Updated profile

6. Delete functionality testing

Test Type	System testing
Objective	To determine the delete functionality
Performed action	The delete button for deleting the user's record is pressed.
Expected outcome	When admin press the delete button, record should be deleted from the system.
Obtained outcome	Admin is able to delete record successfully.
Test result	Successful.

Table 15 delete functionality testing.

Before deleting

#	Full Name	Image	Email	Contact	Address	Category	Status	Assign	Action
1	tek shrestha		tek@gmail.com	56789	dd	Fungal infection	**Authorized**	Cancel	
2	rpshan rai			5787687	dd	Fungal Infection	**Authorized**	Cancel	
3	y y		y@gmail.com	9811049306	dd	fungal Infection	**Authorized**	Cancel	
4	doctor doc		d@gmail.com	7678	doc	None	**Authorized**	Cancel	

Figure 38 before deleting user id 1 exit

After deleting

1	rpshan rai			5787687	dd	Fungal Infection	**Authorized**	Cancel	
2	y y		y@gmail.com	9811049306	dd	fungal Infection	**Authorized**	Cancel	
3	doctor doc		d@gmail.com	7678	doc	None	**Authorized**	Cancel	

Figure 39 after deleting user id 1

We can see user in id 1 has be deleted.

7. Admin assigning doctor in system test.

Test Type	System testing
Objective	To determine whether the admin can assign doctor's in the system or not.
Performed action	The assigned bottom is pressed.
Expected outcome	Doctor should be assigned in the system.
Obtained outcome	Doctor is assigned in the system and able to login the system with his/her username.
Test result	Successful

Table 16 Admin assigning doctor in the system

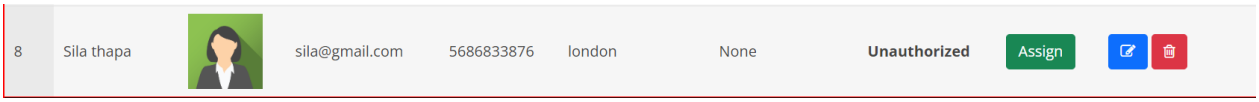

Figure 40 before assigning the doctor in the system

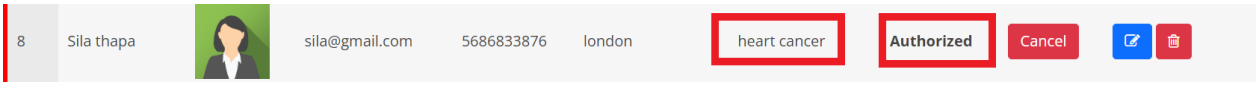

Figure 41 after assigning the doctor in the system

8. Performing prediction test

Test Type	System testing
Objective	To determine whether the prediction functionality works or not / whether users be able to perform prediction based on the symptoms they provide.
Performed action	The multiple symptoms is selected and prediction is performed.
Expected outcome	The system should predict the diseases based on the symptoms provided.
Obtained outcome	System is able to predict diseases successfully.
Test result	Successful

Table 17 Performing prediction test

First user's need to select symptoms as the list of symptoms is listed in the system. When user's select the symptoms and pressed the predict button, the system will correspond the input details with the train model and predict the diseases.

Select symptons that you are facing

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 ○	Cold Hands And Feets ○	Mood Swings ○	Weight Loss ◉
Restlessness ○	Lethargy ○	Patches In Throat ○	Irregular Sugar Level ○
Cough ○	High Fever ◉	Sunken Eyes ○	Breathlessness ○
Sweating ○	Dehydration ○	Indigestion ○	Headache ○
Yellowish Skin ○	Dark Urine ○	Nausea ○	Loss Of Appetite ○
Pain Behind The Eyes ○	Back Pain ○	Constipation ○	Abdominal Pain ○
Diarrhoea ○	Mild Fever ○	Yellow Urine ○	Yellowing Of Eyes ○
Acute Liver Failure ○	Fluid Overload ○	Swelling Of Stomach ○	Swelled Lymph Nodes ○

Figure 42 list of symptoms

Spinning Movements ○	Loss Of Balance ○	Unsteadiness ○	Weakness Of One Body Side ○
Loss Of Smell ○	Bladder Discomfort ○	Continuous Feel Of Urine ○	Passage Of Gases ○
Internal Itching ○	Toxic Look (Typhos) ○	Depression ○	Irritability ○
Muscle Pain ○	Altered Sensorium ○	Red Spots Over Body ○	Belly Pain ○
Abnormal Menstruation ○	Dischromic Patches ○	Watering From Eyes ○	Increased Appetite ○
Polyuria ○	Family History ○	Mucoid Sputum ○	Rusty Sputum ○
Lack Of Concentration ○	Visual Disturbances ○	Receiving Blood Transfusion ○	Receiving Unsterile Injections ○
Coma ○	Stomach Bleeding ○	Distention Of Abdomen ○	History Of Alcohol Consumption ○
Fluid Overload ○	Blood In Sputum ○	Prominent Veins On Calf ○	Palpitations ○
Painful Walking ○	Pus Filled Pimples ○	Blackheads ○	Scurring ○
Skin Peeling ○	Silver Like Dusting ○	Small Dents In Nails ○	Inflammatory Nails ○
Blister ○	Red Sore Around Nose ○	Yellow Crust Ooze ○	Prognosis ○

Predict

Figure 43 list of symptoms

Result after prediction

RESULT AFTER PREDICTION

Model Name	Prediction Output
RandomForestClassifier Prediction	Jaundice
SVC Prediction	Fungal infection
Final Prediction	Fungal infection

Figure 44 Prediction made according to the symptoms provided

9. Testing view functionality

Test Type	System testing
Objective	To determine whether the view functionality works or not.
Performed action	The view record function of in the admin, user's and doctor portal is pressed.
Expected outcome	When view record functionality is pressed, system should display the records.
Obtained outcome	After clicking view record functionality, particular record is displayed successfully
Test result	Successfully

Table 18 testing view functionality

Figure 45 displaying user record on clicking

Figure 46 displaying all user's record on clicking

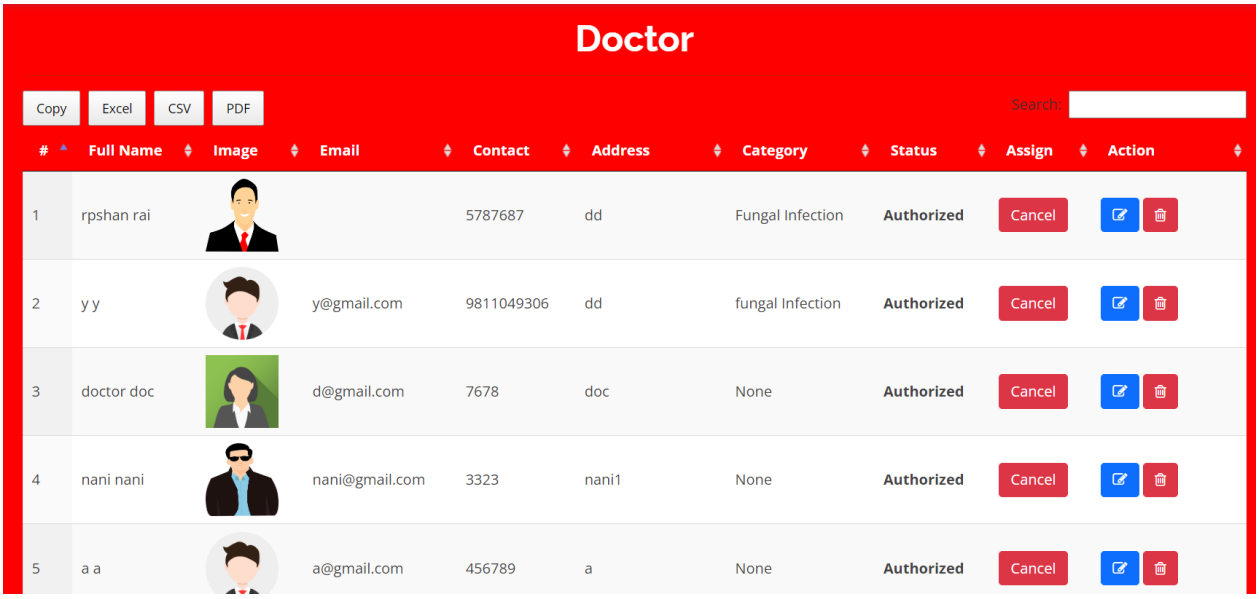

Figure 47 displaying all doctor's on clicking

CHAPTER 5 – CONCLUSION

5.1. Overview

A reliable, versatile, and effective smart health prediction system was developed. The technology develops an appealing user interface and which provided promising accuracy. The system is connected to a large dataset that is trained using machine learning methods, and it is capable of predicting 135 different sorts of diseases. In this research, SVM and Random forest, two well-known classification methods, are combined together. The system consist of efficient features where users can registered, login, view record, manage profile, make prediction etc.

5.1. Summary of the investigation study

Both algorithms result was satisfactory. When both algorithms are individually tested in the jupyter notebook, the accuracy of the SVM was 93.49% and the accuracy of the random forest was 99.45. Comparatively, it can be noticed that the random forest provided a better prediction then that of the SVM but when both algorithms are combined, 95.39% accuracy. The model was created in jupyter notebook to study the nature of dataset. From the jupyter notebook it is noted that there is no strong dependencies as well as correlation between the variables.

5.2. Findings and recommendation

After conducting research and taking this project's findings into practice, I discovered that many problems in other domains can be resolved if the right data is gathered and then evaluated using

data mining techniques. Similarly, I observed that creating an effective system requires a precise dataset and suitable algorithms.

However, doing project in medical sector is exceedingly delicate. Making the system with unreliable data could worsen user health. Therefore, it is essential to get data reviewed by medical professionals before beginning any initiative in the health sector.

5.3. Future work

Well, this system is merely a prototype designed to show how far the domain of "health prediction" can advance. This system was created specifically to meet the criteria for my final dissertation. But now that the system has been built, I'm astonished by how accurate it is. I had planned to advance this project significantly in the future. I will use my own initiative effort to gather tens of thousands of data points from medical sources like pharmacies and hospitals, and I will have those points confirmed by various medical professionals because accurate data is essential for the development of an accurate system. I'll organize a research team and work with a healthcare facility like a hospital or the NHS. I'll create a true algorithm that honestly predicts diseases based on their symptoms. I'll include a lot of features that will make it simple for people to access health services. It will resemble a virtual hospital where various medical specialties and researchers will be linked. Users will be able to schedule appointments with doctors and communicate with doctors online. Users will be able to receive doctor recommendations from the system based on their specific medical needs. To eliminate the need for the administrator to manually enter data into the system, the system will store all the information provided by users and train itself automatically.

5.4. Personal evaluation

Although the system offers good accuracy, it performs slowly. According to my research, the medical industry is extremely sensitive, making it potentially unsafe for users to rely on such systems because they may show the same ailments even though different persons have identical symptoms caused by distinct conditions. It may be risky to rely on someone else's data about our health since diseases depend on more than just symptoms. They also depend on the user's age, medical history, and surroundings. However, diligent utilization of data across a broad range may possibly be able to make correct predictions.

CHAPTER 6 – REFERENCE

References

Boog, J., 2022. *What Is DSDM (Dynamic Systems Development Method)?.* [Online] Available at: https://theqalead.com/development-devops-agile/dsdm-dynamic-systems-development-method/
[Accessed 12 06 2022].

Chauhan, N. S., 2022. *Decision Tree Algorithm, Explained.* [Online] Available at: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
[Accessed 06 05 2022].

Deloitte, 2020. *The socio-economic impact of AI in healthcare.* [Online] Available at: https://www.medtecheurope.org/wp-content/uploads/2020/10/mte-ai_impact-in-healthcare_oct2020_report.pdf
[Accessed 26 09 2022].

Donges, N., 2021. *Random Forest Algorithm: A Complete Guide.* [Online] Available at: https://builtin.com/data-science/random-forest-algorithm
[Accessed 12 06 2022].

frontiers, 2022. *legal and ethical consideration in artificial intelligence in healthcare.* [Online] Available at: https://www.frontiersin.org/articles/10.3389/fsurg.2022.862322/full
[Accessed 1 09 2022].

Gandhi, R., 2018. *Naive Bayes Classifier.* [Online] Available at: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[Accessed 10 05 2022].

Gupta, S., 2022. *quotium.* [Online] Available at: https://www.quotium.com/performance/dsdm-project-lifecycle/
[Accessed 13 06 2022].

IBM, 2020. *Random Forest.* [Online] Available at: https://www.ibm.com/cloud/learn/random-forest
[Accessed 8 06 2022].

IBM, 2021. *K-Nearest Neighbors Algorithm.* [Online] Available at: https://www.ibm.com/uk-en/topics/knn [Accessed 12 05 2022].

java T point , 2021. *K-Nearest Neighbor(KNN) Algorithm for machine learning.* [Online] Available at: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning [Accessed 05 05 2022].

L. Mohan, J. P. P. S. a. A. K., 2020. Support Vector Machine Accuracy Improvement with classification. In: *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN).* Bhimtal, India: IEEE, pp. 477-481.

M. A. Jabbar, S. S., 2016. *Heart disease prediction system based on hidden naïve bayes classifier*. Bangalore, India , IEEE .

Martin, M., 2022. *Prototype Model in software engineering.* [Online] Available at: https://www.guru99.com/software-engineering-prototyping-model.html [Accessed 12 06 2022].

McGregor, M., 2020. *Freecodecamp.* [Online] Available at: https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/ [Accessed 08 05 2022].

scikit-learn, 2022. *Scikit-learn.* [Online] Available at: https://scikit-learn.org/stable/getting_started.html [Accessed 08 07 2022].

scit-learn , 2021. *Decision Trees.* [Online] Available at: https://scikit-learn.org/stable/modules/tree.html [Accessed 05 06 2022].

Shahadat Uddin, L. H. H. L. M. A. M. &. E. G., 2022. Comparative performance analysis of K-nearest neighbour (KNN) algorithms and its different variants for disease prediction. *Comparative performance analysis of K-nearest neighbour (KNN) algorithms and its different variants for disease prediction,* 15 04, p. 12.

softwaretesting , 2022. *software testing.* [Online] Available at: https://www.softwaretestinghelp.com/system-testing/ [Accessed 20 08 2022].

Tech Target, 2021. *unit testing.* [Online] Available at: https://www.techtarget.com/searchsoftwarequality/definition/unit-testing [Accessed 26 8 2022].

toolshero, 2022. *Rational Unified Process (RUP).* [Online] Available at: https://www.toolshero.com/information-technology/rational-unified-process-rup/#:~:text=Rational%20Unified%20Process%20(RUP)%20is,%2C%20implementation%2C%20testing%20and%20application. [Accessed 09 06 2022].

tutorialspoint , 2022. *SDLC - RAD Model.* [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm [Accessed 12 06 2022].

Victor chang, Y. S. Y. C. T. L., 2019. *researchgate.* [Online] Available at: https://www.researchgate.net/publication/333407135_Smart_Healthcare_and_Ethical_Issues [Accessed 12 09 2022].

Yiu, T., 2019. *Understanding Random Forest.* [Online] Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 [Accessed 12 06 2022].

Youn-jung Son, H.-G. K. E.-H. K. S. C. S.-K. L., 2010. Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients. *Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients*, 31 12, pp. 253-259.

CHAPTER 7 – BIBLIOGRAPHY

Bibliography

Boog, J., 2022. *What Is DSDM (Dynamic Systems Development Method)?.* [Online] Available at: https://theqalead.com/development-devops-agile/dsdm-dynamic-systems-development-method/
[Accessed 12 06 2022].

Chauhan, N. S., 2022. *Decision Tree Algorithm, Explained.* [Online] Available at: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
[Accessed 06 05 2022].

Deloitte, 2020. *The socio-economic impact of AI in healthcare.* [Online] Available at: https://www.medtecheurope.org/wp-content/uploads/2020/10/mte-ai_impact-in-healthcare_oct2020_report.pdf
[Accessed 26 09 2022].

Donges, N., 2021. *Random Forest Algorithm: A Complete Guide.* [Online] Available at: https://builtin.com/data-science/random-forest-algorithm
[Accessed 12 06 2022].

frontiers, 2022. *legal and ethical consideration in artificial intelligence in healthcare.* [Online] Available at: https://www.frontiersin.org/articles/10.3389/fsurg.2022.862322/full
[Accessed 1 09 2022].

Gandhi, R., 2018. *Naive Bayes Classifier.* [Online] Available at: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c
[Accessed 10 05 2022].

Gupta, S., 2022. *quotium.* [Online] Available at: https://www.quotium.com/performance/dsdm-project-lifecycle/
[Accessed 13 06 2022].

IBM, 2020. *Random Forest.* [Online] Available at: https://www.ibm.com/cloud/learn/random-forest
[Accessed 8 06 2022].

IBM, 2021. *K-Nearest Neighbors Algorithm.* [Online] Available at: https://www.ibm.com/uk-en/topics/knn [Accessed 12 05 2022].

java T point , 2021. *K-Nearest Neighbor(KNN) Algorithm for machine learning.* [Online] Available at: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning [Accessed 05 05 2022].

L. Mohan, J. P. P. S. a. A. K., 2020. Support Vector Machine Accuracy Improvement with classification. In: *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN).* Bhimtal, India: IEEE, pp. 477-481.

M. A. Jabbar, S. S., 2016. *Heart disease prediction system based on hidden naïve bayes classifier*. Bangalore, India , IEEE .

Martin, M., 2022. *Prototype Model in software engineering.* [Online] Available at: https://www.guru99.com/software-engineering-prototyping-model.html [Accessed 12 06 2022].

McGregor, M., 2020. *Freecodecamp.* [Online] Available at: https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/ [Accessed 08 05 2022].

scikit-learn, 2022. *Scikit-learn.* [Online] Available at: https://scikit-learn.org/stable/getting_started.html [Accessed 08 07 2022].

scit-learn , 2021. *Decision Trees.* [Online] Available at: https://scikit-learn.org/stable/modules/tree.html [Accessed 05 06 2022].

Shahadat Uddin, L. H. H. L. M. A. M. &. E. G., 2022. Comparative performance analysis of K-nearest neighbour (KNN) algorithms and its different variants for disease prediction. *Comparative performance analysis of K-nearest neighbour (KNN) algorithms and its different variants for disease prediction,* 15 04, p. 12.

softwaretesting , 2022. *software testing.* [Online] Available at: https://www.softwaretestinghelp.com/system-testing/ [Accessed 20 08 2022].

Tech Target, 2021. *unit testing.* [Online] Available at: https://www.techtarget.com/searchsoftwarequality/definition/unit-testing [Accessed 26 8 2022].

toolshero, 2022. *Rational Unified Process (RUP).* [Online] Available at: https://www.toolshero.com/information-technology/rational-unified-process-rup/#:~:text=Rational%20Unified%20Process%20(RUP)%20is,%2C%20implementation%2C%20testing%20and%20application.
[Accessed 09 06 2022].

tutorialspoint , 2022. *SDLC - RAD Model.* [Online] Available at: https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm [Accessed 12 06 2022].

Victor chang, Y. S. Y. C. T. L., 2019. *researchgate.* [Online] Available at: https://www.researchgate.net/publication/333407135_Smart_Healthcare_and_Ethical_Issues [Accessed 12 09 2022].

Yiu, T., 2019. *Understanding Random Forest.* [Online] Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 [Accessed 12 06 2022].

Youn-jung Son, H.-G. K. E.-H. K. S. C. S.-K. L., 2010. Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients. *Application of Support Vector Machine for Prediction of Medication Adherence in Heart Failure Patients*, 31 12, pp. 253-259.

CHAPTER 8- APPENDIX: DESIGNS

8.1. Gantt chart

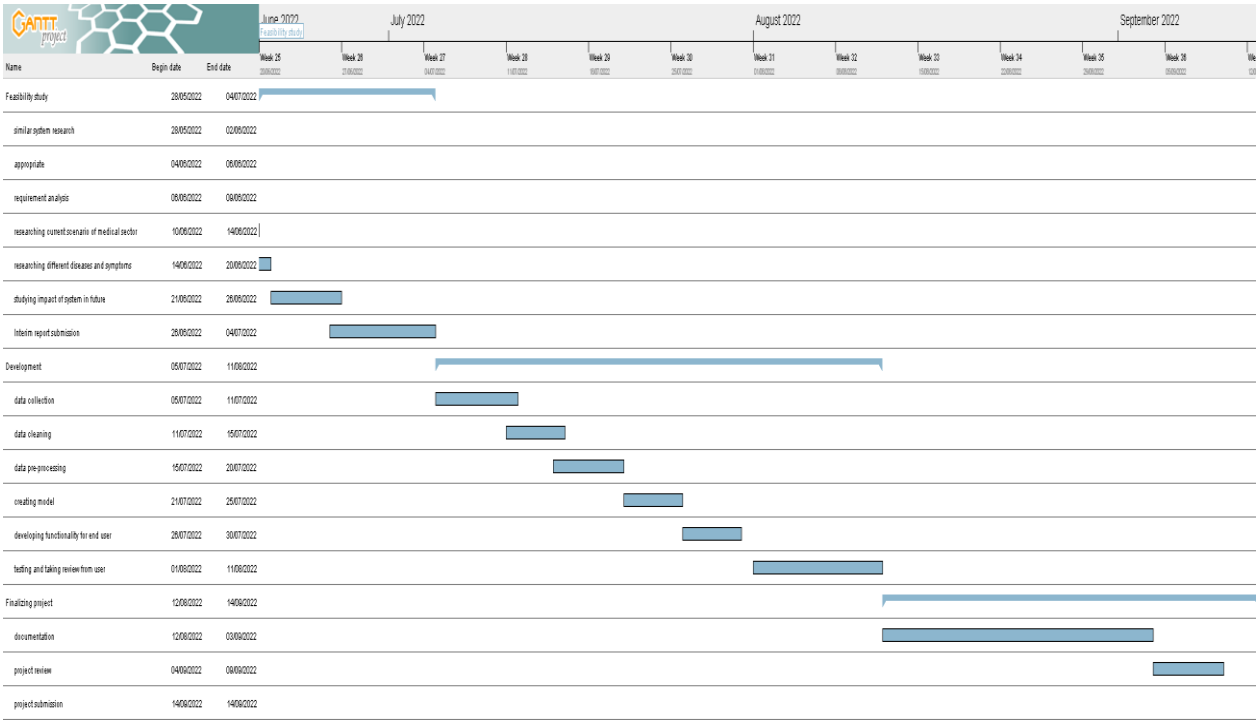

Figure 48 Gantt chart

8.2. Work breakdown Structure (WBS)

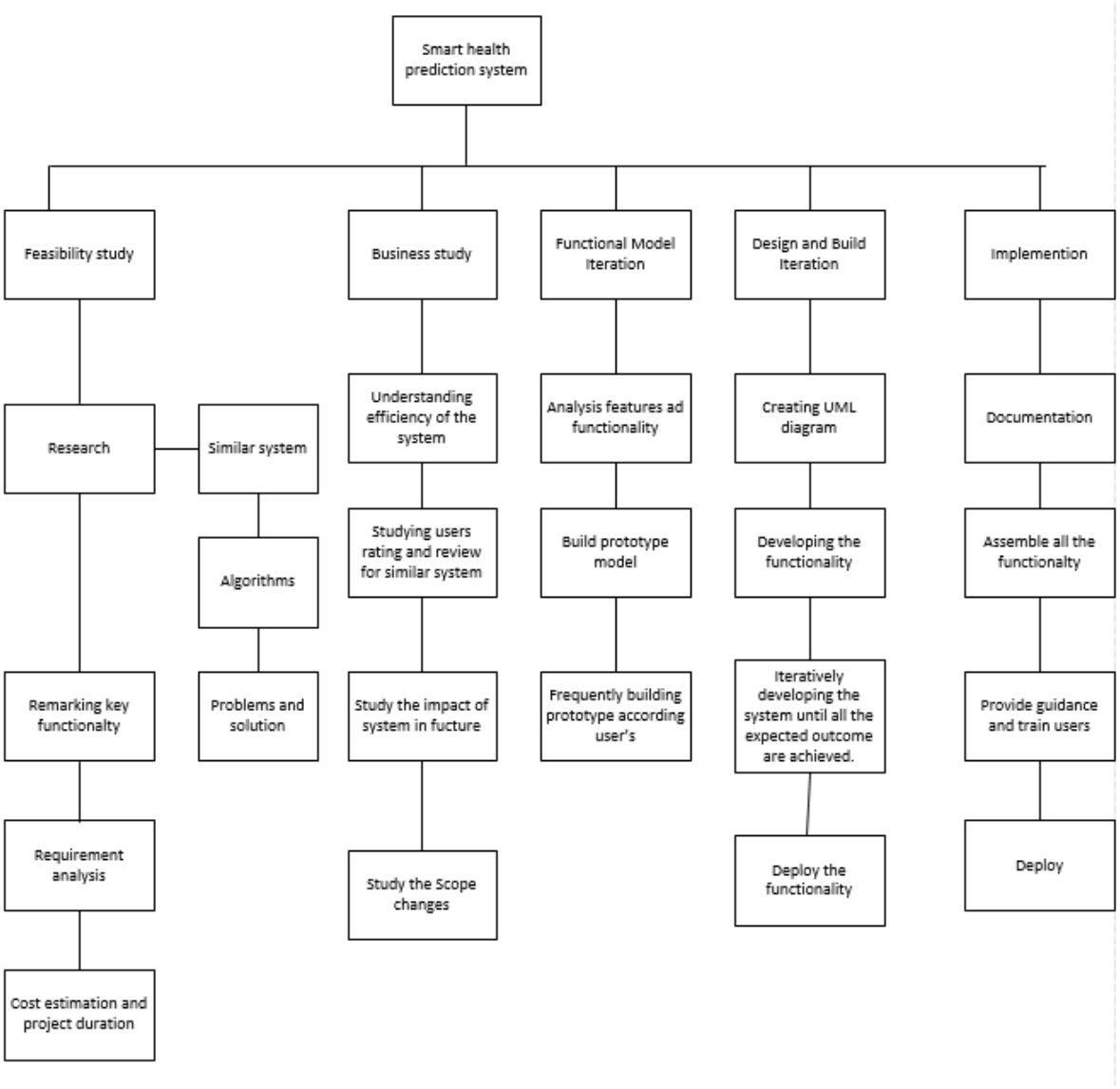

Figure 49 Work breakdown structure

8.3. Use case diagram

Use case diagram represent the interaction between the system and the actor's. It represent the relationship between the system and the actors. It consist of various use cases and actor. In this project there are three actor, patient, doctor and admin.

8.3.1. Admin and system use case diagram

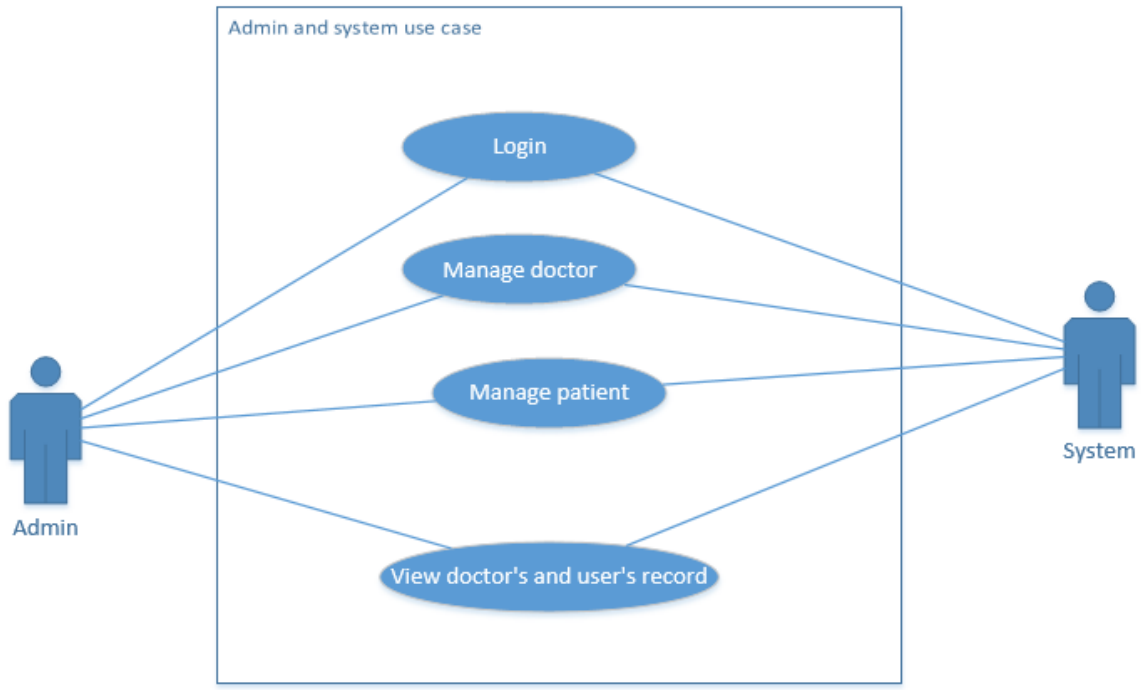

Figure 50 Use case diagram of admin and system

8.3.2. Patient and system use case diagram

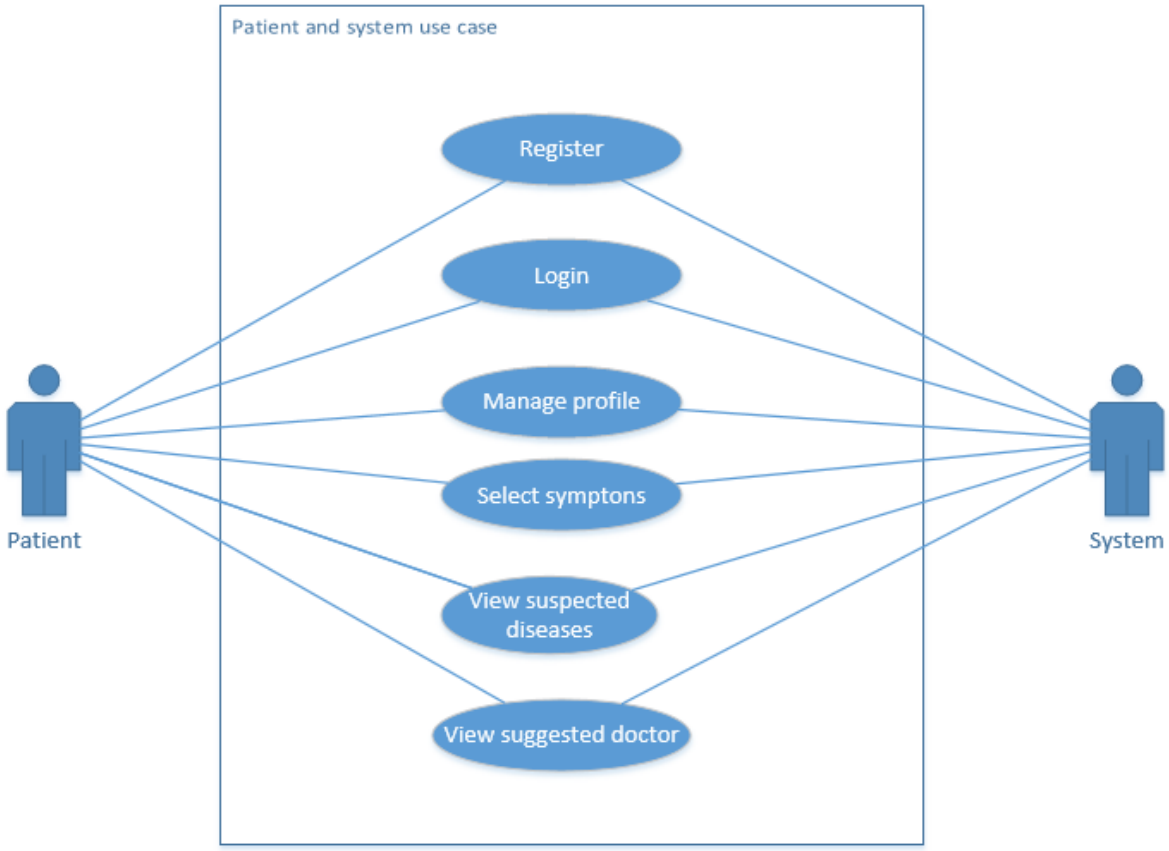

Figure 51 Use case diagram of patient and the system

8.3.3. Doctor and the system use case diagram

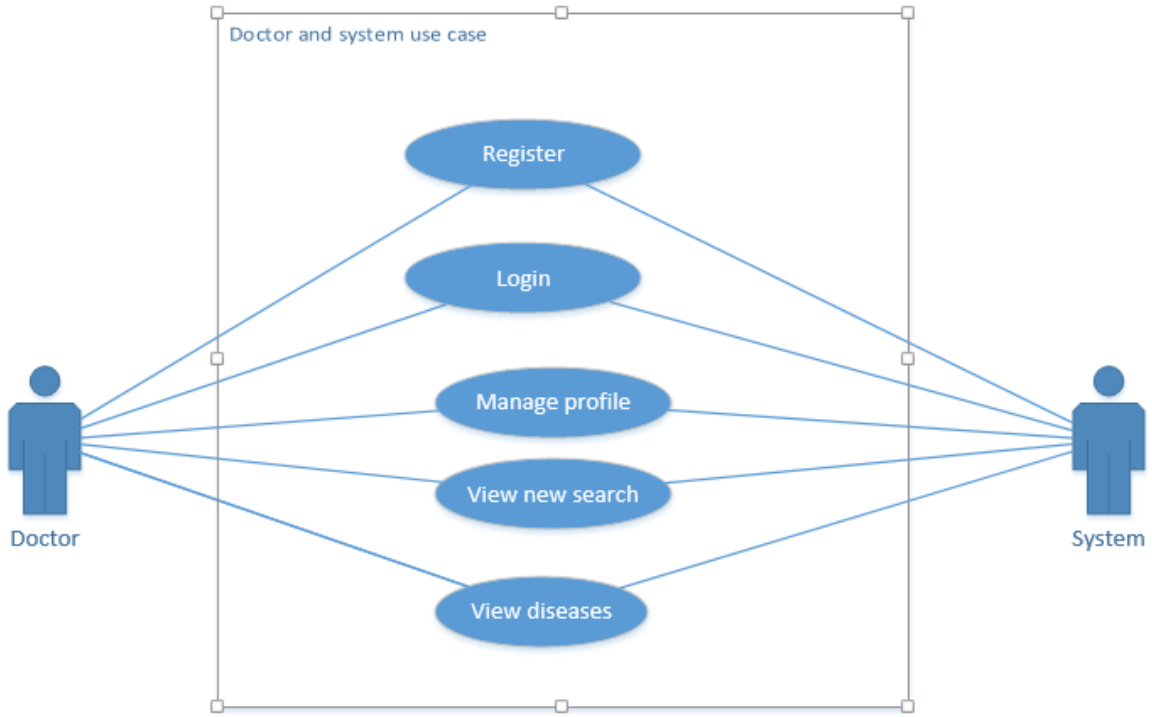

Figure 52 Use case diagram of doctor and system

8.4. Algorithm for users for using the system

Algorithm for performing the prediction

Step 1: Run program

Step: 2 Registered with username and password

Step: 3 Login the system with the registered username and password

Step: 4 Click on the prediction

Step: 5 select the symptoms that you are facing

Step: 6 Stop

Algorithm for viewing previous predicted history

Step 1: Run program

Step 2: Login the system with username and password

Step 3: Click on the view record

Step 4: stop

8.5. Data flow Diagram

Data Flow Diagrams (DFDs) are visual representations of the "flow" of information within a database or other type of information system. One further tool for depicting Data Processing is the data flow diagram. Typically, a designer will begin by sketching a DFD at the context level, which depicts the system's relationship to its environment. This DFD is "exploded" to reveal additional information about the modelled system at the context level.

A data flow diagram (DFD) is used to depict how bits of information move through a system. Problem analysis often employs the usage of data flow diagrams. From its perspective, a system is a function that maps inputs to outputs. Data flow diagrams (DFDs) depict how information travels through a system's many transformations and operations.

From the placement of an order to its shipment and subsequent restocking, the development of any system can be deduced with the help of a dataflow diagram. a recorded list that is kept in a database by the appropriate authorities .

Level : 0

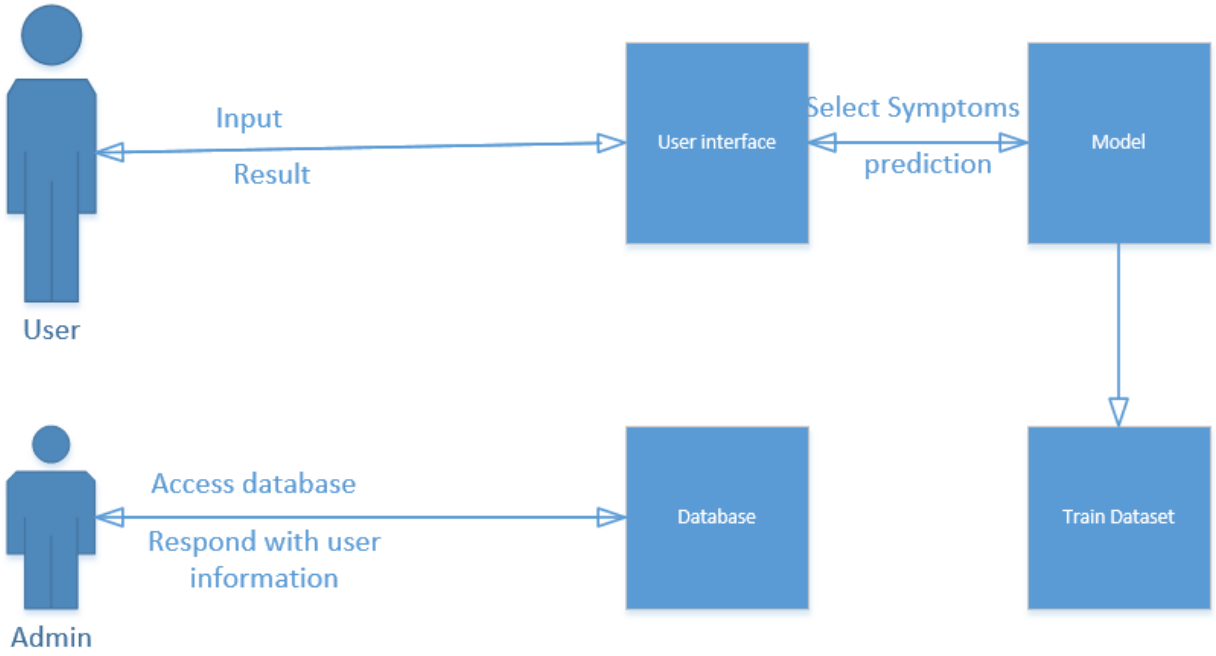

Figure 53 DFD (data flow diagram)

Level 1

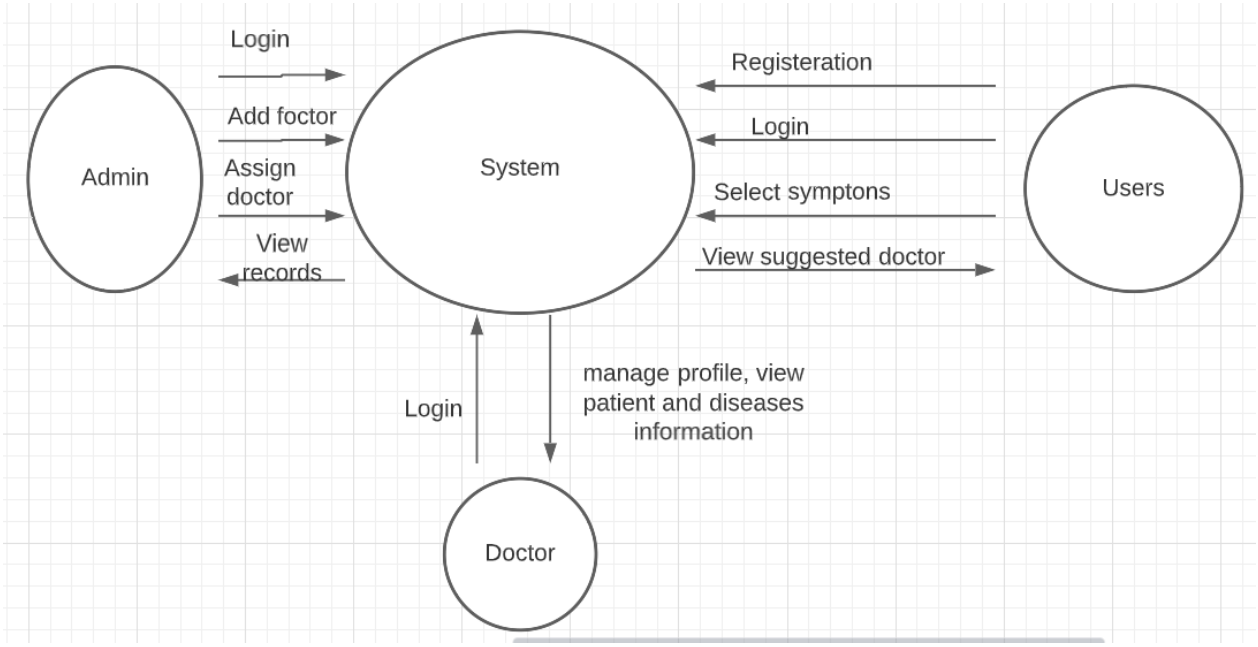

Figure 54 Level 1 dfd diagram

CHAPTER 9: APPENDIX: Sample code

9.1. Sample code of the UI

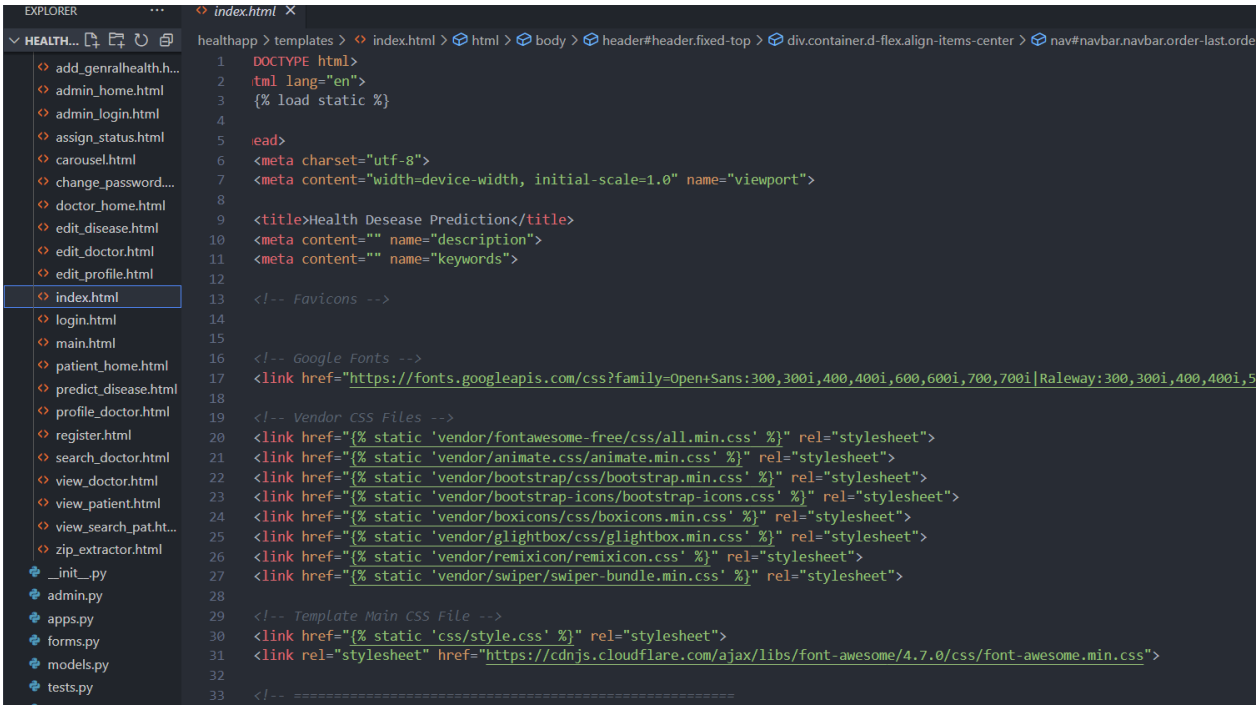

Figure 55 sample code of the UI

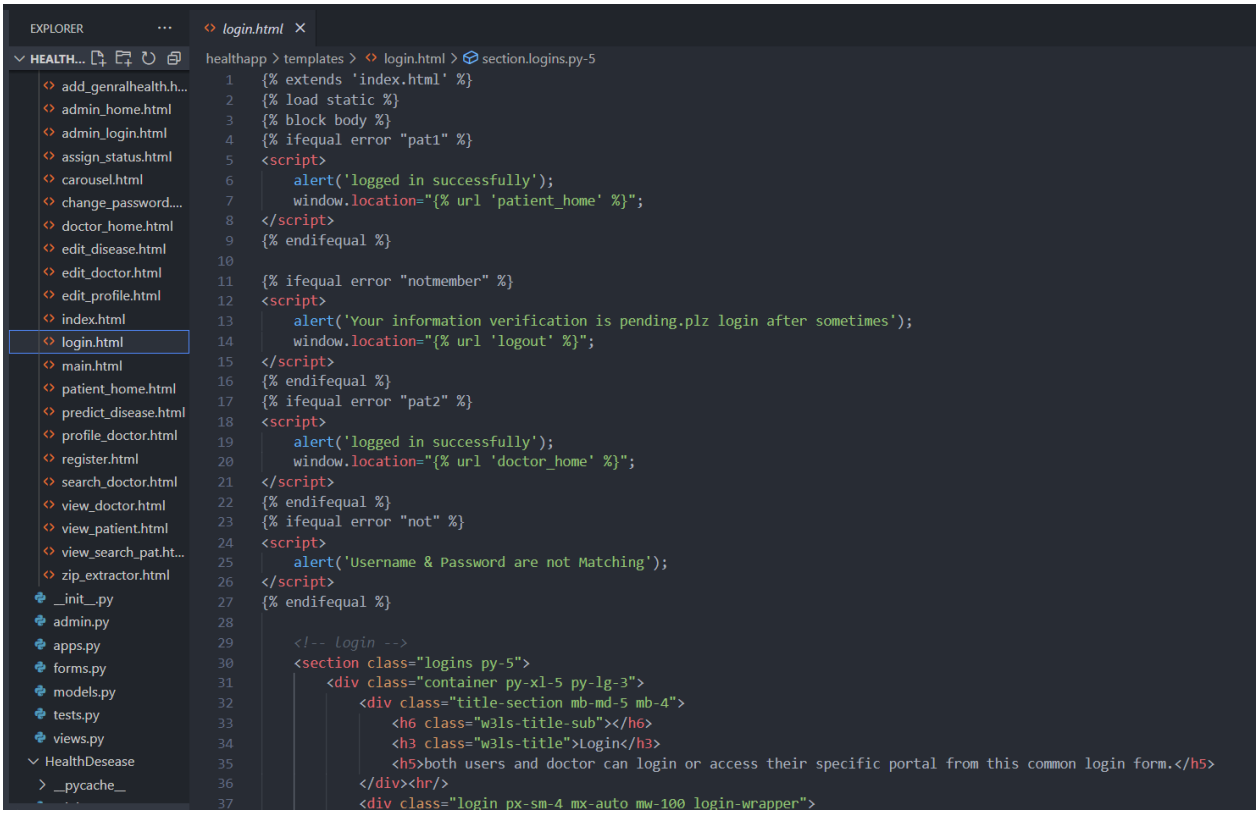

Figure 56 Sample code of the UI

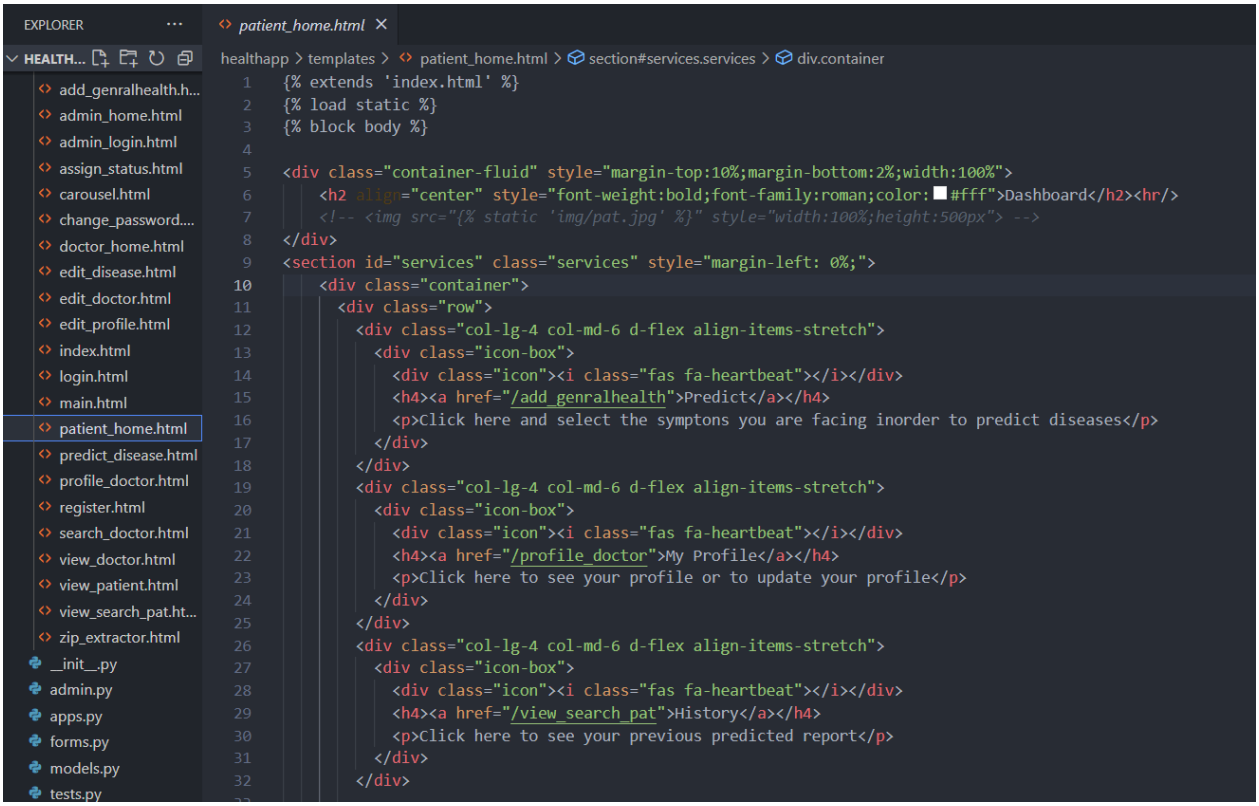

Figure 57 Sample code of the UI

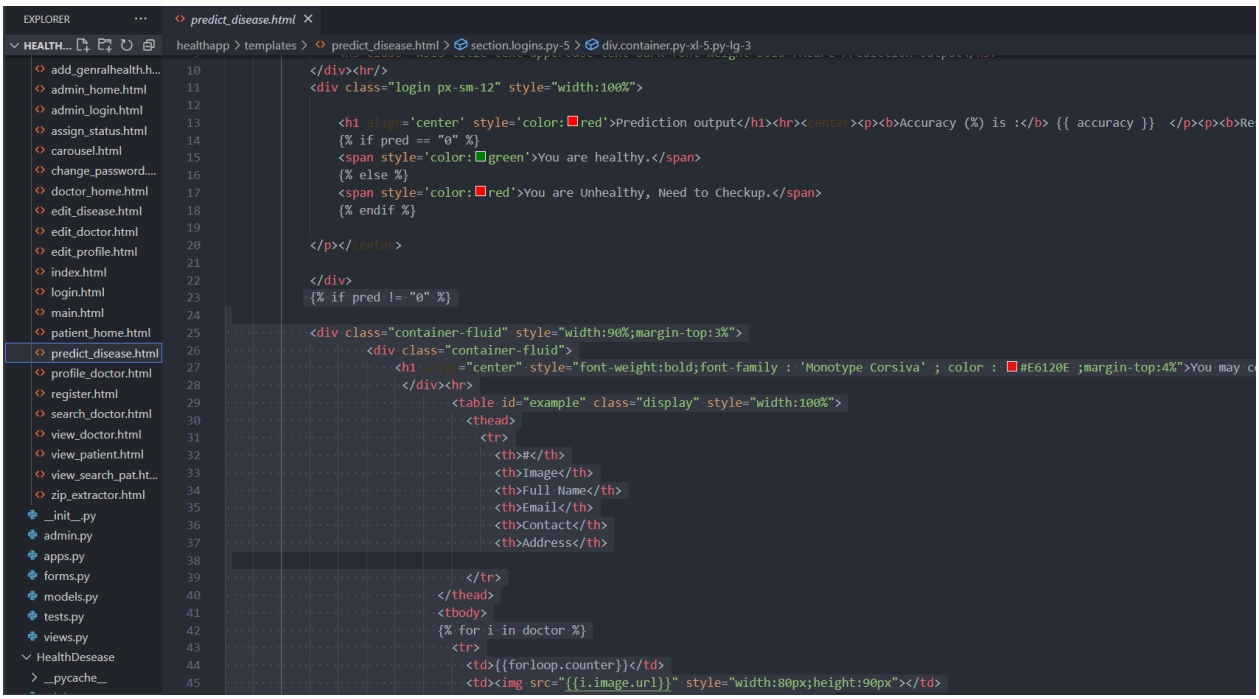

Figure 58 Sample code of the UI

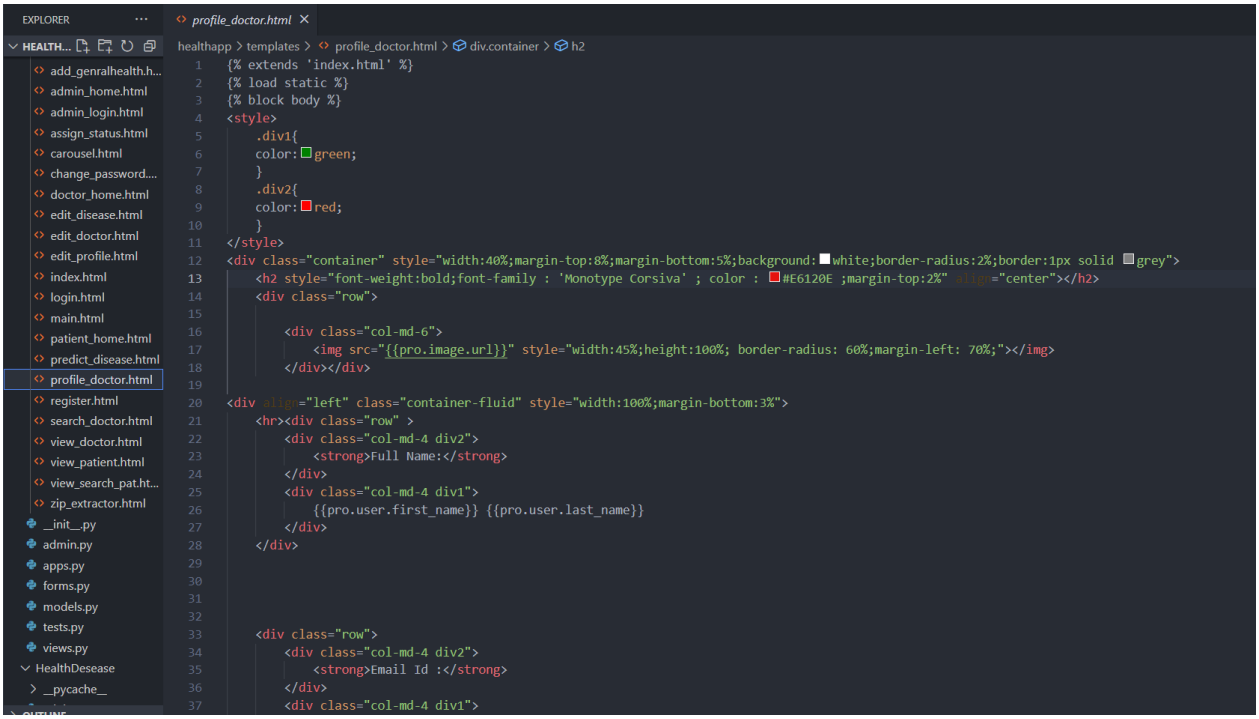

Figure 59 Sample code of the UI

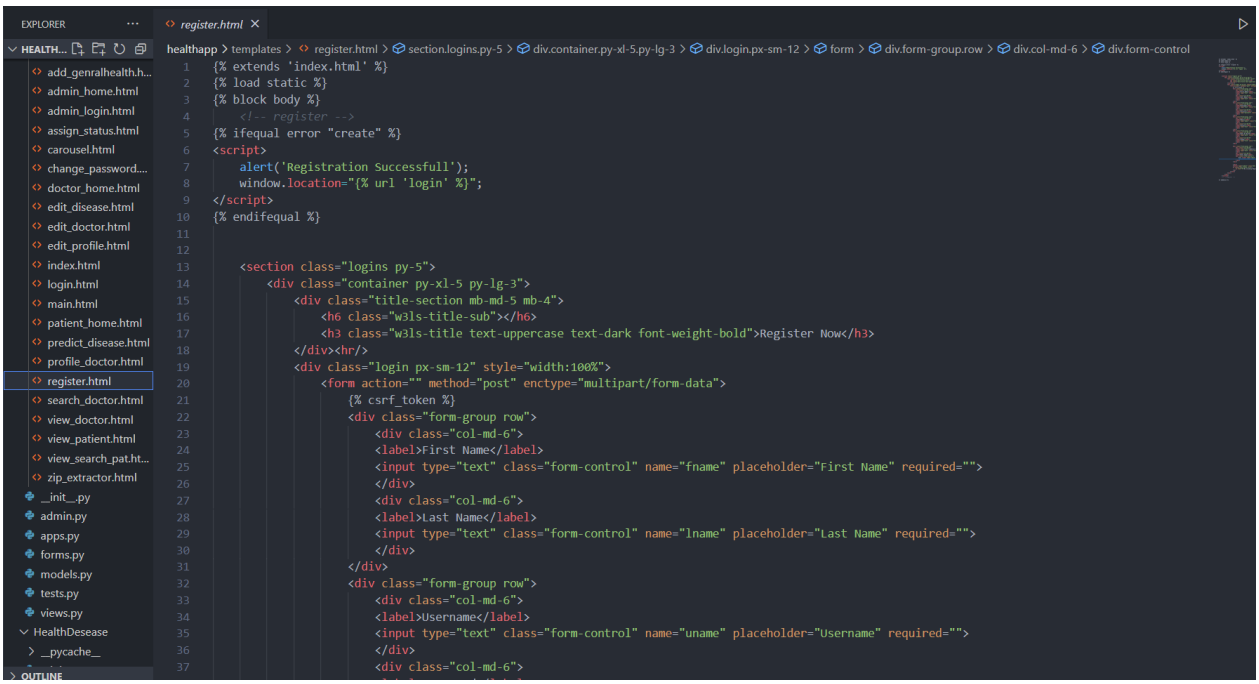

Figure 60 Sample code of the UI

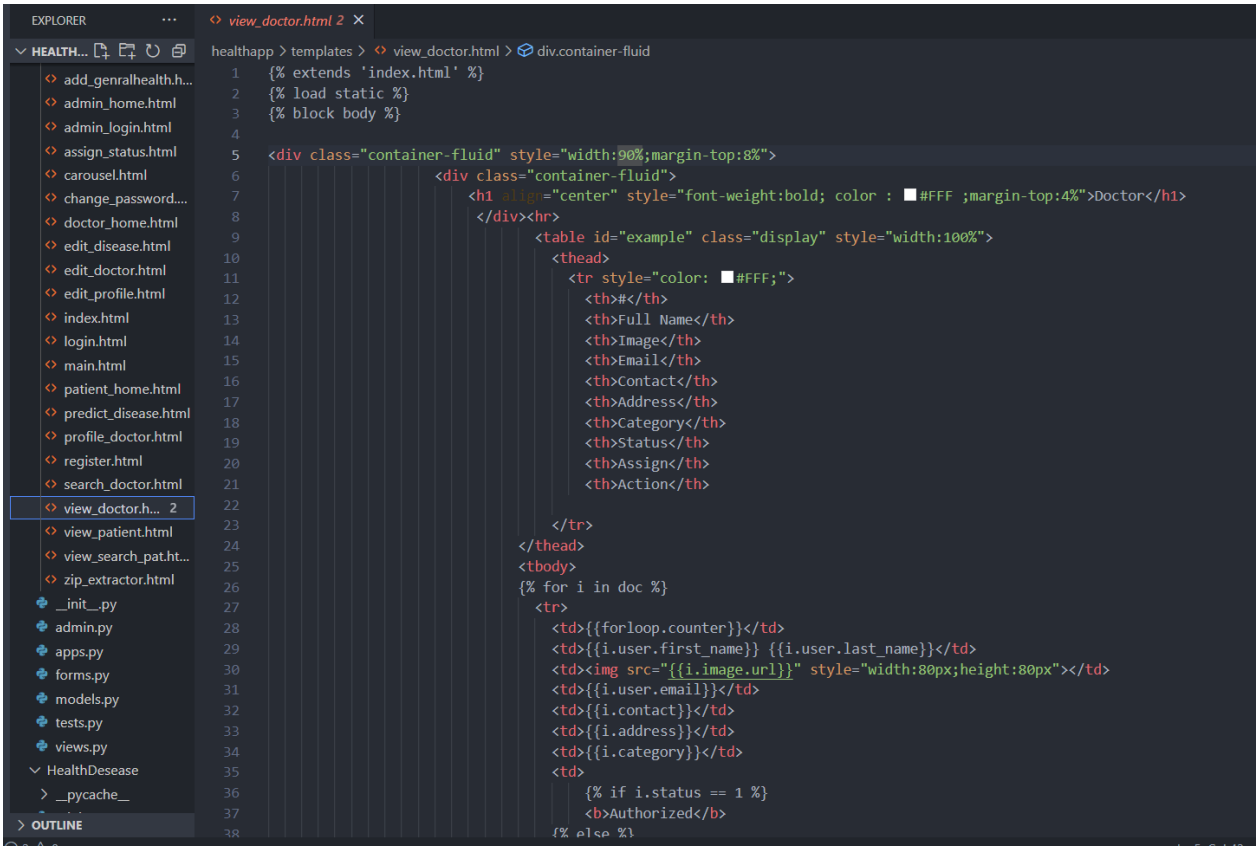

Figure 61 Sample code of the UI

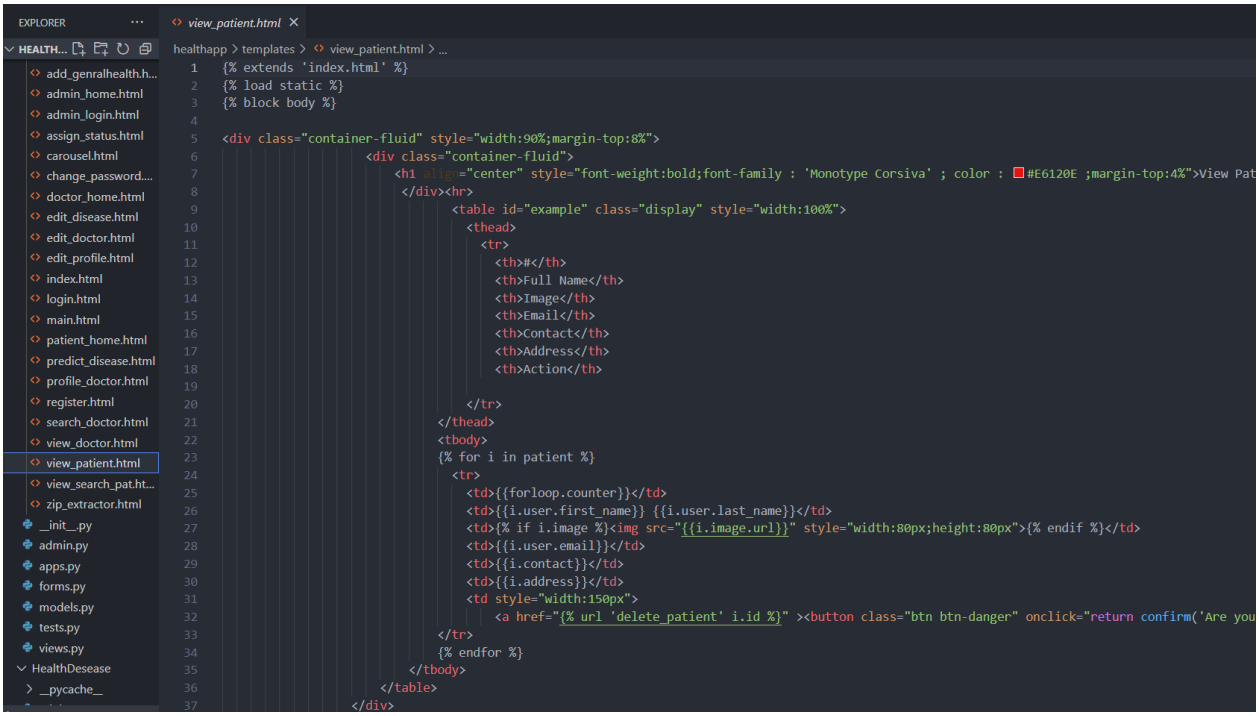

Figure 62 Sample code of the UI

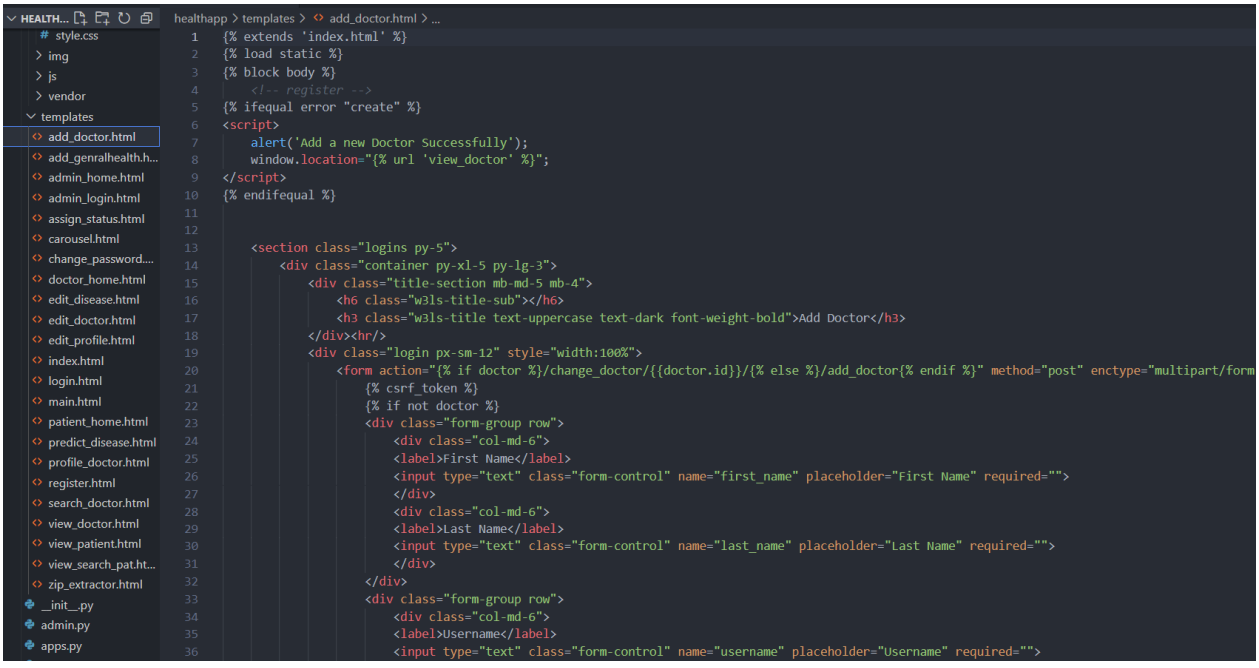

Figure 63 Sample code of the UI

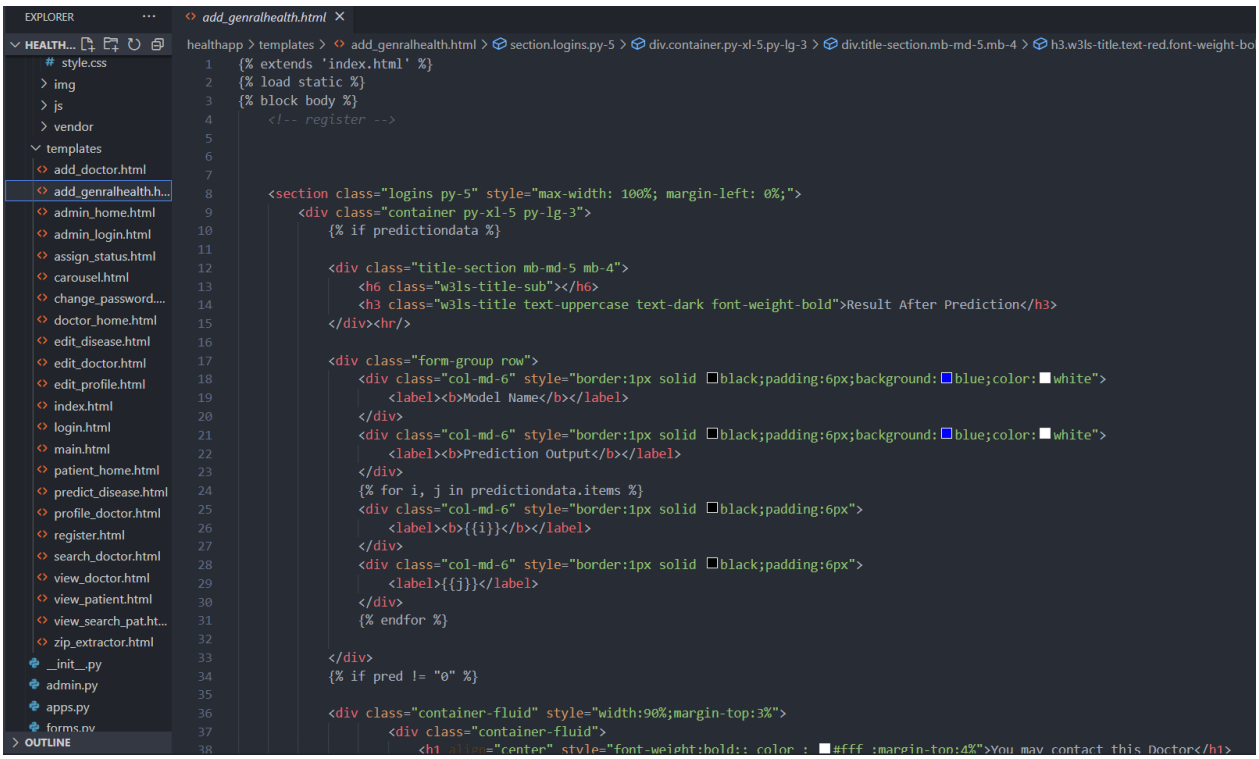

Figure 64 Sample code of the UI

Figure 65 Sample code of the UI

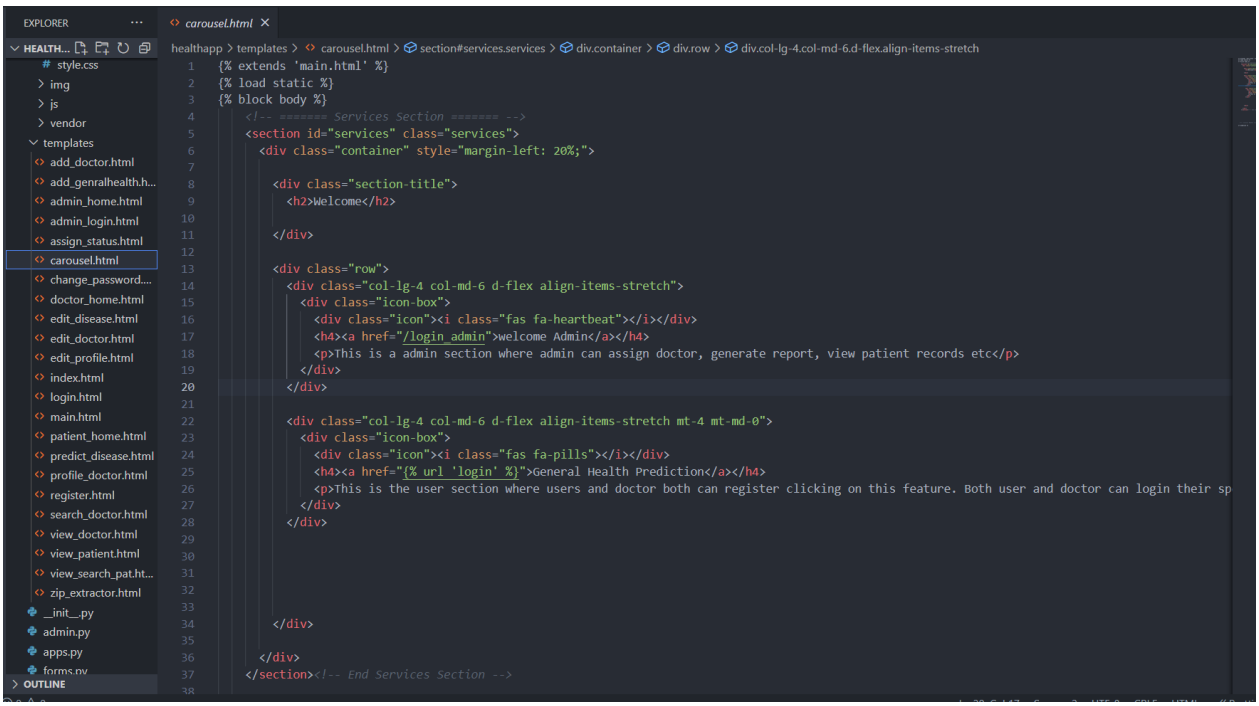

Figure 66 Sample code of the UI

9.2. Sample code of the backend script.

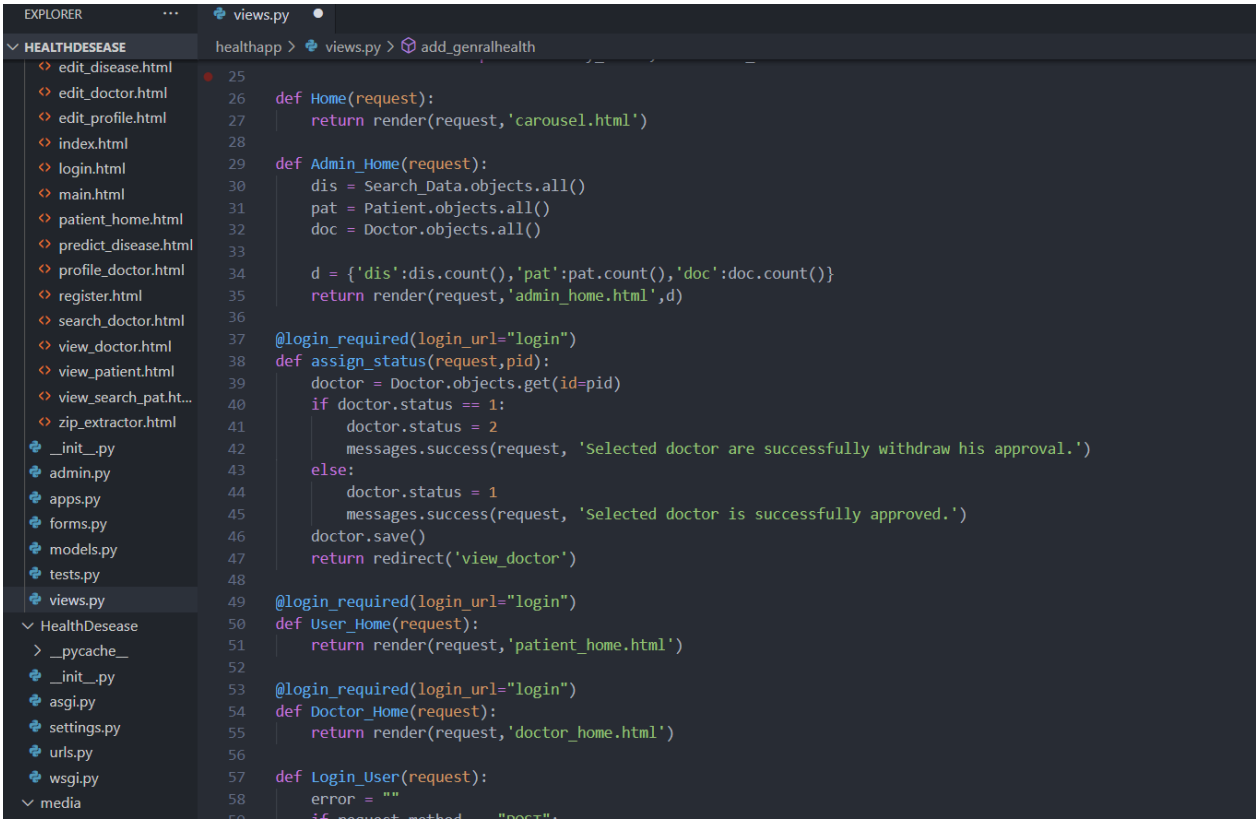

Figure 67 Sample code of the backend script

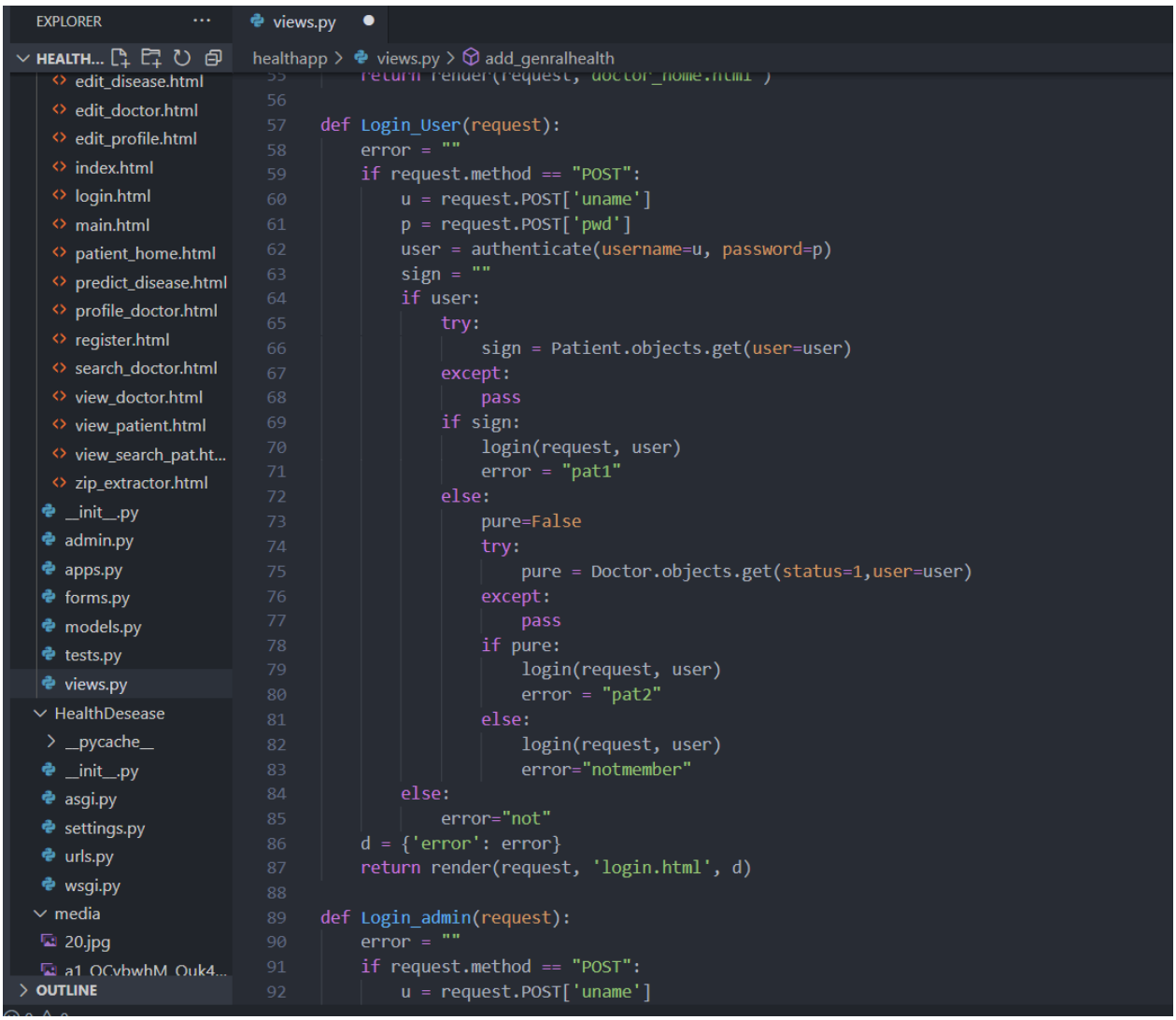

Figure 68 Sample code of the backend script

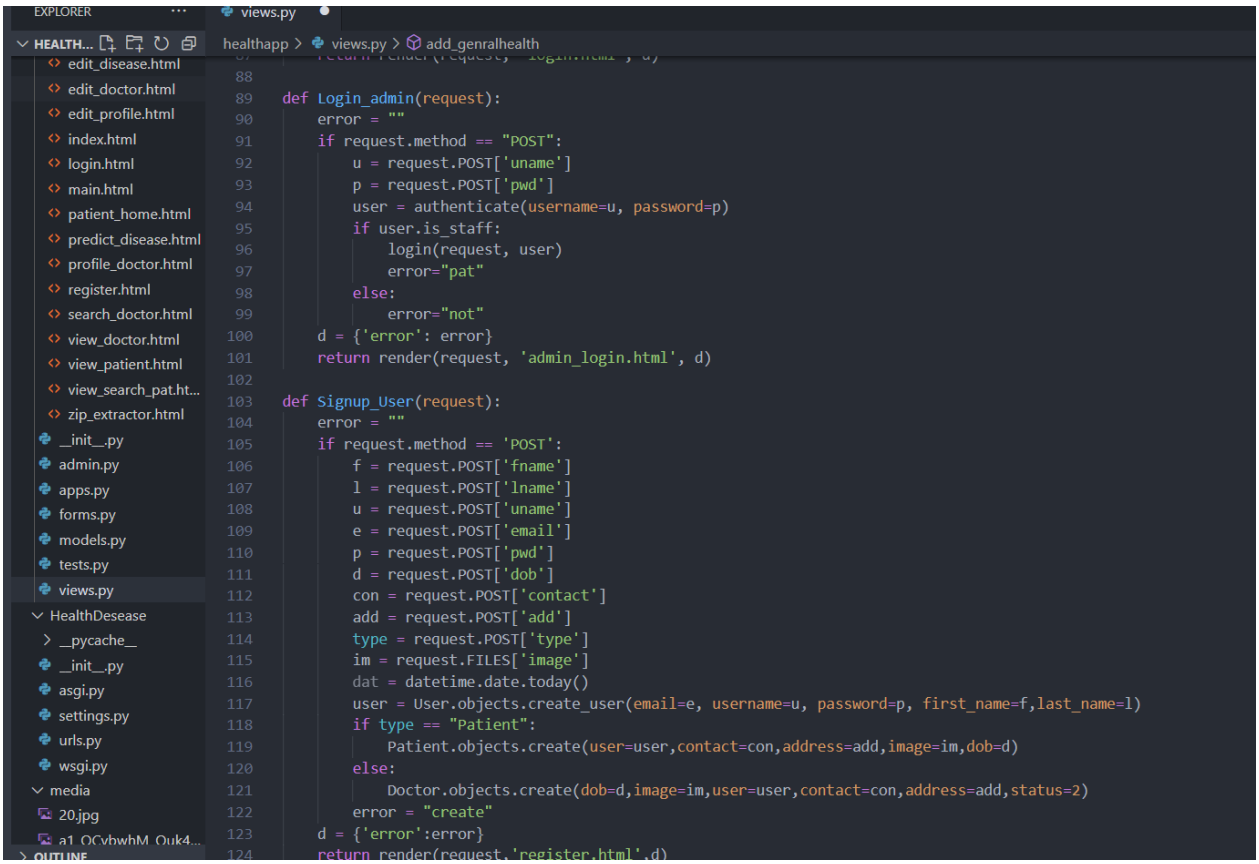

Figure 69 Sample code of the backend script

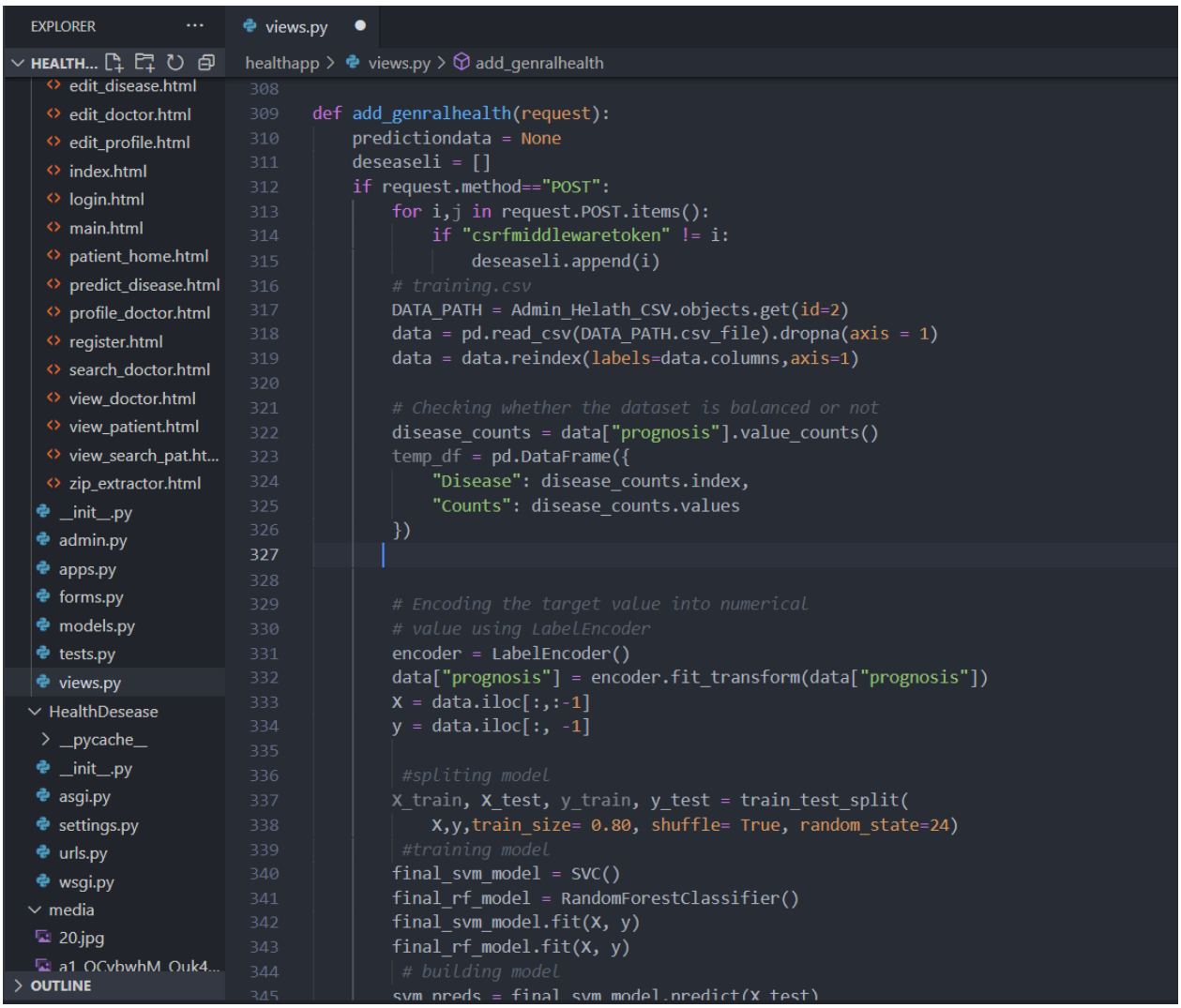

Figure 70 Sample code of the backend

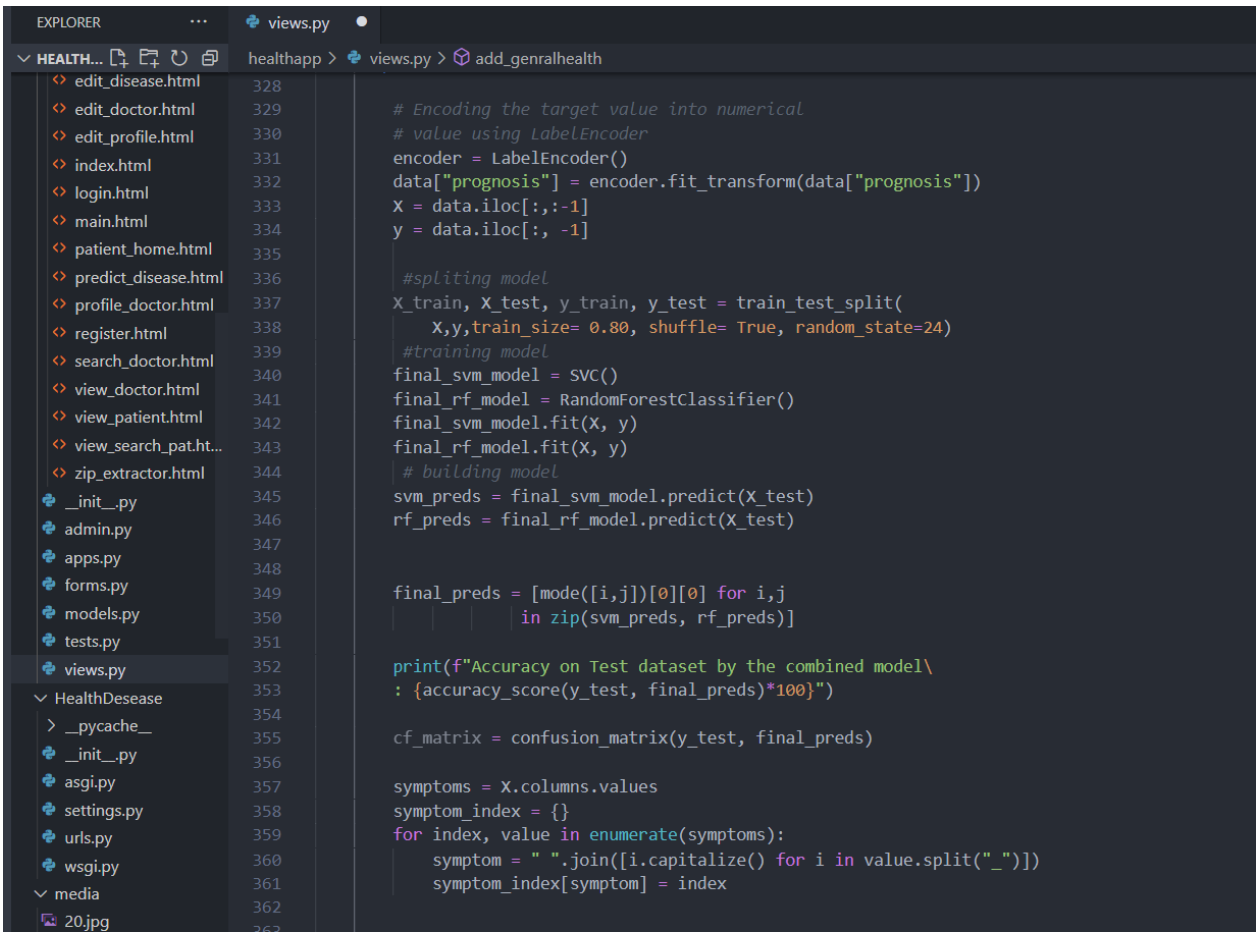

Figure 71 Sample code of the backend

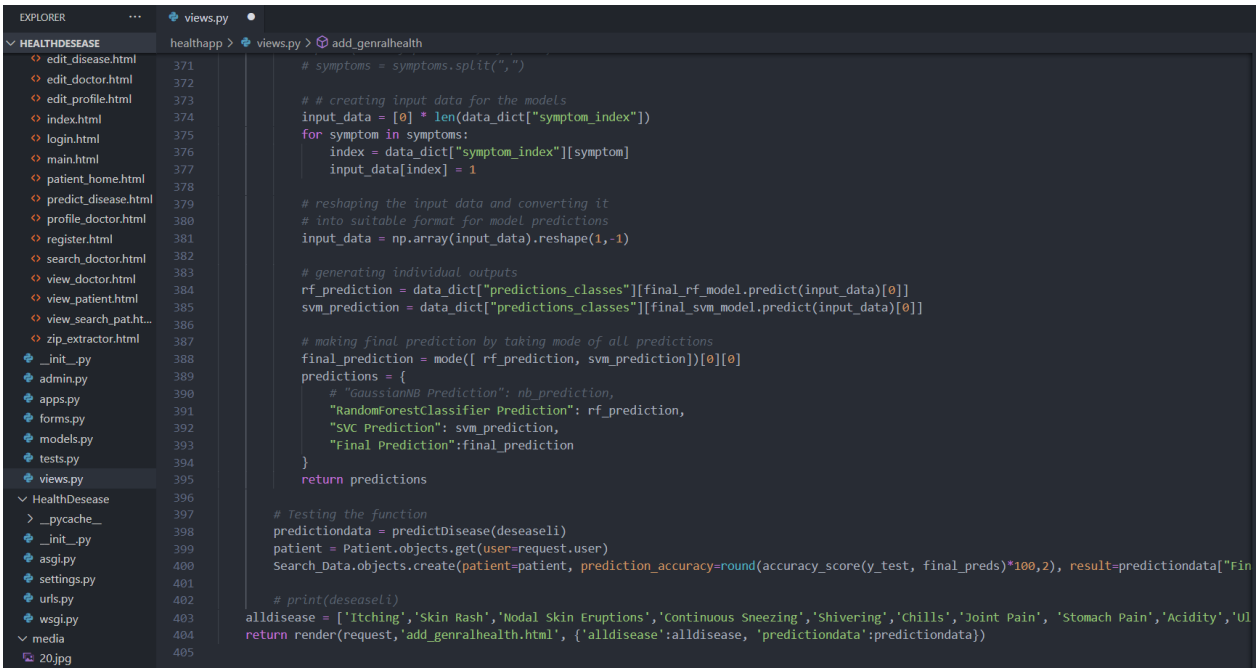

Figure 72 Sample code of the backend

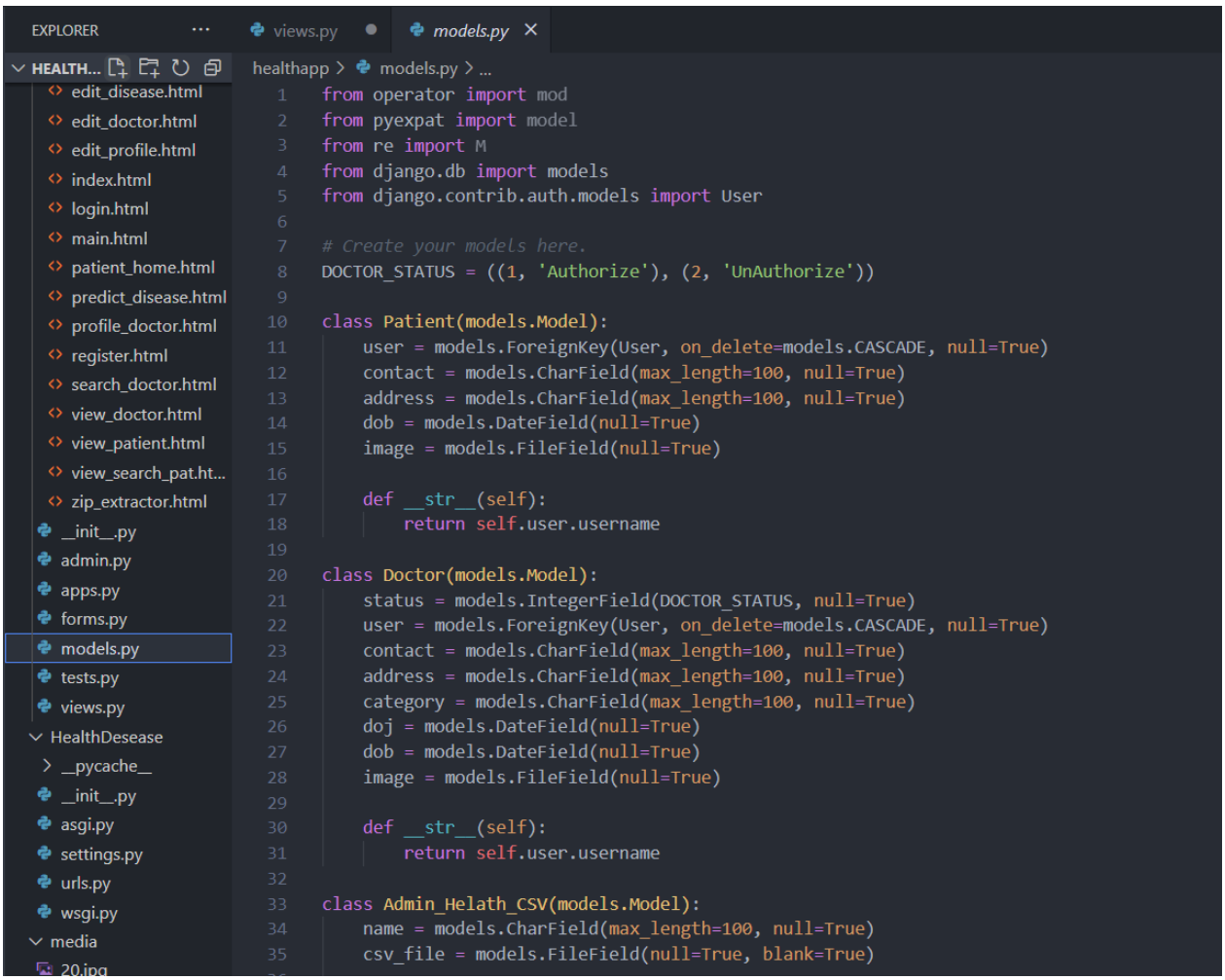

Figure 73 Sample code of the backend

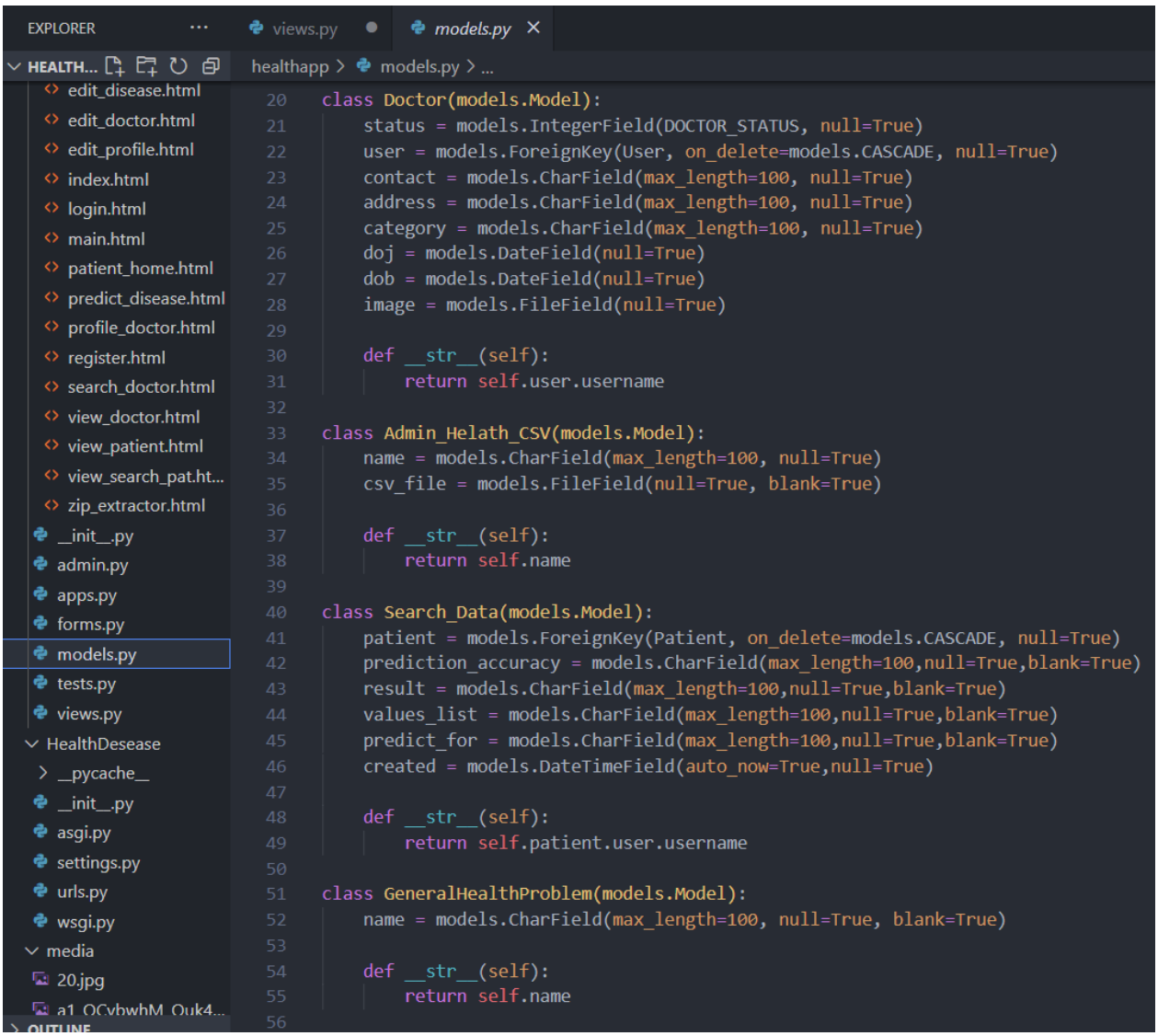

Figure 74 Sample code of the backend

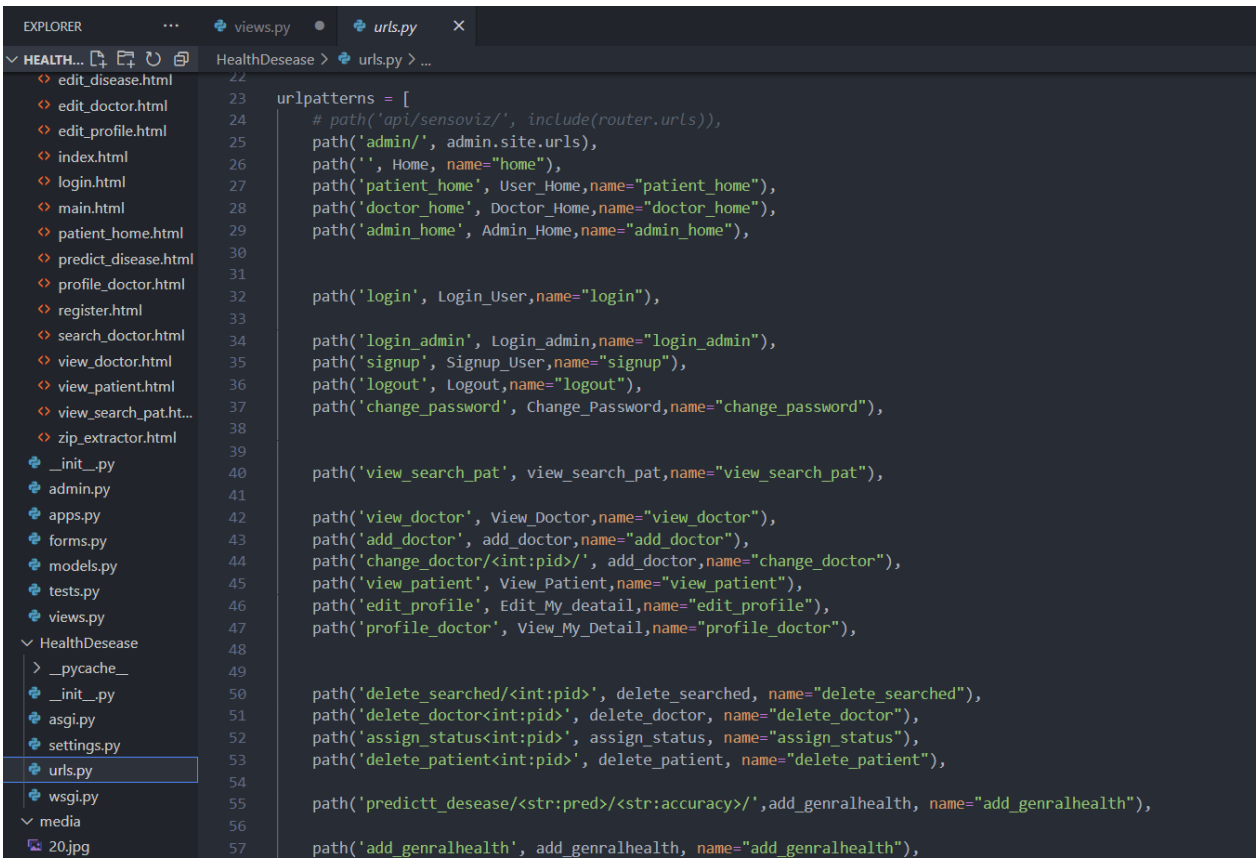

Figure 75 Sample code of the backend

CHAPTER 10: APPENDIX: Screenshots of the system

1. Home page

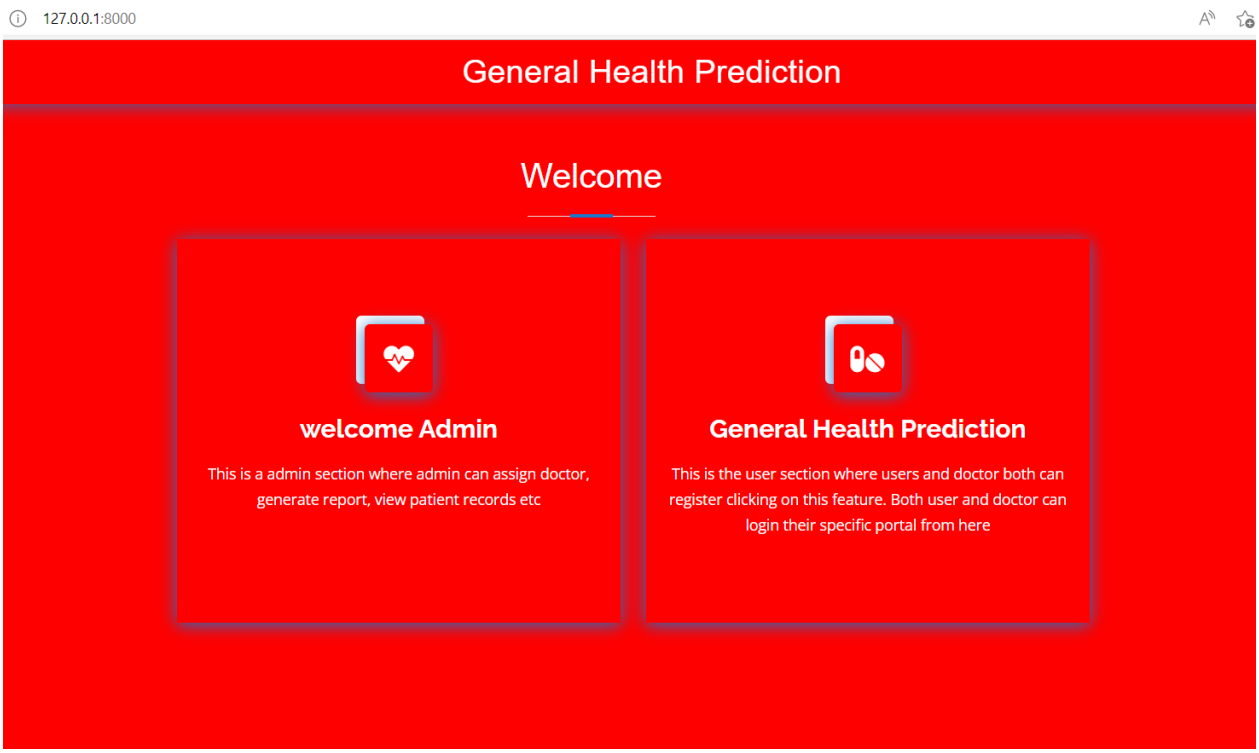

Figure 76 Home page

2. Admin page

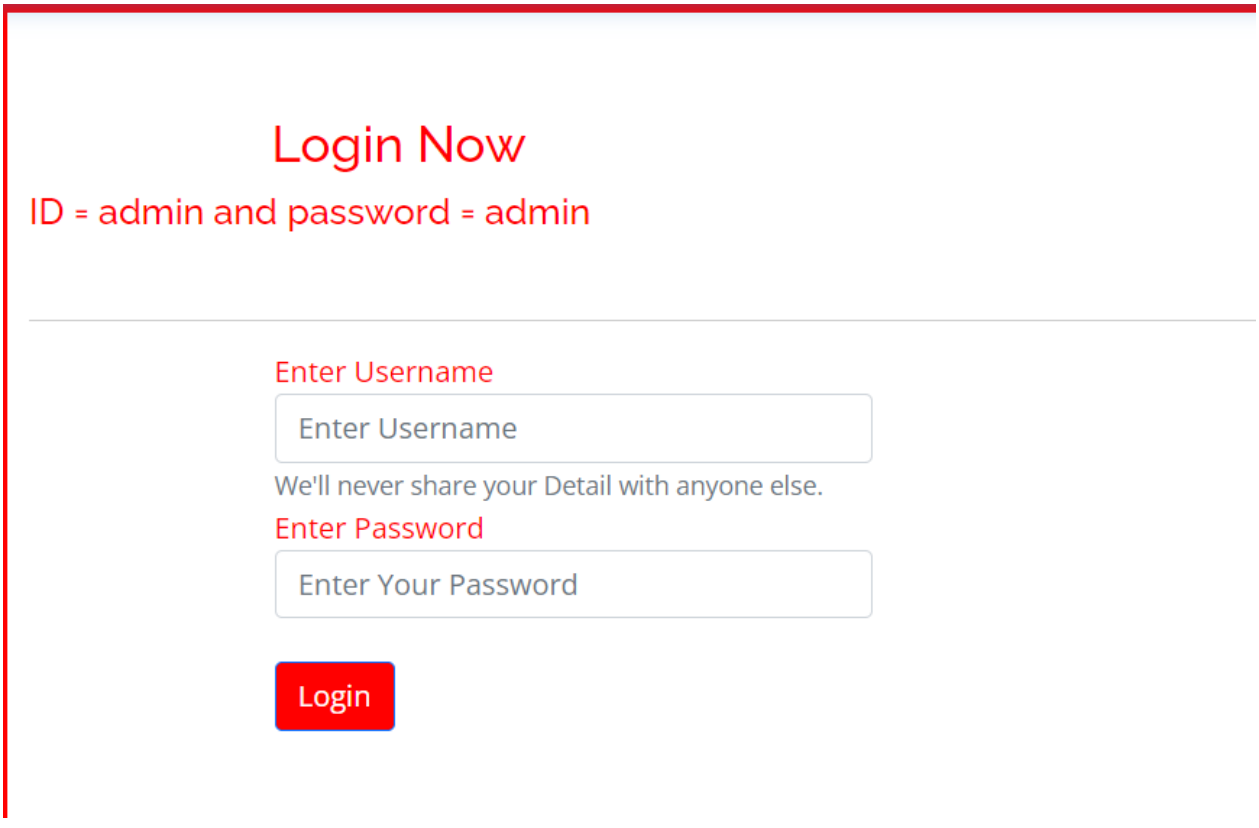

Figure 77 admin login page

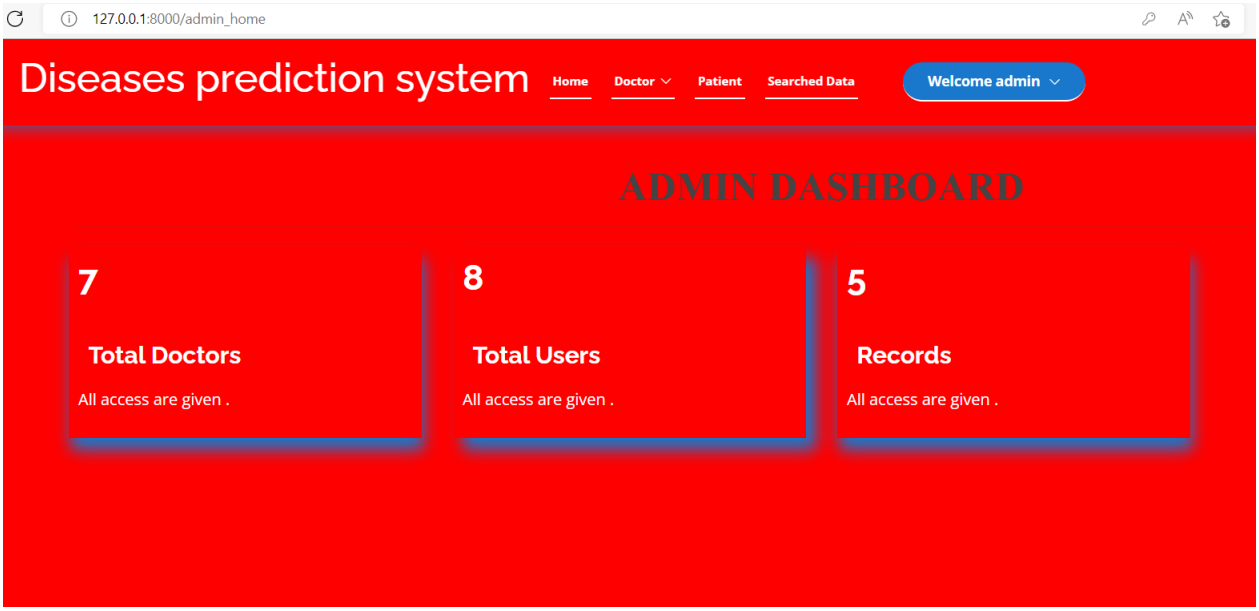

Figure 78 Admin portal page

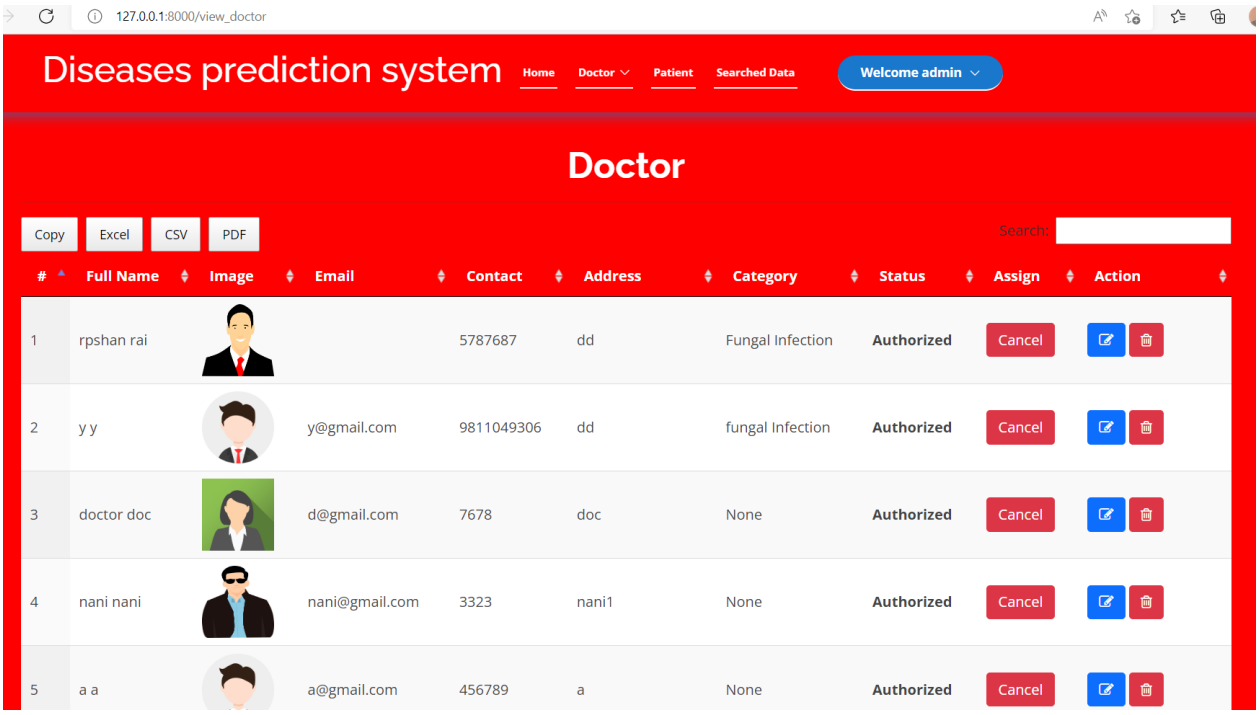

Figure 79 Viewing doctor

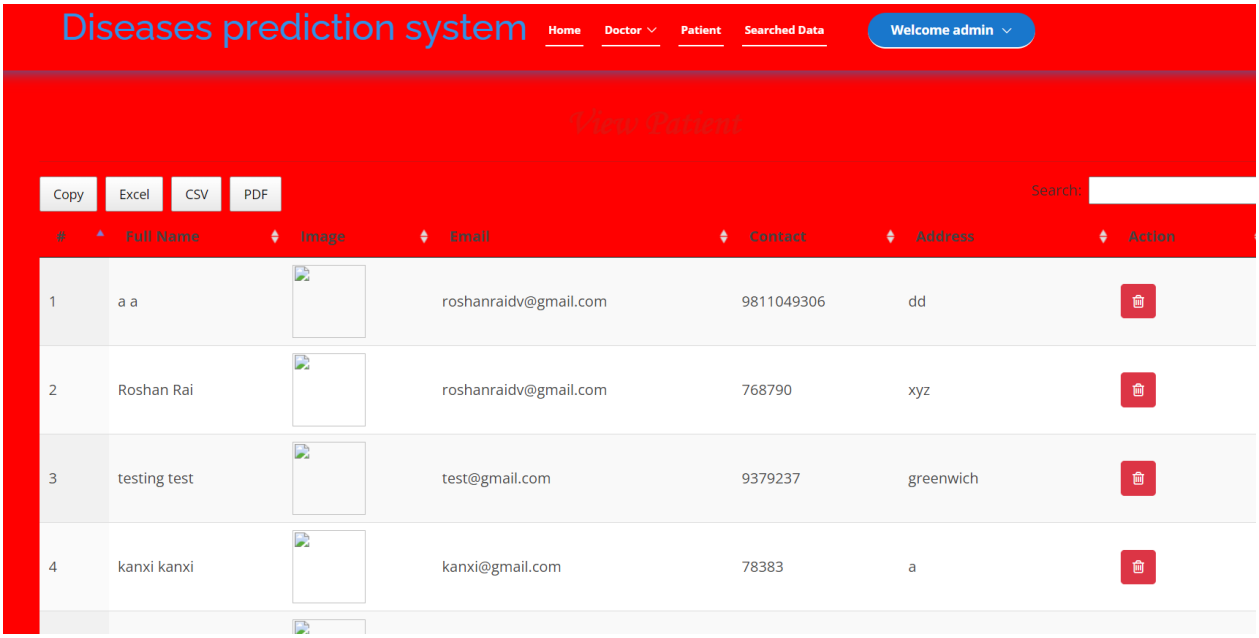

Figure 80 Users detail

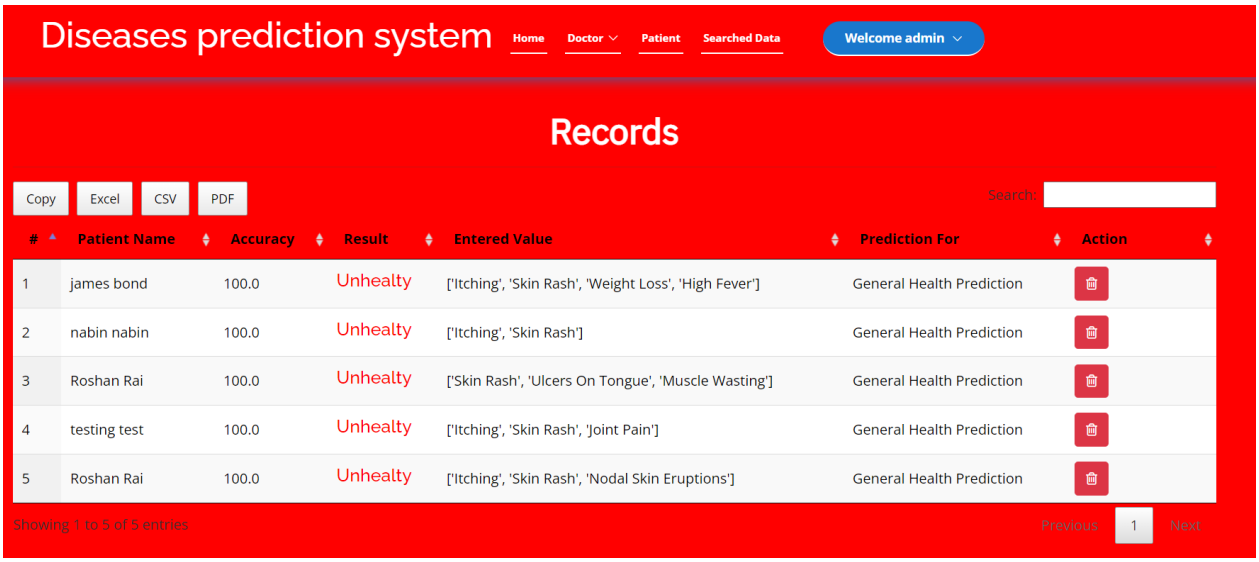

Figure 81 patient record

3. Users

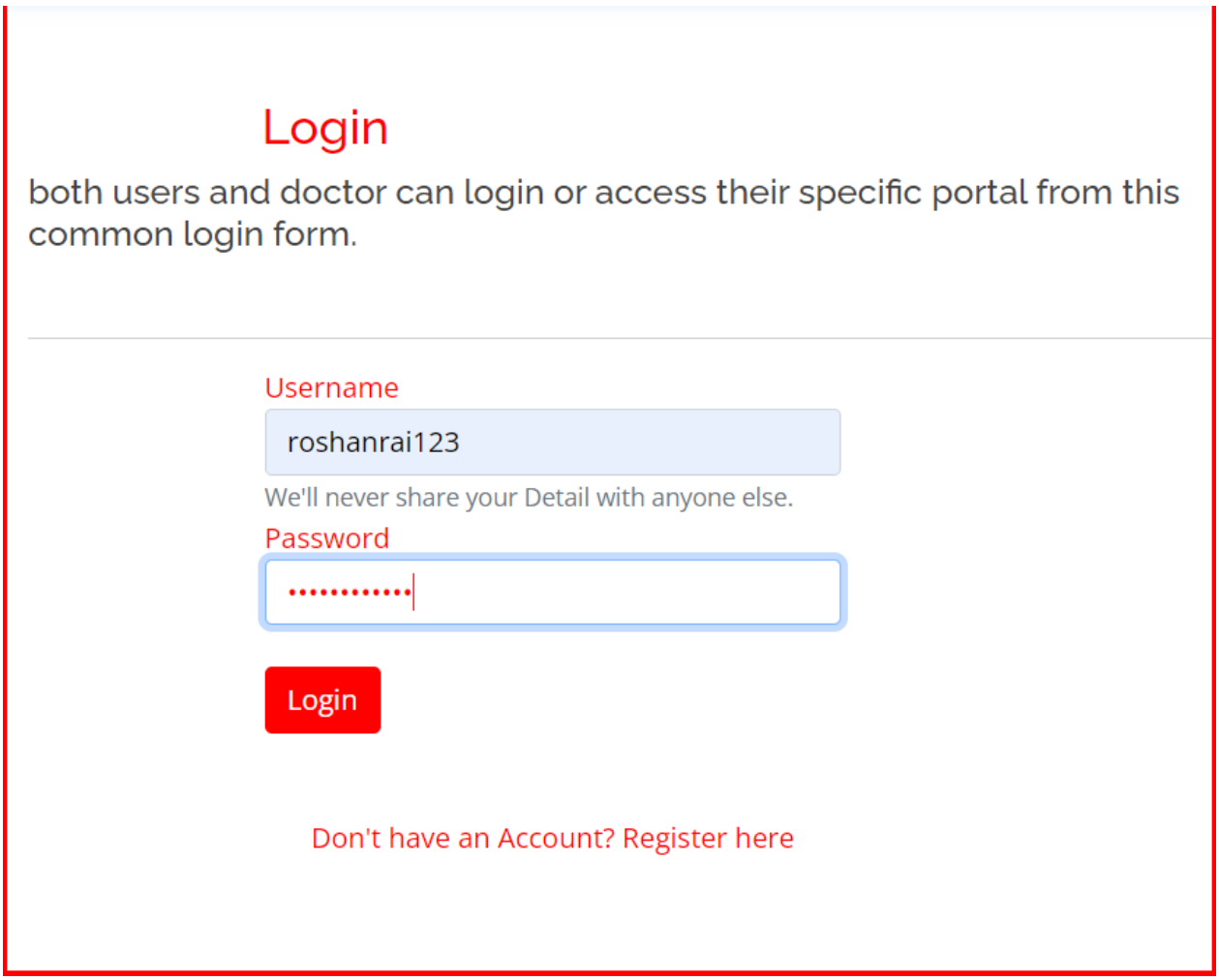

Figure 82 User's login page

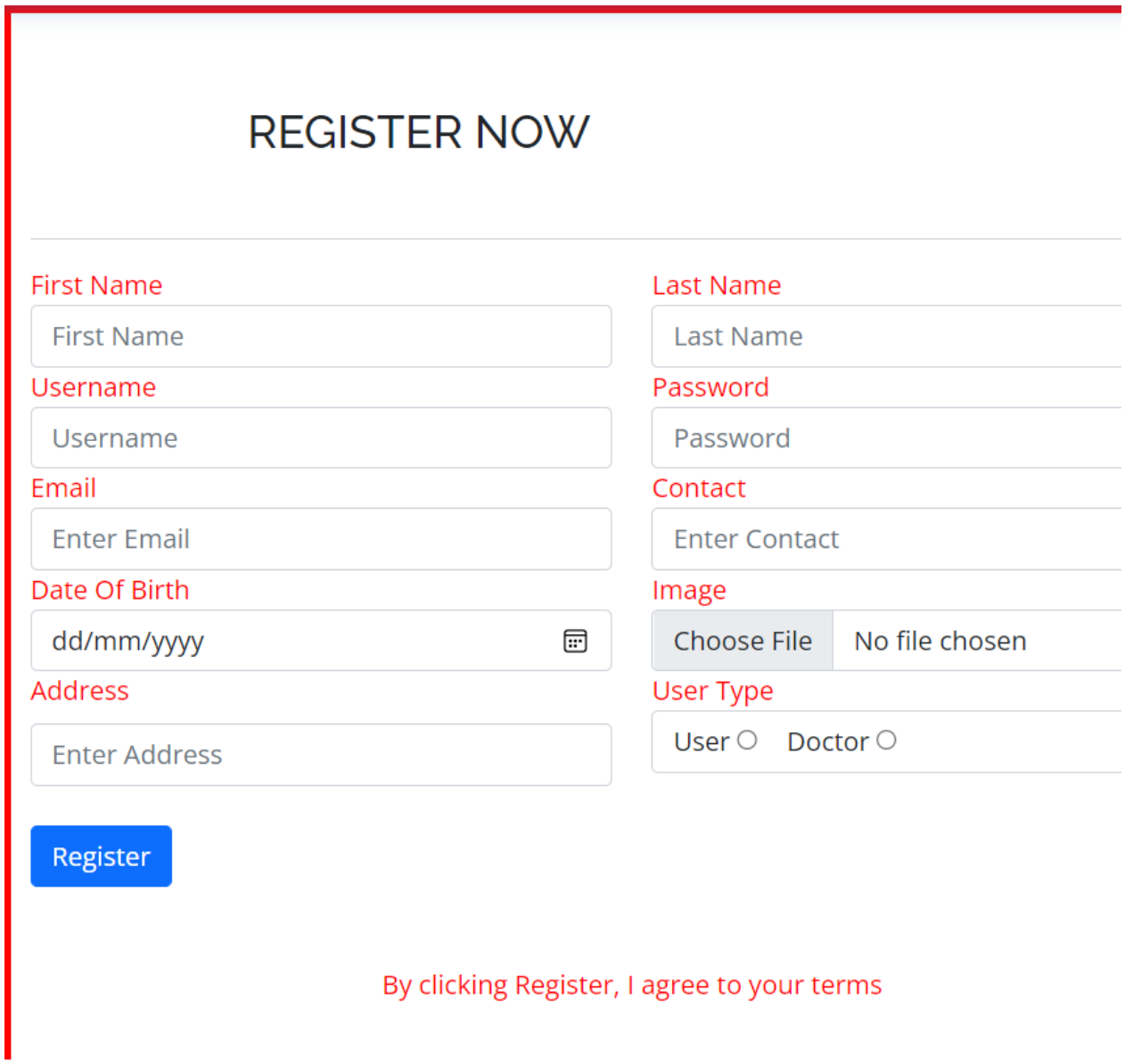

Figure 83 Registration page

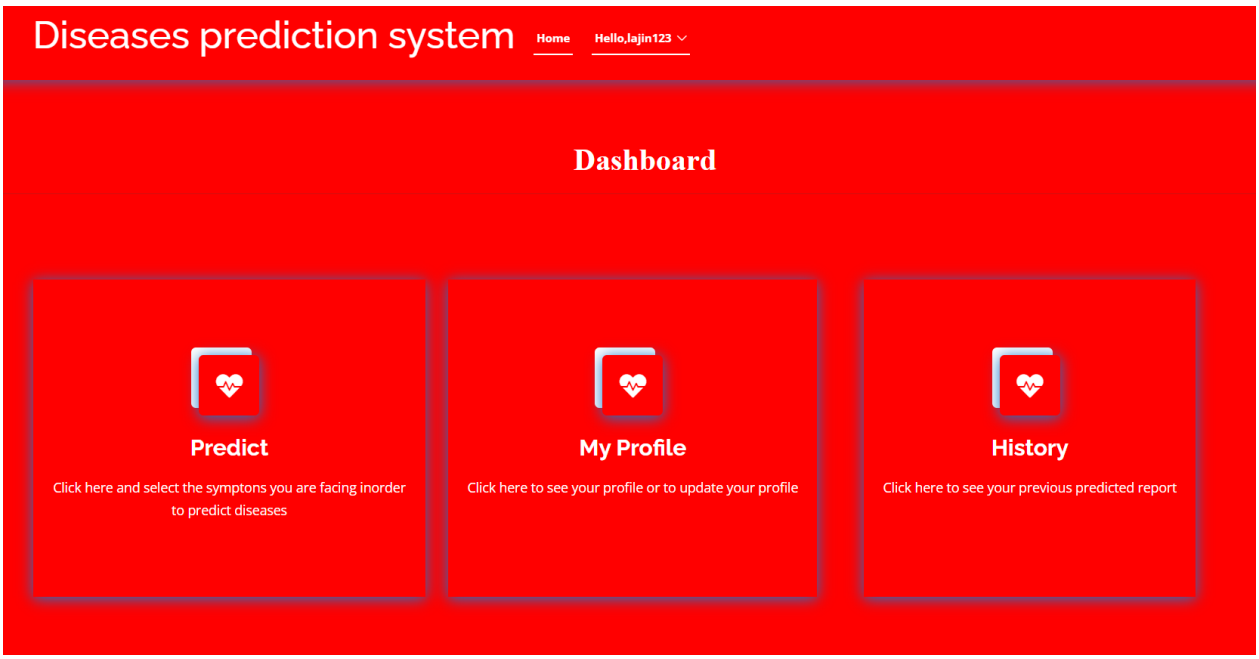

Figure 84 User's home page

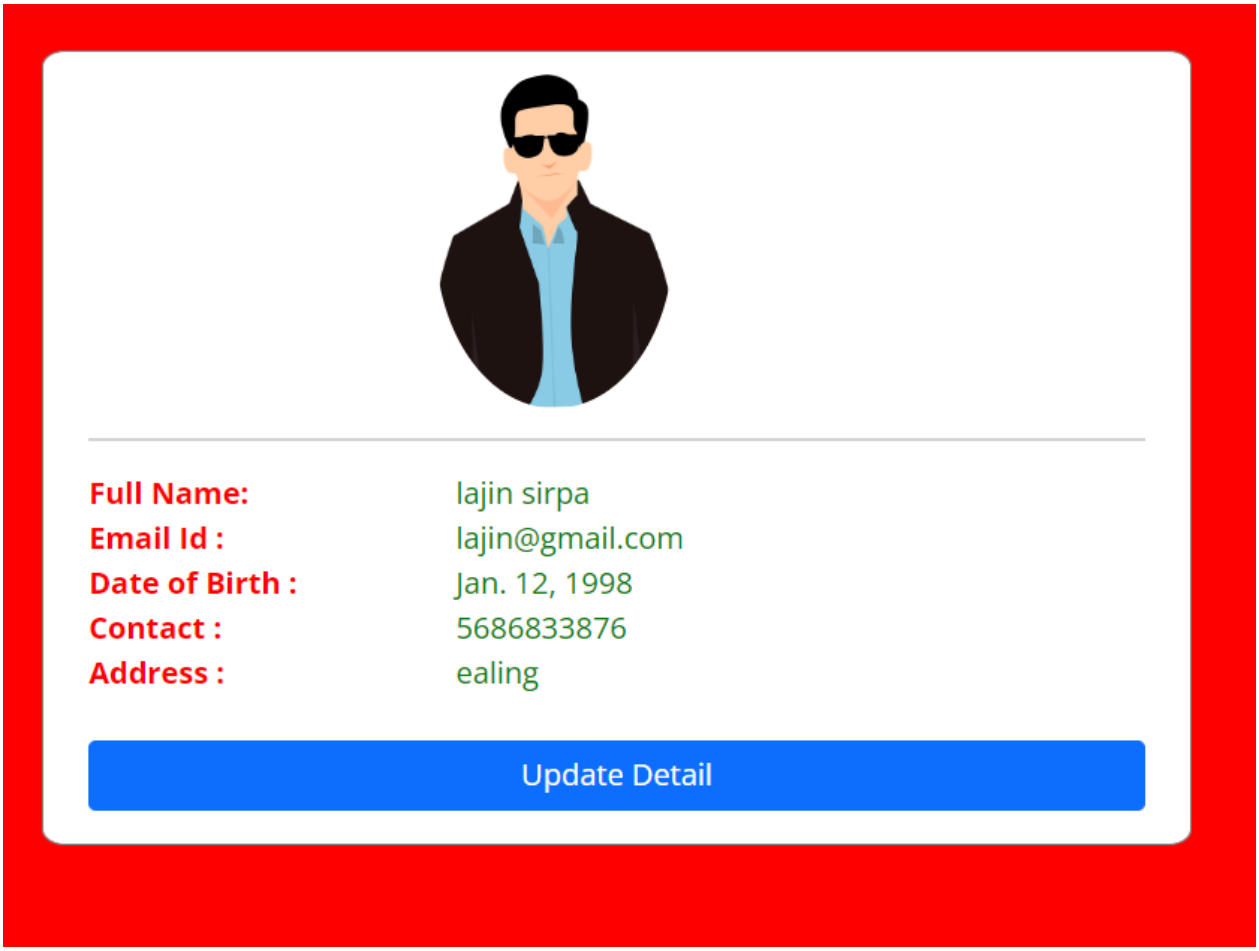

Figure 85 User profile page

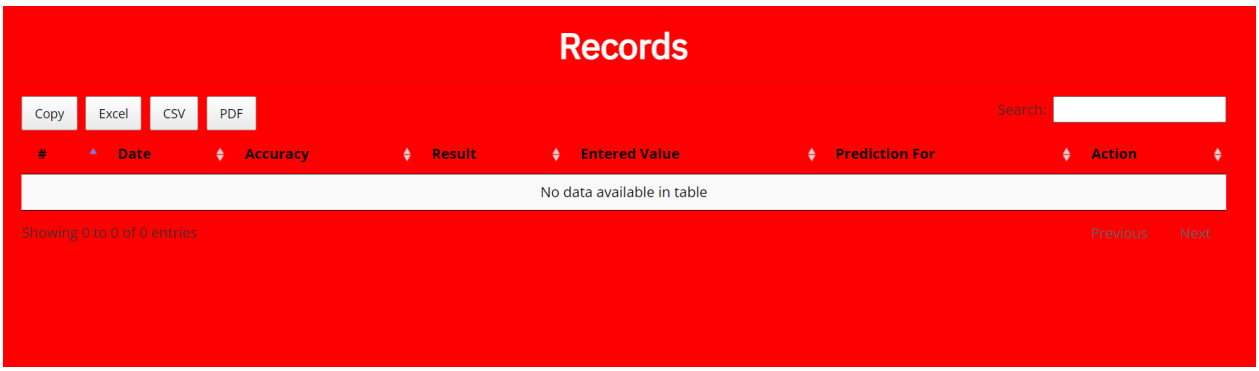

Figure 86 User history page

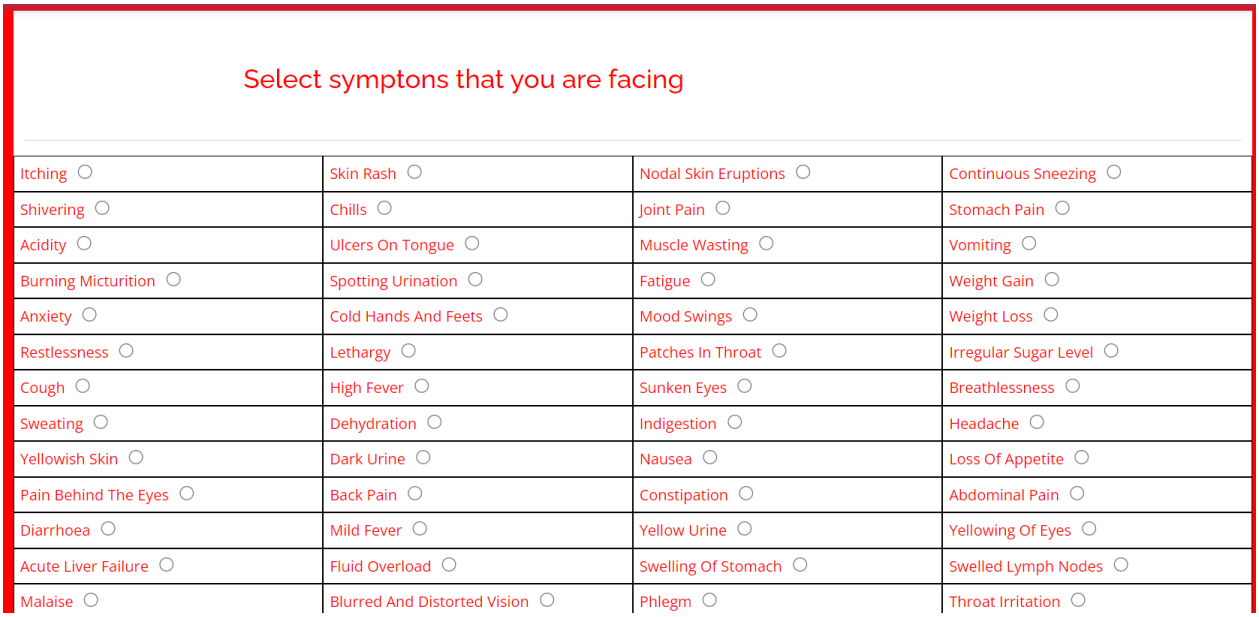

Figure 87 Symptoms listed page

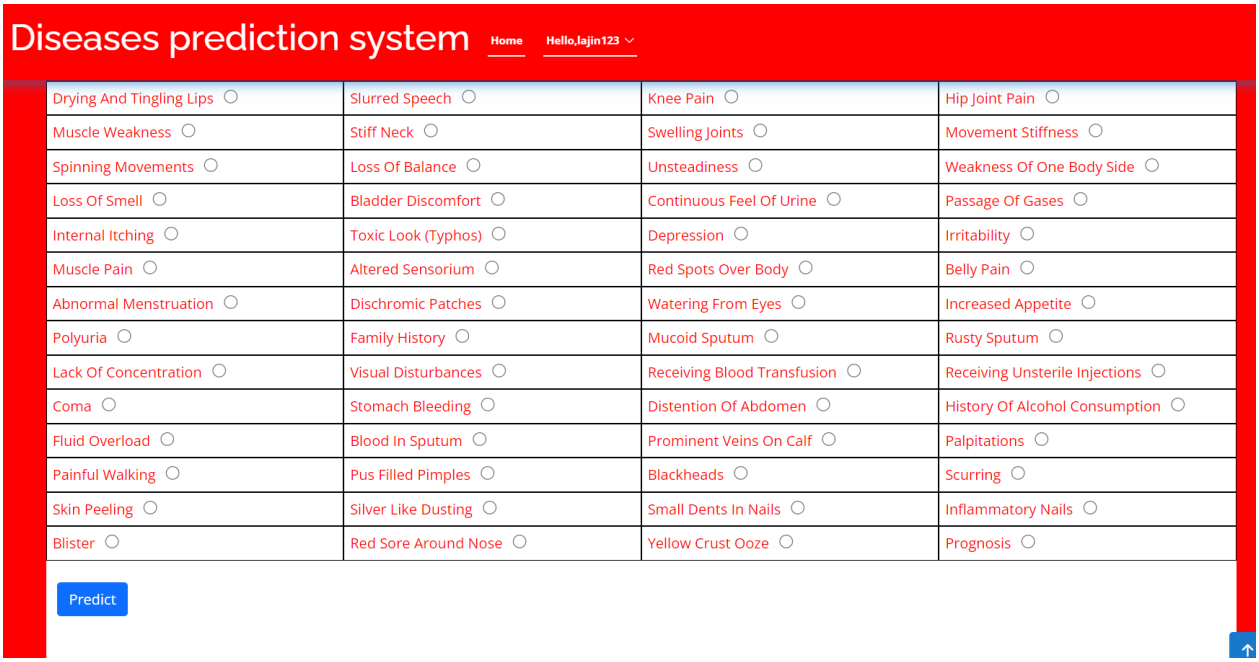

Figure 88 Symptoms listed page

4. Doctor

Figure 89 Doctor home page

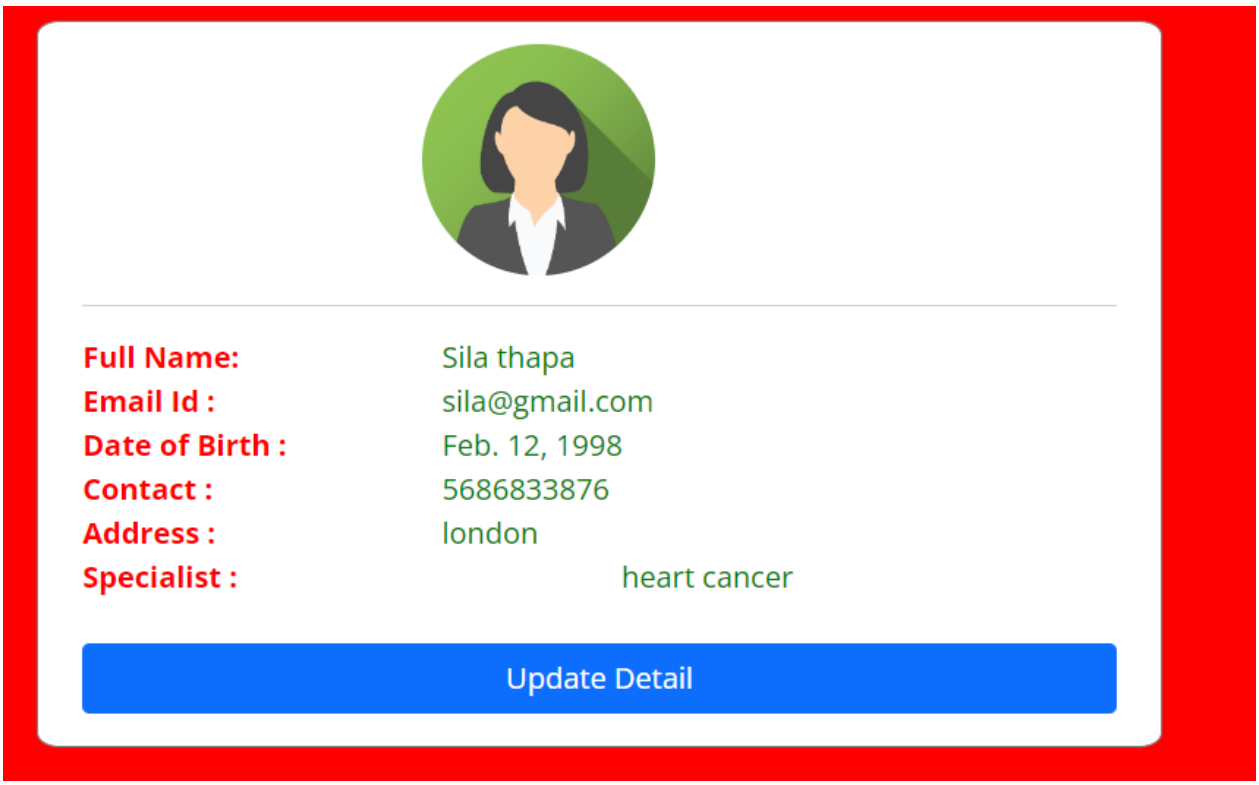

Figure 90 doctor profile page

CHAPTER 11: APPENDIX: Software requirement specification (SRS)

11.1. Overview of the system:

Well, we build a system that can detect diseases from the signs and symptoms users provide. The system's integrated model was created through the use of Random forest and the SVM algorithm. There is a registration process before users may use the login feature. Customers can update their information and look at their predicted medical history from the past. Patient recommendations to the doctor are made using the system's recommendation functions, which take into account the specifics of each ailment. As with patients, doctors can join the system and have access to the doctor portal by registering. They get access to their personal profile and medical history. In addition, the administrator is a crucial component of the system. The administrative staff can monitor the doctor and the patient. The administrator has access to all patient and physician data and may make edits and view the history of the patient and the medical team.

11.2. User types and description

This system is intended for three types of users:

- Patient: Patient can access the system and provide the symptoms they are facing by which system can predict the diseases and recommend nearest doctor for further treatment.
- Doctor: - Doctor can access the system to view the patient report as well as to provide the medical consultation for patient.
- Admin: - Admin access the system for monitoring the records, and reports of the patient and doctor. Also, admin can access the system for adding doctors are making change in the system.

11.3. System features (Functional requirement)

11.3.1. Registration page

This features provide the user with the registration page where they can provide the details and can registered in the system. When user's click on the registration page, framework display the registration page where the system ask the personal details for registration, after which the system validate the details provided by the user's and the login page is displayed.

11.3.2. Login page

This features provide the user with a login page where user can access the system by login with their registered username and password. The detail provided by the users are validate by the database, when their detail are verified in the system, they are redirected to their particular home page.

11.3.3. View and Manage User Profile

This features provided the user to view their details and also they can make change in their details. When user click on their particular profile, a page is displayed where the users can make changes.

11.3.4. Input symptoms

This features allow users to input their symptoms by selecting through the interface. When they select the symptoms, selected symptoms correspond with the symptoms that are trained using the model and as per the prediction made by the model, result is displayed.

11.3.5. View Patient Profile

This features allows admin and doctor to view the patient profile through which they can acknowledge the medical situation of the patient.

11.3.6. Search for the doctor

This features will allow admin as well to patient for search the doctor.

11.4. Non-functional requirement

11.4.1. Safety requirement

The user should be very careful while providing their symptoms because the system relies on this information and could produce inaccurate results if it receives inaccurate symptoms. Users shouldn't rely completely on this method, though, as diseases and symptoms depend on factors like age, environment, and medical history that are not taken into account by it.

11.4.2. Security requirement

The personal health data of user's are secure using the password features in the system. The admin ensured the authenticity of doctor as there might be threat of false claiming doctors.

11.4.3. Software quality attributes

This system is reliable, reusable, availability and usability as it aim to provide a medical service to the number people and only the authentic doctor allow to access the system after they are verified by the admin.

11.5. Technical aspect

11.5.1. Hardware requirement

- Minimum 5^{th} generation computer.
- Storage minimum 100gb
- RAM minimum 4gb

11.5.2. Software requirement

- Python

  For codding.
- Jupyter

  For studying and visualization dataset
- Visual studio code

  For developing code
- Bootstrap

  For designing user interface
- HTML

  For designing user interface
- CSS

  For designing user interface

- Java script

  For writing some logical script
- Anaconda

  For installing dependencies.