CHAPTER 1

INTRODUCTION

1.1 What is Machine Learning?

Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step.

The discipline of machine learning employs various approaches to teach computers to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithms it uses to determine correct answers.

Neural networks, or artificial neural networks (ANNs), are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. The “deep” in deep learning is just referring to the depth of layers in a neural network. A neural network that consists of more than three layers—which would be inclusive of the inputs and the output—can be considered a deep learning algorithm or a deep neural network. A neural network that only has two or three layers is just a basic neural network.

1.2 How Machine Learning works?

Machine learning is a form of artificial intelligence (AI) that teaches computers to think in a similar way to how humans do: learning and improving upon past experiences. It works by exploring data, identifying patterns, and involves minimal human intervention. Almost any task that can be completed with a data-defined pattern or set of rules can be automated with machine learning. Machine learning uses two main techniques:

Supervised learning: It allows you to collect data or produce a data output from a previous ML deployment. Supervised learning is exciting because it works in much the same way humans actually learn. In supervised tasks, we present the computer with a collection of labelled data points called a training set. In supervised learning, models are trained using labelled dataset, where the model learns about each type of data. Once the training process is completed, the model is tested on the basis of test data (a subset of the training set), and then it predicts the output. Supervised techniques adapt the model to reproduce outputs known from a training set (e.g. recognize car types on photos). In the beginning, the system receives input data as well as output data. Its task is to create appropriate rules that map the input to the output. The training process should continue until the level of performance is high enough. After training, the system should be able to assign an output objects which it has not seen during the training phase. In most cases, this process is really fast and accurate.

There are two types of Supervised Learning techniques: Regression and Classification. Classification separates the data Regression fits the data. Regression is a technique that aims to reproduce the output value. We can use it, for example, to predict the price of some product, like a price of a house in a specific city or the value of a stock. There is a huge number of things we can predict if we wish. Classification is a technique that aims to reproduce class assignments. It can predict the response value and the data is separated into “classes”. Examples? Recognition of a type of car in a photo, is this mail spam or a message from a friend, or what the weather will be today.

- Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.
- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.
- Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y)**.**
- In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

Examples:

- **Neural Networks:** Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another. Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are finetuned for accuracy, they are powerful tools in computer science and artificial intelligence, allowing us to classify and cluster data at a high velocity. Tasks in speech recognition or image recognition can take minutes versus hours when compared to the manual identification by human experts. One of the most well-known neural networks is Google’s search algorithm.

- **Decision Trees:** Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split. An example of a decision tree can be explained using above binary tree. Let’s say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like ‘What’s the age?’, ‘Does he exercise?’, ‘Does he eat a lot of pizzas’? And the leaves, which are outcomes like either ‘fit’, or ‘unfit’. In this case this was a binary classification problem (a yes no type problem).

- **Naive Bayes**: It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c).

- **K-Nearest Neighbor:** In statistics, the k-nearest neighbors algorithm (KNN) is a non-parametric classification method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. It is used for classification and regression. In both cases, the input consists of the k closest training examples in data set. The output depends on whether k-NN is used for classification or regression: In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors. k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation. Since this algorithm relies on distance for classification, if the features represent different physical units or come in vastly different scales then normalizing the training data can improve its accuracy dramatically. Both for classification and regression, a useful technique can be to assign weights to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbors. k-NN is a type of classification where the function is only approximated locally and all computation is deferred until function evaluation.

Unsupervised learning: It cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

Example: Suppose the unsupervised learning algorithm is given an input dataset containing images of different types of cats and dogs. The algorithm is never trained upon the given dataset, which means it does not have any idea about the features of the dataset. The task of the unsupervised learning algorithm is to identify the image features on their own. Unsupervised learning algorithm will perform this task by clustering the image dataset into the groups according to similarities between images.

Examples:

- **K-Mean Clustering**: k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids. The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. They both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the Gaussian mixture model allows clusters to have different shapes. The unsupervised k-means algorithm has a loose relationship to the k-nearest neighbor classifier, a popular supervised machine learning technique for classification that is often confused with k-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by k-means classifies new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

- **Principal Component Analysis(PCA):** The principal components of a collection of points in a real coordinate space are a sequence of unit vectors, where the vector is the direction of a line that best fits the data while being orthogonal to the first vectors. Here, a best-fitting line is defined as one that minimizes the average squared distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data. The principal component can be taken as a direction orthogonal to the first principal components that maximizes the variance of the projected data. From either objective, it can be shown that the principal components are eigenvectors of the data's covariance matrix. Thus, the principal components are often computed by eigen decomposition of the data covariance matrix or singular value decomposition of the data matrix. PCA is the simplest of the true eigenvector-based multivariate analyses and is closely related to factor analysis. Factor analysis typically incorporates more domain specific assumptions about the underlying structure and solves eigenvectors of a slightly different matrix. PCA is also related to canonical correlation analysis (CCA). CCA defines coordinate systems that optimally describe the cross-covariance between two datasets while PCA defines a new orthogonal coordinate system that optimally describes variance in a single dataset.

- **Apriori Algorithm:** Apriori is an algorithm for frequent item set mining and association rule learning over relational databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as analysis. The Apriori algorithm was proposed by Agrawal and Srikant in 1994. Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation or IP addresses). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold , the Apriori algorithm identifies the item sets which are subsets of at least transactions in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length from item sets of length . Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent -length item sets.

1.3 Characteristics of Machine Learning:

(i) **The ability to perform automated data visualization**: A massive amount of data is being generated by businesses and common people on a regular basis. By visualizing notable relationships in data, businesses can not only make better decisions but build confidence as well. Machine learning offers a number of tools that provide rich snippets of data which can be applied to both unstructured and structured data. With the help of user-friendly automated data visualization platforms in machine learning, businesses can obtain a wealth of new insights in an effort to increase productivity in their processes.

(ii) **Automation at its best:** One of the biggest characteristics of machine learning is its ability to automate repetitive tasks and thus, increasing productivity. A huge number of organizations are already using machine learning-powered paperwork and email automation

(iii) **Customer engagement like never before:** For any business, one of the most crucial ways to drive engagement, promote brand loyalty and establish long-lasting customer relationships is by triggering meaningful conversations with its target customer base. Machine learning plays a critical role in enabling businesses and brands to spark more valuable conversations in terms of customer engagement. The technology analyzes particular phrases, words, sentences, idioms, and content formats which resonate with certain audience members. You can think of Pinterest which is successfully using machine learning to personalize suggestions to its users. It uses the technology to source content in which users will be interested, based on objects which they have pinned already.

(iv) **The ability to take efficiency to the next level when merged with IoT**: Machine learning has experienced a great rise in popularity. IoT is being designated as a strategically significant area by many companies. And many others have launched pilot projects to gauge the potential of IoT in the context of business operations. But attaining financial benefits through IoT isn’t easy. In order to achieve success, companies, which are offering IoT consulting services and platforms, need to clearly determine the areas that will change with the implementation of IoT strategies. Many of these businesses have failed to address it. In this scenario, machine learning is probably the best technology that can be used to attain higher levels of efficiency. By merging machine learning with IoT, businesses can boost the efficiency of their entire production processes.

(v) **The ability to change the mortgage market**: It’s a fact that fostering a positive credit score usually takes discipline, time, and lots of financial planning for a lot of consumers. When it comes to the lenders, the consumer credit score is one of the biggest measures of creditworthiness that involve a number of factors including payment history, total debt, length of credit history etc. But wouldn't it be great if there is a simplified and better measure? With the help of machine learning, lenders can now obtain a more comprehensive consumer picture. They can now predict whether the customer is a low spender or a high spender and understand his/her tipping point of spending.

(vi) **Accurate data analysis:** Data analysis has always been encompassing trial and error method, an approach which becomes impossible when we are working with large and heterogeneous datasets. Machine learning comes as the best solution to all these issues by offering effective alternatives to analyzing massive volumes of data. By developing efficient and fast algorithms, as well as, data-driven models for processing of data in real-time, machine learning is able to generate accurate analysis and results.

(vii) **Business intelligence at its best:** Machine learning characteristics, when merged with big data analytical work, can generate extreme levels of business intelligence with the help of which several different industries are making strategic initiatives.

1.4 Python programming language

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object- oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in the year 2000, introduced features like list comprehensions and a garbage collection system with reference counting.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported and many of its features support functional programming (including by metaprogramming and metaobjects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy.

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

1.4.1 What can Python technology do?

1) Readable and Maintainable Code: The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

2) Multiple Programming Paradigms: Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

3) Compatible with Major Platforms and Systems: At present, Python is supporting many operating systems. Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, not required to recompile thecode after making any alteration. Can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

4) Robust Standard Library: Its large and robust standard library makes Python score over other programming languages. The standard library allows to choose from a wide range of modules according to user precise needs. Each module further enables to add functionality to the Python application without writing additional code. For instance, while writing a web application in Python, can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group Monty Python—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

5) Many Open-Source Frameworks and Tools: As an open-source programming language, Python helps to curtail software development cost significantly. Can even use several open-source Python frameworks, libraries and development tools to curtail development time without increasing development cost. For instance, can simplify and speedup web application development by using robustPython web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy.

6) Simplify Complex Software Development: Python is a general-purpose programming language. Hence, can use the programming language for developing both desktop and web applications. Also, can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis andvisualization. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

7) Adopt Test Driven Development: Python can be used to create prototype of the software application rapidly. Also, can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier to perform coding and testing simultaneously by adopting test driven development (TDD) approach. The tests can also be used for checking if the application meets predefined requirements based on its sourcecode.

CHAPTER 2

LITERATURE SURVEY

[1] “Prediction of Cardiovascular Disease Using Machine Learning Algorithms”,

Kumar G Dinesh, K Arumugaraj, Kumar D Santhosh, V Mareeswari, June 2017

It contributes the correlative application and analysis of distinct machine learning algorithms in the R software which gives an immediate mechanism for the user to use the machine learning algorithms in R software for forecasting the cardiovascular diseases. The results show that the system has great potential in predicting the heart disease risk level more accurately. ID3 has some features like removing outliers, handling missing values and but their major disadvantage is to over-fitting. And it’s not so easy to implement as that of Naïve Bayes algorithm.

[2] “Multi Disease Prediction Using Data Mining Techniques”,

K. Gomathi Kamaraj, D. Shanmuga Priyaa, February 2019

In this study two different data mining classification techniques was used for the prediction of various diseases and their performance was compared in order to evaluate the best classifier. The Artificial Neural Network, K Means Clustering Algorithm and Frequent Item Set generation using Apriori Techniques are used to classify whether a patient suffers from heart disease or not. Up to now, several studies have been reported that have focused on cardiovascular disease diagnosis. These dies have applied different approaches to the given problem and achieved high classification vacancies of 77% or higher.

[3] “Prediction of Heart Disease Using Machine Learning Algorithms”

Santhana Krishnan J, Geetha S, April 2019

Here two supervised data mining algorithms was applied on the dataset to predict the possibilities of having heart disease of a patient, were analyzed with classification model namely Naïve Bayes Classifier and Decision tree classification. The Decision tree model has predicted the heart disease patient with an accuracy level of 91% and Naïve Bayes classifier has predicted heart disease patient with an accuracy level of 87%.

## [4] "A Proposed Model for Lifestyle Disease Predict Vectorion Using Support Machine"

Rajeev D M, Shadab Arshad, January 2019

This study aims to understand support vector machine and use it to predict lifestyle diseases that an individual might be susceptible to. In the existing system the info set is often little, for patients and diseases with specific conditions. These systems are principally designed for the additional prodigious diseases like cardiovascular disease, Cancer etc. The pre-selected characteristics could generally not satisfy the changes within the malady and its influencing factors that may lead to quality in results. The main feature will be the machine learning, in which algorithm use are such as Naïve Bayes Algorithm, K-Nearest Algorithm, Decision Tree Algorithm, Random Forest Algorithm and Support Vector Machine, which will help us in getting accurate predictions.

## [5] "Review of Medical Disease Symptoms Prediction Using Data Mining Technique"

Rahul Deo Sah, Dr Jitendra Sheetalani, September 2018

It evaluates the performance of medical disease prediction based on data mining technique. Classification proceeds based on classifier selection to medical disease data and propose a clustering-based classifier selection method. In the method, many clusters are selected for a ensemble process. Then, the standard presentation of each classifier on selected clusters is calculated and the classifier with the best average performance is chosen to classify the given data. Data Mining Technique: KNN Classifier, Rough Set Theory, SVM (Support Vector Machine). ID3 has some features like removing outliers, handling missing values and but their major disadvantage is to over-fitting. And it's not so easy to implement as that of Naïve Bayes algorithm.

[6] “A. L. Predicting Individual Disease Risk Based On Medical History ”

A.Davis, D, V.Chawla, Blumm, Christakis, & Barbasi (2008)

Darcy A. Davis, Nitesh V. Chawla, Nicholas Blumm, Nicholas Christakis, Albert-Laszlo Barabasi have found that global treatment of chronic disease is neither time or cost efficient. So the authors conducted this research to predict future disease risk. For this CARE was used (which relies only on a patient’s medical history using ICD- 9-CM codes in order to predict future diseases risks). CARE combines collaborative filtering methods with clustering to predict each patient’s greatest disease risks based on their own medical history and that of similar patients.

[7] “Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction.”

Jyoti Soni, Ansari,U Sharma & Soni S March 2011

Jyoti Soni, Ujma Ansari, Dipesh Sharma and Sunita Soni have done this research paper into provide a survey of current techniques of knowledge discovery in databases using data mining techniques that are in use in today’s medical research particularly in Heart Disease Prediction. Number of experiment has been conducted to compare the performance of predictive data mining technique on the same dataset and the outcome reveals that Decision Tree outperforms and some time Bayesian classification is having similar accuracy as of decision tree but other predictive methods like KNN, Neural Networks, Classification based on clustering is not performing well.

[8] “Disease Predicting System Using Data Mining”

Nishar Banu MA Gomathy (2013)

M.A. Nishara Banu, B Gomathy used medical data mining techniques like association rule mining, classification, clustering I to analyze the different kinds of heart-based problems. Decision tree is made to illustrate every possible outcome of a decision. Different rules are made to get the best outcome. In this research age, sex, smoking, overweight, alcohol intake, blood sugar, hear rate, blood pressure are the parameters used for making the decisions. Risk level for different parameter.

2.1 EXISTING SYSTEM

Existing system reviews concepts of Machine learning (ML), ML prediction model and applications, describes implementation of ML applications in health care.

The existing system proposes various features to be implemented in health care with various benefits. At first Machine learning algorithms are used to analyze medical data sets effectively in present. The main aim was to implement supervised machine learning concept by using datasets regarding blood cells collected from blood cells detecting and counting sensors, of a human as the input. This input was trained by artificial neural network algorithm and decision tree classification learning algorithm to perform classification It resulted in recognizing and also possibly predict the disease based on the nature of blood cells and classify accordingly. Later The application of machine learning in the field of medical diagnosis is increasing gradually. This contributed primarily to the improvement in the classification and recognition systems used in disease diagnosis. Different classification algorithms are applied, on three separate databases of disease (Heart, Breast cancer, Diabetes) available in UCI repository for disease prediction. The feature selection for each dataset was accomplished by backward modeling using the p-value test. ML is playing a vital role in health care.

DISADVANTAGES OF EXISTING SYSTEM:

- The sub diseases fall under the same category. So, it becomes difficult to differentiate between diseases of the same category but of different types.
- The data set includes the data of mostly the frequently occurring diseases so it leaves out rare diseases or less occurring diseases.
- No sorts of precautions available to patients in case they cannot immediately pay a visit to the doctors.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROPOSED SYSTEM

The proposed system focuses on overcoming all the disadvantages of existing system. Proposed system predicts all diseases and sub diseases which occur in society. The system combines the results using multi classifier into subsets and increase the accuracy. Collecting data from user using disease prediction model, where the user selects the symptoms which the user is experiencing. Dataset is obtained from Kaggle.com and various health related websites. Dataset consists of about 5000 rows, with consisting of 140 symptoms and around 50 diseases which can be predicted. We use different algorithms which increase the overall operational efficiency which include: From an open-source dataset, an excel sheet was created where we listed down all the symptoms for the respective diseases. After which depending on the diseases, age and gender were specified as a part of the dataset. We listed down around 230 diseases with more than 1000 unique symptoms in all. The symptoms, age, and gender of an individual were used as input to various machine learning algorithms.

Decision Tree: generate pattern of disease and define disease and sub disease. The scheming efficiency of our system is quicker than that of the existing systems. when the user symptoms are received, the symptoms are given to the trained model which consists of decision tree and Naïve Bayes algorithm in order to predict the output i.e. the predicted disease. After successful completion of machine learning algorithm, the disease predicted is displayed on the user screen with the probability of its occurrence. The system also recommends the specialist doctors of the disease and platform where patient can text to doctor and take appointments.

3.1.1 Prediction System

Prediction system predicts the disease for the user based on the given symptoms. This is the main processing system where the user’s data is compared with the trained data and various algorithms like decision tree, Naïve Bayes are applied in order to predict the diseases of the patient is based on the symptoms given by the user. This prediction system involves many various algorithms like:

Decision Tree: Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

Decision tree uses tree structure and the tree begins with a single node representing the training samples. If the symptoms are all of the same disease, then node becomes the leaf and the class marks it. Otherwise, the algorithm chooses the discriminatory attribute as the current node of the decision tree. According to the value of the current decision node attribute, the training samples are divided into several subsets, each of which forms a branch, and there are several values that form several branches. For each subset or branch obtained in the previous step, the previous steps are repeated, recursively forming a decision tree on each of the partitioned samples.

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

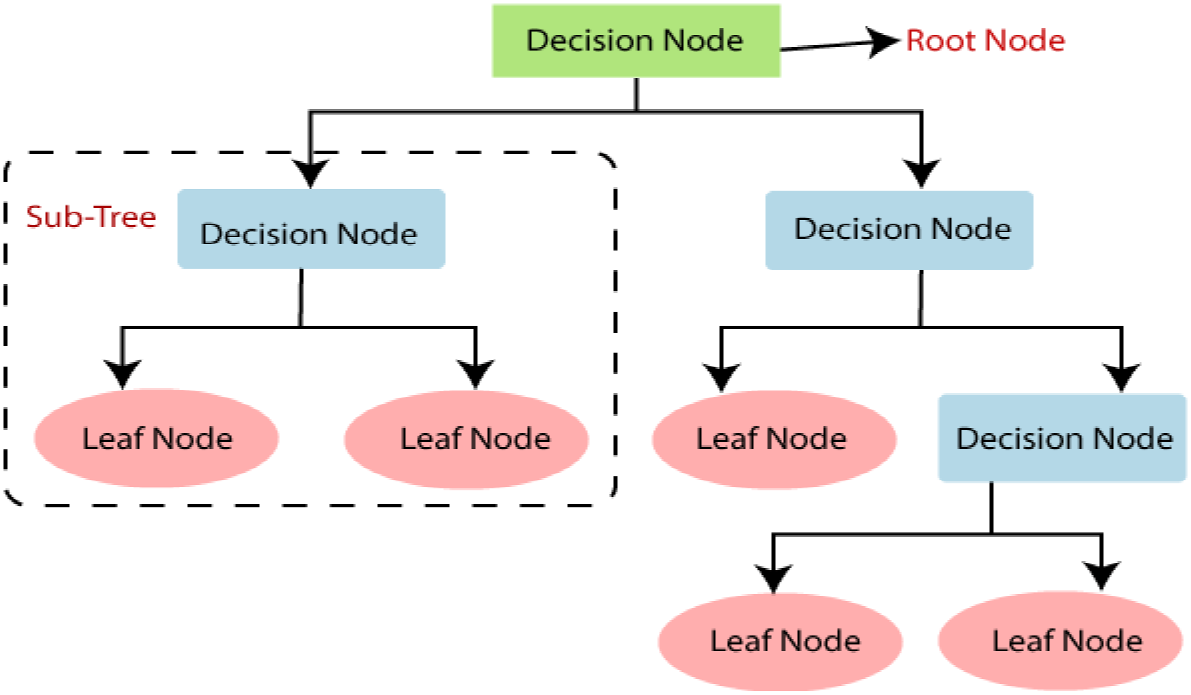
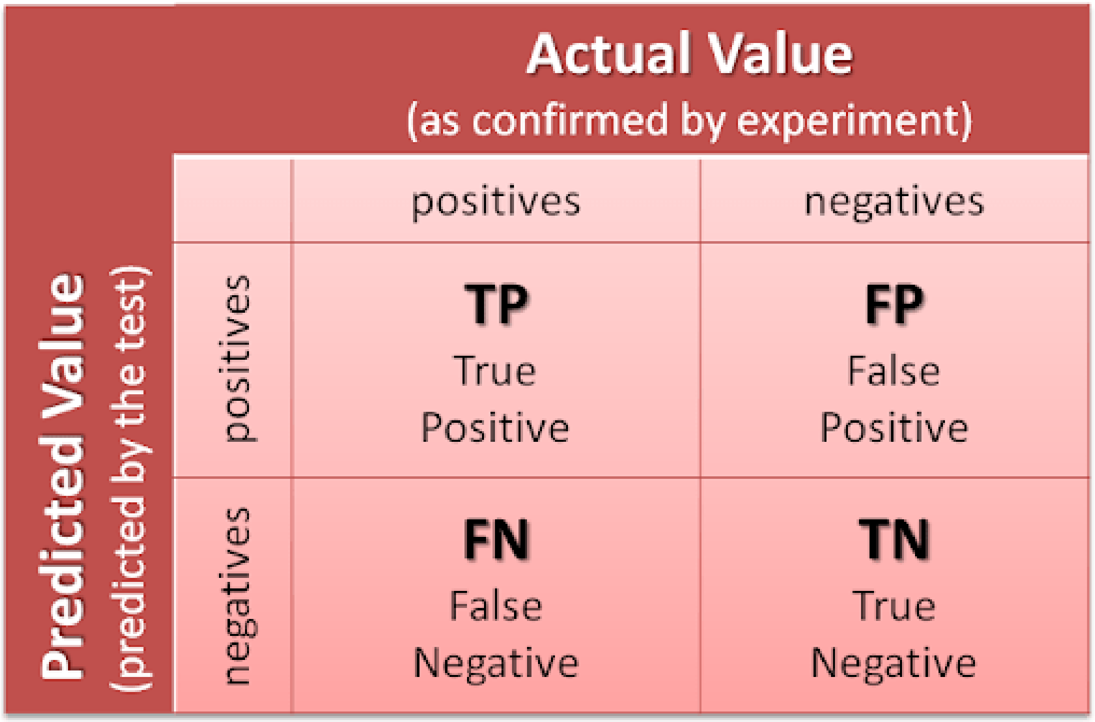


FIG 3.1: DECISION TREE

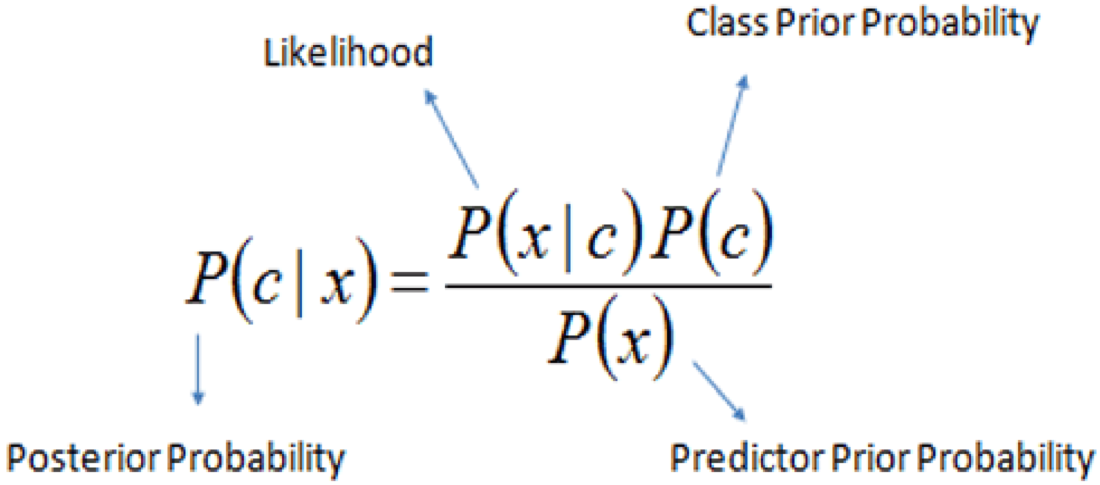


		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

FIG 3.2: CONFUSION MATRIX

Naïve Bayesian: Naïve It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of colour, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other. Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem. Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability. Naive Bayes is a machine learning model that is used for large volumes of data, even if you are working with data that has millions of data records the recommended approach is Naive Bayes. It gives very good results when it comes to NLP tasks such as sentimental analysis. It is a fast and uncomplicated classification algorithm.

Naïve Bayes classifiers are a collection of algorithms based on Bayes' theorem. The dataset is divided into two parts namely Feature matrix and Response vector. Feature matrix contains all vectors(rows) of the dataset in which each vector consists of value of dependent features. In Our data set, features are the symptoms such as 'Fever', 'Cold', 'Headache' etc. Sponse vector contains the values of Class variable (prediction or the output) for each row of feature matrix in our dataset, the class variables are the diseases like ‘malaria’, ‘dengue’ etc. Bayes’ theorem will also find the probability of an event occurring and hence in our model it not only helps in predicting the disease but also gives the probability of the occurrence of the disease.



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Equ 3.1: Bayes Theorem

3.2 PyCharm

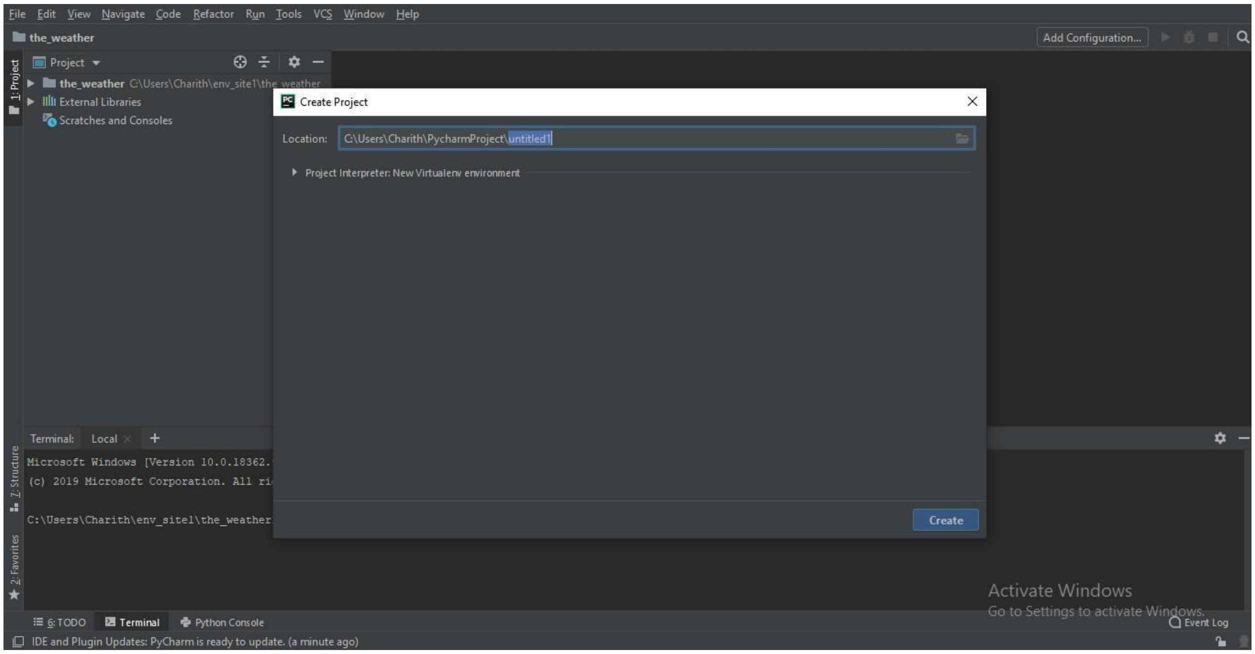
PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCS), and supports web development with Django as well as Data Science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

Some of the features include:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

3.2.1 How to create a new project

1. Select the file from the menu bar.
2. Select New Project.
3. Give the File Name.
4. Press create button. A project is created.



Location: C:\Users\Charith\PycharmProject\untitled1

Project Interpreter: New Virtualenv environment

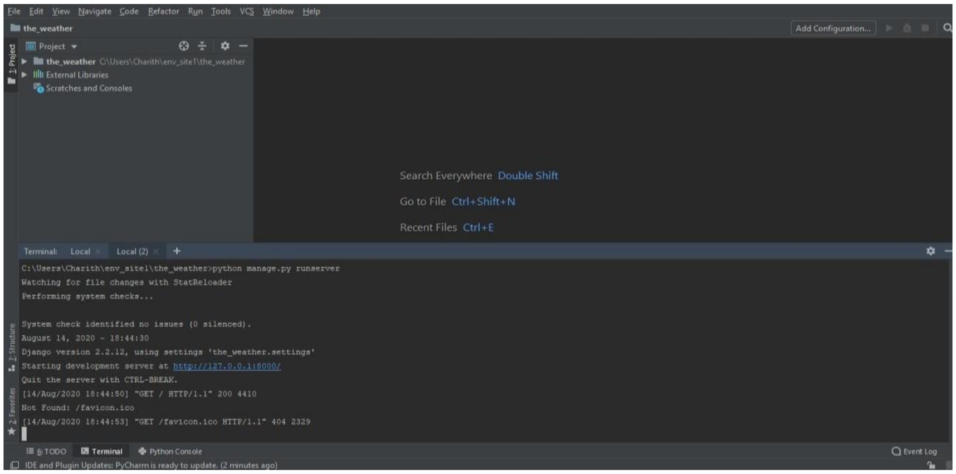
```
Terminal:  Local
Microsoft Windows [Version 10.0.18362.
(c) 2019 Microsoft Corporation. All ri

C:\Users\Charith\env_site1\the_weather
```

IDE and Plugin Updates: PyCharm is ready to update. (a minute ago)

FIG 3.3: PROJECT CREATION

3.2.2 Running the code

1. open the terminal.
2. run the command "python manage.py runserver".
3. An url will be displayed on clicking it opens the web page of the project.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
the_weather
Add Configuration...
Project
the_weather C:\Users\Charith\env_site1\the_weather
External Libraries
Scratches and Consoles
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Terminal: Local Local (2)
6: TODO
Terminal
Python Console
Event Log

```
C:\Users\Charith\env_site1\the_weather>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 14, 2020 - 18:44:30
Django version 2.2.12, using settings 'the_weather.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[14/Aug/2020 18:44:50] "GET / HTTP/1.1" 200 4410
Not Found: /favicon.ico
[14/Aug/2020 18:44:53] "GET /favicon.ico HTTP/1.1" 404 2329
```

IDE and Plugin Updates: PyCharm is ready to update. (2 minutes ago)

Fig 3.4: RUNNING THE PROGRAM

3.3 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django is a Python-based free and open-source web framework that follows the model-view-controller (MVC) architectural pattern. It is maintained by the Django Software Foundation (DSF), an American independent organization established as a non-profit.

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Django's configuration system allows third party code to be plugged into a regular project, provided that it follows the reusable app conventions. More than 2500 packages are available to extend the framework's original behavior, providing solutions to issues the original tool didn't tackle: registration, search, API provision and consumption, CMS, etc.

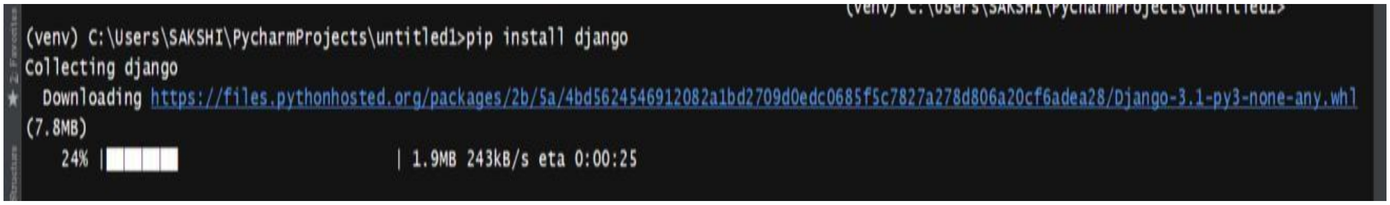
This extensibility is, however, mitigated by internal components' dependencies. While the Django philosophy implies loose coupling, the template filters and tags assume one engine implementation, and both the auth and admin bundled applications require the use of the internal ORM. None of these filters or bundled apps are mandatory to run a Django project, but reusable apps tend to depend on them, encouraging developers to keep using the official stack in order to benefit fully from the app's ecosystem.

Django can be run in conjunction with Apache, Nginx using WSGI, Gunicorn, or Cherokee using flup (a Python module). Django also includes the ability to launch a FastCGI server, enabling use behind any web server which supports FastCGI, such as Lighttpd or Hiawatha. It is also possible to use other WSGI-compliant web servers. Django officially supports five database backends: PostgreSQL, MySQL, MariaDB, SQLite, and Oracle. Microsoft SQL Server can be used with django-mssql on Microsoft operating systems, while similarly external backends exist for IBM Db2,SQL Anywhere and Firebird. There is a fork named django-nonrel, which supports NoSQL databases, such as MongoDB and Google App Engine's Datastore. Django may also be run in conjunction with Jython on any Java EE application server such as GlassFish or JBoss. In this case django-jython must be installed in order to provide JDBC drivers for database connectivity, which also can provide functionality to compile Django in to a .war suitable for deployment. For developing a Django project, no special tools are necessary, since the source code can be edited with any conventional text editor. Nevertheless, editors specialized on computer programming can help increase the productivity of development, e.g., with features such as syntax highlighting. Since Django is written in Python, text editors which are aware of Python syntax are beneficial in this regard.

Integrated development environments (IDE) add further functionality, such as debugging, refactoring, and unit testing. As with plain editors, IDEs with support for Python can be beneficial. Some IDEs that are specialized on Python additionally have integrated support for Django projects, so that using such an IDE when developing a Django project can help further increase productivity. For comparison of such Python IDEs, see the main article: None of these filters or bundled apps are mandatory to run a Django project, but reusable apps tend to depend on them, encouraging developers to keep using the official stack in order to benefit fully from the app’s ecosystem.

3.3.1 How to install Django

1. open pycharm and run the terminal
2. type the command “pip install django”

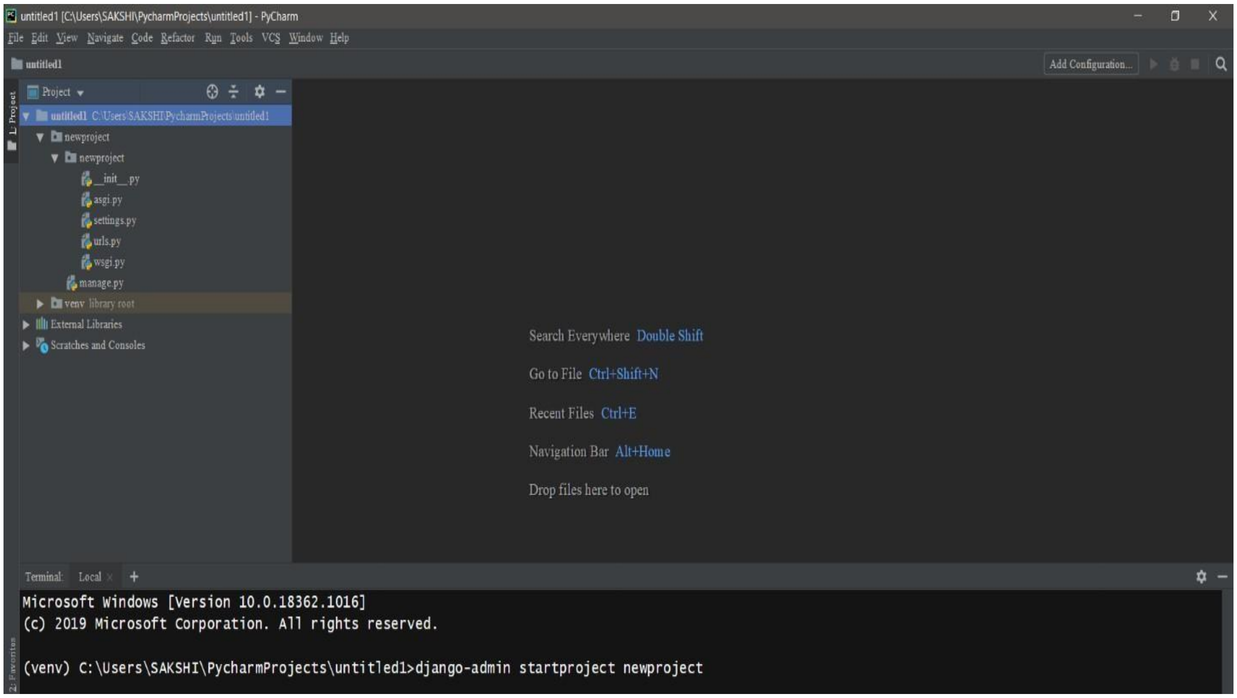


```
(venv) C:\Users\SAKSHI\PycharmProjects\untitled1>
(venv) C:\Users\SAKSHI\PycharmProjects\untitled1>pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/2b/5a/4bd5624546912082a1bd2709d0edc0685f5c7827a278d806a20cf6adea28/Django-3.1-py3-none-any.whl
(7.8MB)
    24% |█████                          | 1.9MB 243kB/s eta 0:00:25
```

FIG 3.5: INSTALLATION OF DJANGO

3.3.2 Setting up the project

1. Once Django is installed open the terminal
2. Type the command Django- admin startproject Project_name
3. Django by default creates some files like settings.py views.py urls.py and anage.py model.py Which we will use to create our project.



```
Microsoft Windows [Version 10.0.18362.1016]
(c) 2019 Microsoft Corporation. All rights reserved.

(venv) C:\Users\SAKSHI\PycharmProjects\untitled1>django-admin startproject newproject
```

FIG 3.6: SETTING UP PROJECT

3.4 PostgreSQL

PostgreSQL also known as Postgres, is a free and open-source relational database management system (RDBMS) emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley. In 1996, the project was renamed to PostgreSQL to reflect its support for SQL. After a review in 2007, the development team decided to keep the name PostgreSQL and the alias Postgres.

PostgreSQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users. It is the default database for macOS Server, and is also available for Linux, FreeBSD, OpenBSD, and Windows.

PostgreSQL manages concurrency through multi version concurrency control (MVCC), which gives each transaction a "snapshot" of the database, allowing changes to be made without affecting other transactions. This largely eliminates the need for read locks, and ensures the database maintains ACID principles. PostgreSQL offers three levels of transaction isolation: Read Committed, Repeatable Read and Serializable. Because PostgreSQL is immune to dirty reads, requesting a Read Uncommitted transaction isolation level provides read committed instead. PostgreSQL supports full serializability via the serializable snapshot isolation (SSI) method.

PostgreSQL evolved from the Ingres project at the University of California, Berkeley. In 1982, the leader of the Ingres team, Michael Stonebraker, left Berkeley to make a proprietary version of Ingres. He returned to Berkeley in 1985, and began a post-Ingres project to address the problems with contemporary database systems that had become increasingly clear during the early 1980s. He won the Turing Award in 2014 for these and other projects,[21] and techniques pioneered in them. Starting in 1986, published papers described the basis of the system, and a prototype version was shown at the 1988 ACM SIGMOD Conference. The team released version 1 to a small number of users in June 1989, followed by version 2 with a re-written rules system in June 1990. Version 3, released in 1991, again re-wrote the rules system, and added support for multiple storage managers and an improved query engine. By 1993, the number of users began to overwhelm the project with requests for support and features

The new project, POSTGRES, aimed to add the fewest features needed to completely support data types. These features included the ability to define types and to fully describe relationships – something used widely, but maintained entirely by the user. In POSTGRES, the database understood relationships, and could retrieve information in related tables in a natural way using rules. POSTGRES used many of the ideas of Ingres, but not its code.

PostgreSQL includes built-in synchronous replication that ensures that, for each write transaction, the master waits until at least one replica node has written the data to its transaction log. Unlike other database systems, the durability of a transaction (whether it is asynchronous or synchronous) can be specified per-database, per-user, per-session or even per-transaction. PostgreSQL to reflect its support for SQL. After a review in 2007, the development team decided to keep the name PostgreSQL and the alias Postgres from single machines to data warehouses or Web services with many concurrent users. Because PostgreSQL is immune to dirty reads, requesting a Read Uncommitted transaction isolation level provides read committed instead. PostgreSQL supports full serializability via the serializable snapshot isolation (SSI) method. It is the default database for macOS Server, and is also available for Linux, FreeBSD, OpenBSD, and Windows. This can be useful for workloads that do not require such guarantees, and may not be wanted for all data as it slows down performance due to the requirement of the confirmation of the transaction reaching the synchronous standby.

Common Use cases of PostgreSQL

The following are the common use cases of PostgreSQL.

1) A robust database in the LAPP stack

LAPP stands for Linux, Apache, PostgreSQL, and PHP (or Python and Perl). PostgreSQL is primarily used as a robust back-end database that powers many dynamic websites and web applications.

2) General purpose transaction database

Large corporations and startups alike use PostgreSQL as primary databases to support their applications and products.

3) Geospatial database

PostgreSQL with the PostGIS extension supports geospatial databases for geographic information systems (GIS).

PostgreSQL feature

PostgreSQL has many advanced features that other enterprise-class database management systems offer, such as:

- User-defined types
- Table inheritance
- Sophisticated locking mechanism
- Foreign key referential integrity
- Views, rules, subquery
- Nested transactions (savepoints)
- Multi-version concurrency control (MVCC)

3.5 pgAdmin

pgAdmin is a free software project released under the PostgreSQL/Artistic license. The software is available in source and binary format from the PostgreSQL mirror network. Because compiling from source requires technical knowledge, it is recommended to install binary packages whenever possible.

pgAdmin 4 is a complete rewrite of pgAdmin, built using Python and Javascript/jQuery. A desktop runtime written in C++ with Qt allows it to run standalone for individual users, or the web application code may be deployed directly on a webserver for use by one or more users through their web browser. The software has the look and feel of a desktop application whatever the runtime environment is, and vastly improves on pgAdmin III with updated user interface elements, multi-user/web deployment options, dashboards and a more modern design.

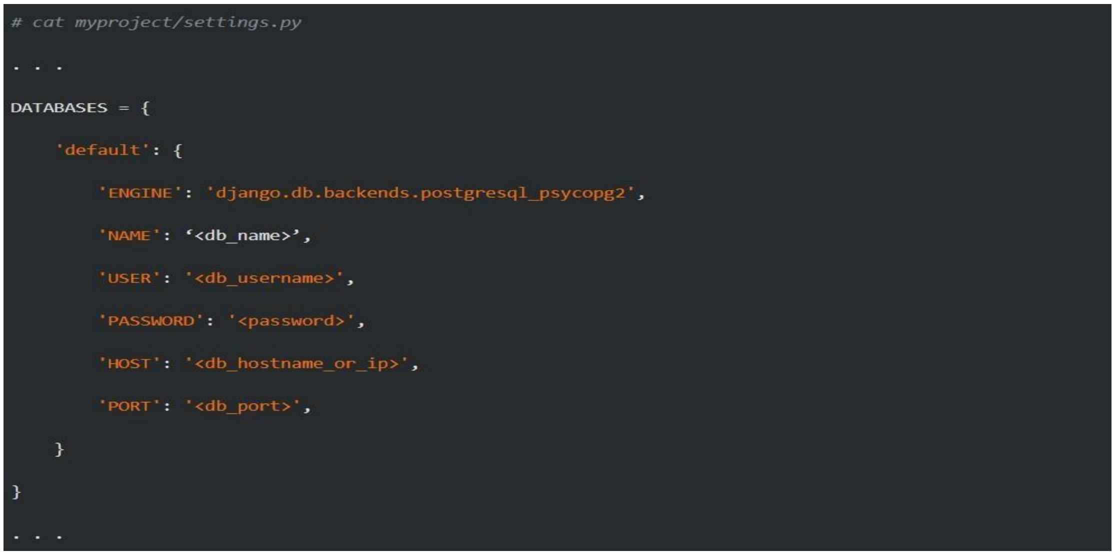
Tools

- Powerful query tool with colour syntax highlighting
- Fast datagrid for display/entry of data
- Graphical query plan display
- Grant Wizard for rapid updates to ACLs
- Procedural language debugger (supporting pl/pgsql and edb-spl)
- Schema diff tool for managing differences between schemas

3.5.1 Connection to the database

To connect to database, we write the code in the settings.py file of the project

- In the settings.py identify the database connection area.
- Add the name of database, username and password.
- Run the command “python manage.py make migrations”
- The database is created and can be viewed in the PgAdmin browser.



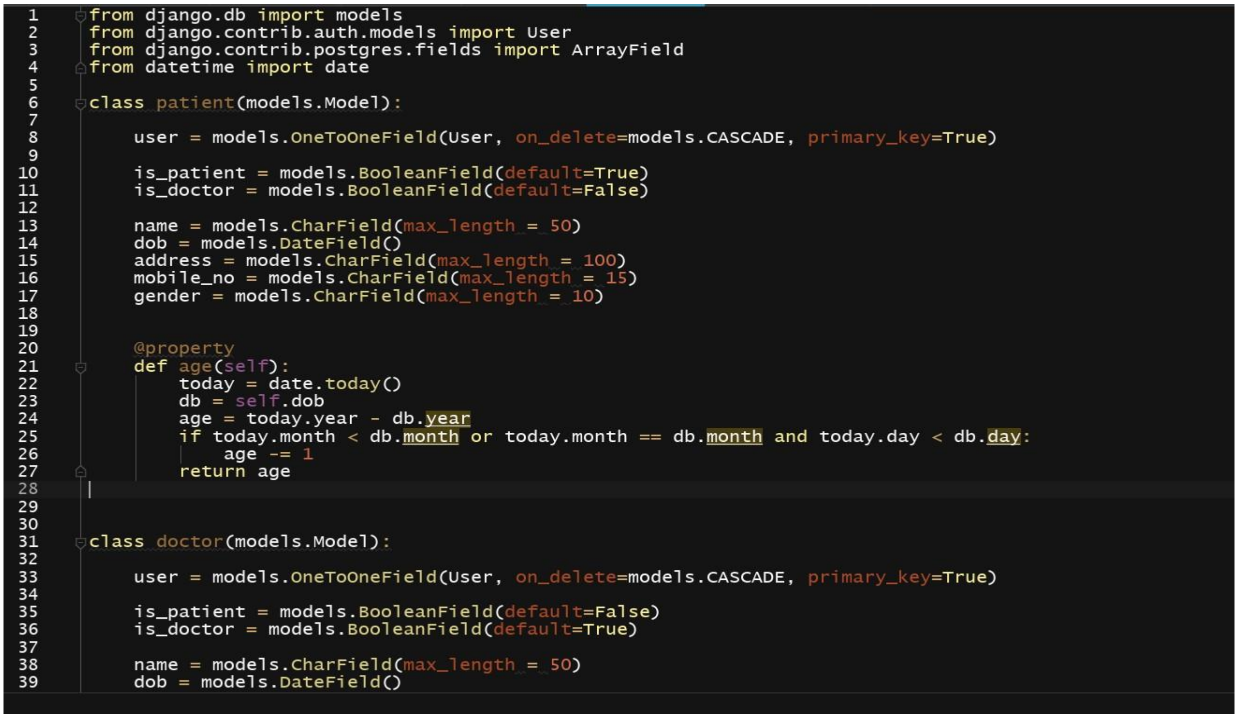
```
# cat myproject/settings.py

. . .

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql_psycopg2',

        'NAME': ‘<db_name>’,

        'USER': '<db_username>',

        'PASSWORD': '<password>',

        'HOST': '<db_hostname_or_ip>',

        'PORT': '<db_port>',

    }

}

. . .
```

FIG 3.7: CONNECTION TO DATABASE

3.5.2 Table Creation

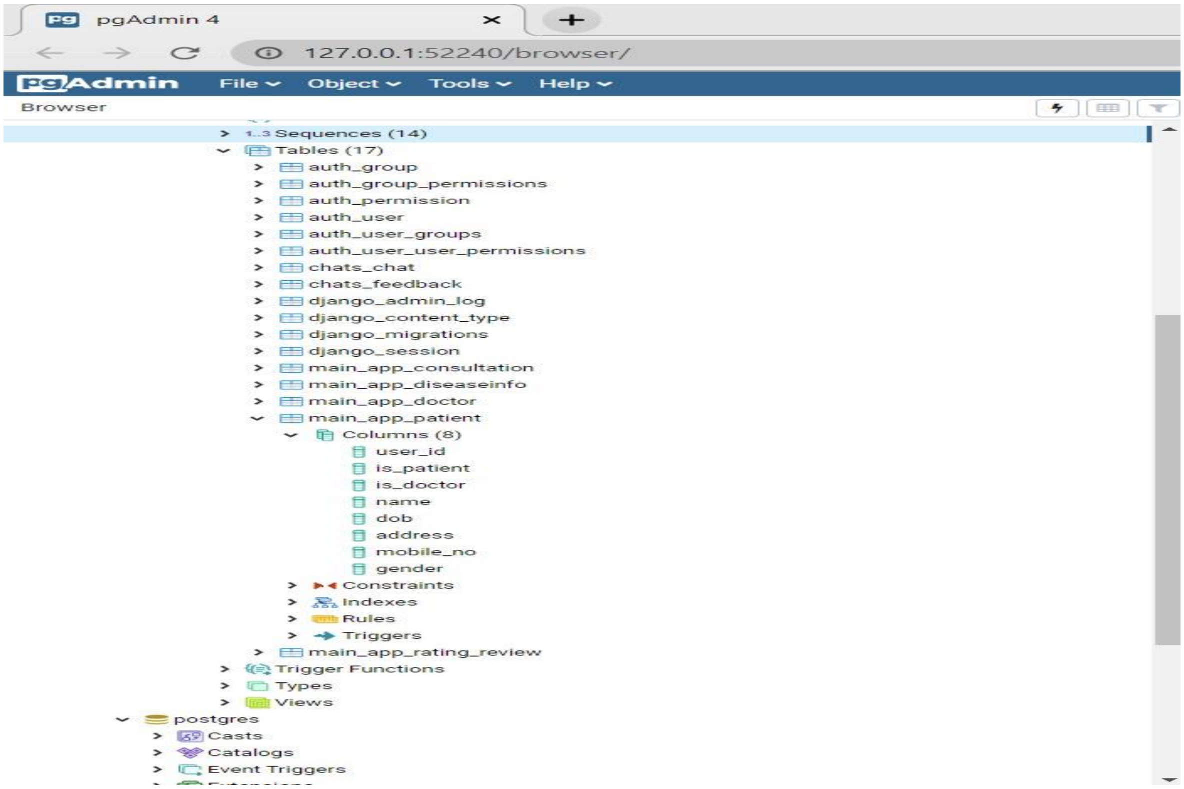
To create tables in Django we define it in model.py each class in the model represents the table

1. Open the models.py file in the project
2. Add the import “from django.db import models”
3. Create a class to represent a table in the database as “class table_name(models.Model)”
4. Add the table columns in the class as “column_name=models.TextField()”
5. Make the migrations to build the tables in the database “python manage.py makemigrations”
6. Tables are created and can be viewed on the pgadmin browser.



```
from django.db import models
from django.contrib.auth.models import User
from django.contrib.postgres.fields import ArrayField
from datetime import date

class patient(models.Model):

    user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True)

    is_patient = models.BooleanField(default=True)
    is_doctor = models.BooleanField(default=False)

    name = models.CharField(max_length = 50)
    dob = models.DateField()
    address = models.CharField(max_length = 100)
    mobile_no = models.CharField(max_length = 15)
    gender = models.CharField(max_length = 10)


    @property
    def age(self):
        today = date.today()
        db = self.dob
        age = today.year - db.year
        if today.month < db.month or today.month == db.month and today.day < db.day:
            age -= 1
        return age



class doctor(models.Model):

    user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True)

    is_patient = models.BooleanField(default=False)
    is_doctor = models.BooleanField(default=True)

    name = models.CharField(max_length = 50)
    dob = models.DateField()
```

FIG 3.8: TABLE CREATION



```
        > Sequences (14)
        ⌄ Tables (17)
            > auth_group
            > auth_group_permissions
            > auth_permission
            > auth_user
            > auth_user_groups
            > auth_user_user_permissions
            > chats_chat
            > chats_feedback
            > django_admin_log
            > django_content_type
            > django_migrations
            > django_session
            > main_app_consultation
            > main_app_diseaseinfo
            > main_app_doctor
            ⌄ main_app_patient
                ⌄ Columns (8)
                        user_id
                        is_patient
                        is_doctor
                        name
                        dob
                        address
                        mobile_no
                        gender
                > Constraints
                > Indexes
                > Rules
                > Triggers
            > main_app_rating_review
        > Trigger Functions
        > Types
        > Views
⌄ postgres
    > Casts
    > Catalogs
    > Event Triggers
```

FIG 3.9: TABLE CREATED

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

In architectural model it contains two databases: Patient Records database and Disease/Symptoms database. Four web services are used to implement the SOA. They are Pattern matching, recent trends, differential diagnosis and recent differential diagnosis. The patient Record database contains all the patient information from all the hospitals in the network.

Diseases/Symptoms database is a centralized database. It contains the list of existing known diseases and their corresponding symptoms along with their weights. These databases are replicated across various servers and these replicated servers are used to achieve the fault tolerance with concurrency protocols to achieve atomic transactions. First the doctor retrieves the symptoms from the patient record database. After retrieving the symptoms, the doctor identify whether any symptom related diseases contains in the Diseases/Symptoms database. Here the pattern matching service is activated. If any diseases match with Symptoms means list out all the possible matched symptoms and presents the result to the doctors. If the doctors not satisfied with results, compare to recent history and recent trend service must be activated. This service makes use of the Diseases/symptoms database and Patient Record database and the result obtained from pattern matching service to get results. After comparing the diseases to the recent history, cluster the shortlisted diseases.

The probability may be computed based on the distance vector. The highest priority cluster produces the accurate result. Finally, to avoid the vagueness in decisions, the doctor use differential diagnosis and recent diagnosis features use Diseases/symptoms database and Patient record database and result acquired from recent trend services to gain the results. Since the large medical data, using simple client server architecture would not produce the effective aforesaid services and would increase the response time of the system. Finally we conclude that SOA was well suited to apply this system because it improve the delivery of important information and sharing of data across the community of healthcare professionals more practical in cost, security and risk deployment. In various existing EHRs, SOA is more essential for data providers to this system, are already using this very successful and efficient architecture.

This service makes use of the Diseases/symptoms database and Patient Record database and the result obtained from pattern matching service to get results. After comparing the diseases to the recent history, cluster the shortlisted diseases.

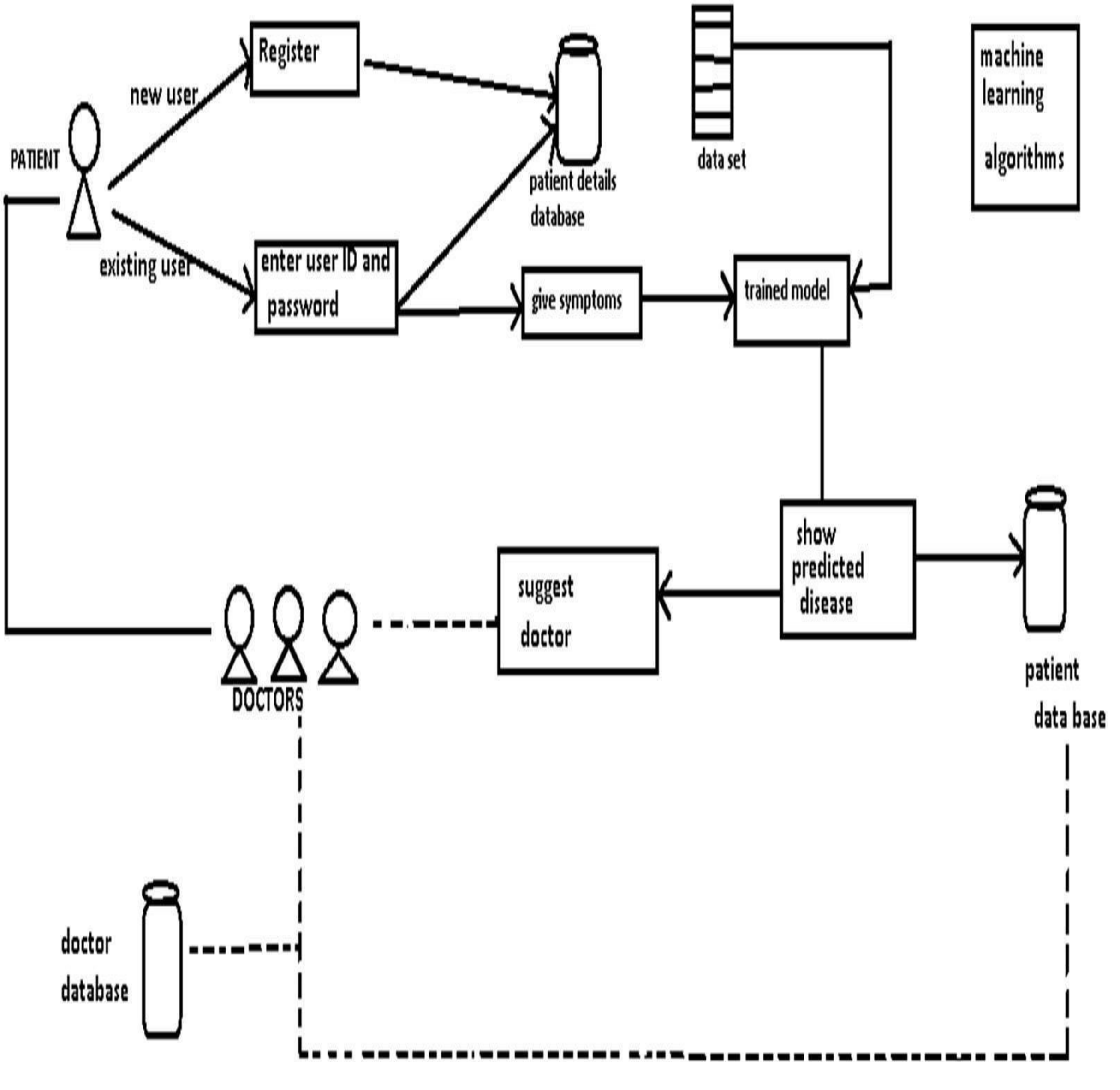


FIG 4.1: SYSTEM ARCHITECTURE

4.2 Use Case Diagram

The Use Case diagram of the project disease prediction using machine learning consist of all the various aspects a normal use case diagram requires. This use case diagram shows how from starting the model flows from one step to another, like he enter into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the use case diagram of all the entities are linked to each other where the user gets started with the system.

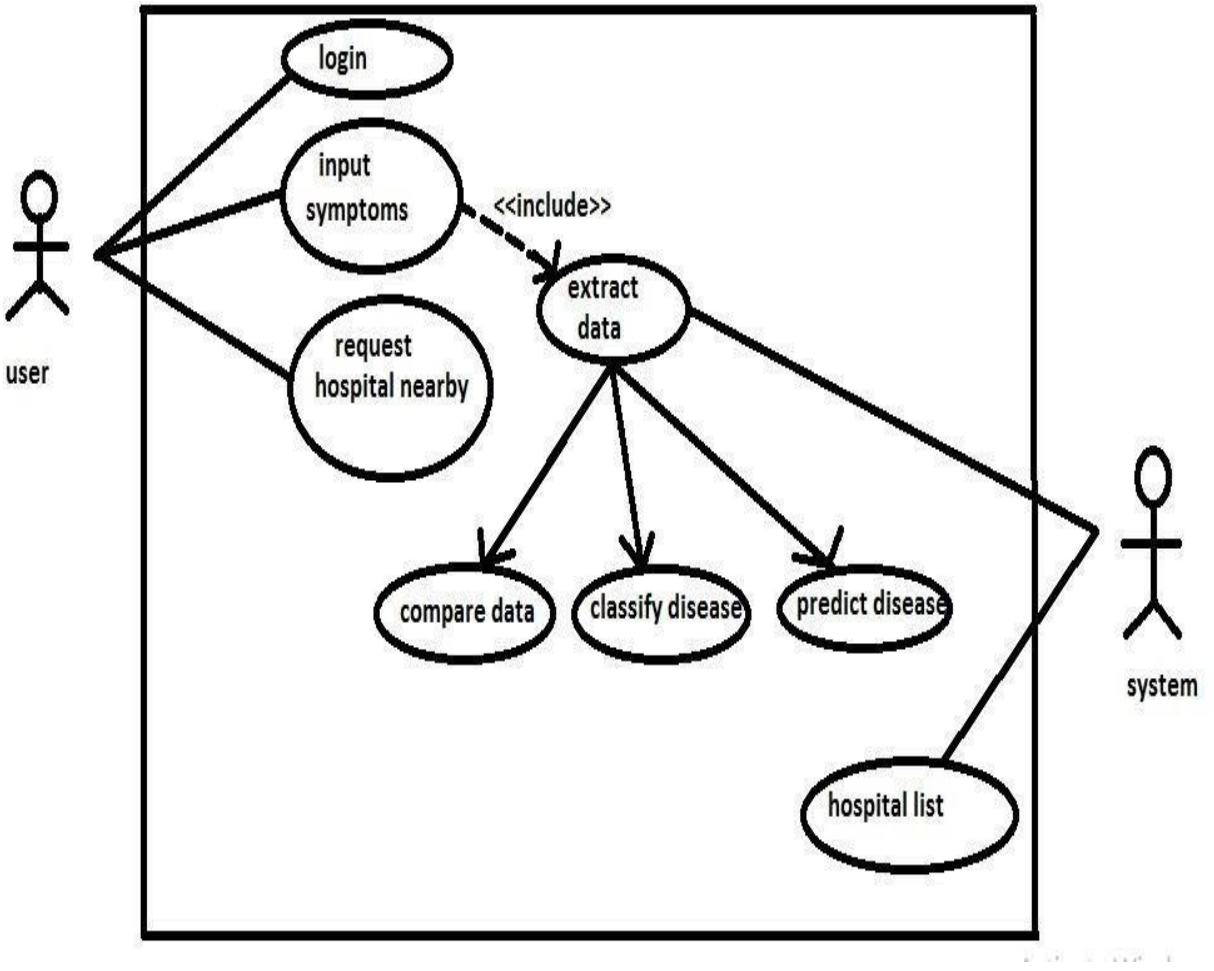


FIG 4.2: USE CASE DIAGRAM

CHAPTER 5

REQUIREMENTS SPECIFICATION

5.1 Hardware requirements

- 4Gb RAM
- I5 Processor ranging 1.90GHz -3.80GHz
- Hard Disk 500Gb recommended
- 512 KB cache memory or more

5.2 Software requirements

- PyCharm
- pgAdmin4
- Django
- PostgreSQL
- Operating System: Windows 10

CHAPTER 6

IMPLEMENTATION

6.1 Modules Implementation

Implementation of a system must be done step by step in order achieve the object of developed system. Identifying the modules makes it easier to work on developing the system. In the proposed system, there are six modules that have been identified. These six modules are the six different features which put together will form a whole system. six modules that have been identified are:

- Input module
- Doctor
- User
- Admin
- Prediction system
- Output model

Above mentioned modules are explained briefly in the below sections.

6.1.1 Input module

Input module is the user module, where user can enter the symptoms from the given set of predefined symptoms. User can select multiple symptoms which user is suffering from. These entered symptoms are analyzed and the disease is predicted based on the entered symptoms.

This input module is the important module for the system, were user enters symptoms and prediction is done by the input module. Whole prediction processes is done through the Input module. User can be a patient or any user or even doctor, were doctors can use this system to analyze the disease in a better way.

6.1.2 Doctor’s module

In this module the doctors can create their account by sing in to the system with their specialization, which will be displayed to the user after prediction of disease. User can see doctor’s details by clicking in to the doctor recommended in the prediction page.

This module also support user to contact with the patient, where patient can take appointment from the specialized doctor. If any issues regarding health, then patient can text to doctors where doctor suggest some medications for the patient during emergency.

Doctor is authenticated in the sign-up form with the username and password, with successful authentication doctor can check for appointments or any patient text. Patient can clarify any issue through chat box available. This doctors’ details are stored in the database for authentication, which increases security.

6.1.3 User

User module plays an important role in this system. User can access to the system by signup to the system, with user details after successful account creation user is directed to prediction page were user should select the symptoms from the set of predefined symptoms, after selection the system predicts the disease associated and provides link which leads to more elaborate explanation about the particular disease. User is also be provided with the doctor specialized in that field.

User is authenticated in the sign-up form with the username and password, with successful authentication user is directed to prediction page. User can take appointments from doctors and even clarify any issues regarding health through chat box provided. User’s details are stored in the database for authentication, which increases security.

User is also provided with doctors and hospitals for unpredicted diseases also were user can refer to any hospital or doctor according to their convenience.

6.1.4 Admin

Users and doctors have unrestricted access to system, were as Admin can access all the modules in the system. Admin can update or remove modules based on the feedback provided by user and doctor. Admin modules mainly adds improved version of the system. Patient can clarify any issue through chat box available. This doctors’ details are stored in the database for authentication, which increases security. Admin can control unrestricted access done by any user or doctor, and appropriate action can be taken accordingly.

6.1.5 Output module

This is the output which the user receives where it displays the predicted disease and the recommended doctor. This is the last module of the system which provides result page. This module provides output to user after selection of symptoms, and analyzation of given symptoms sending trained set in to various algorithm and output is predicted to the user.

CHAPTER 7

SYSTEM TESTING

7.1 Definition

System testing is a development procedure where programmers create tests as they develop software. The tests are simple short tests that test functionally of a particular unit or module of their code, such as a class or function. Using open-source libraries like c unit, opp unit and run it these tests can be automatically run and any problems found quickly. As the tests are developed in parallel with the source unit test demonstrates its correctness.

7.2 Types of Testing

7.2.1 Unit testing

Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code, sometimes called Scaffolding, may be necessary to support an individual test. This type of testing is driven by the architecture and implementation teams. This focus is also called black-box testing because only the details of the interface are visible to the test. Limits that are global to a unit are tested here.

In the construction industry, scaffolding is a temporary, easy to assemble and disassemble, frame placed around a building to facilitate the construction of the building. The construction workers first build the scaffolding and then the building.

Later the scaffolding is removed, exposing the completed building similarly, in software testing, one particular test may need some supporting software. This software establishes can a correct evaluation of the test take place. The scaffolding software may establish state and values for data structures as well as providing dummy external functions for the test. Different scaffolding software may be needed form one test to another test. Scaffolding software rarely is considered part of the system.

Sometimes the scaffolding software becomes larger than the system software being tested. Usually, the scaffolding software is not of the same quality as the system software and frequently is quite fragile. A small change in test may lead to much larger changes in the scaffolding.

Internal and unit testing can be automated with the help of coverage tools. Analyzes the source code and generated a test that will execute every alternative thread of execution. Typically, the coverage tool is used in a slightly different way. First the coverage tool is used to augment the source by placing information prints after each line of code.

Table 7.1: Patient Login Test Cases

Test Case No.	1
Name of Test	Patient Login Page
Test Case Description	When patient enters correct username and password it should open the profile page.
Sample Input	Patient enters username and password.
Expected Output	It opens the patient profile page.
Actual Output	It opens the patient profile page.
Remarks	Pass
Comments	Working properly.

Table 7.2: Doctor Login Test Cases

Test Case No.	2
Name of Test	Doctor Login Page
Test Case Description	When doctor enters correct username and password it should open the profile page.
Sample Input	Doctor enters username and password.
Expected Output	It opens the doctor profile page.
Actual Output	It opens the doctor profile page.
Remarks	Pass
Comments	Working properly

Table 7.3: Admin Login Test Cases

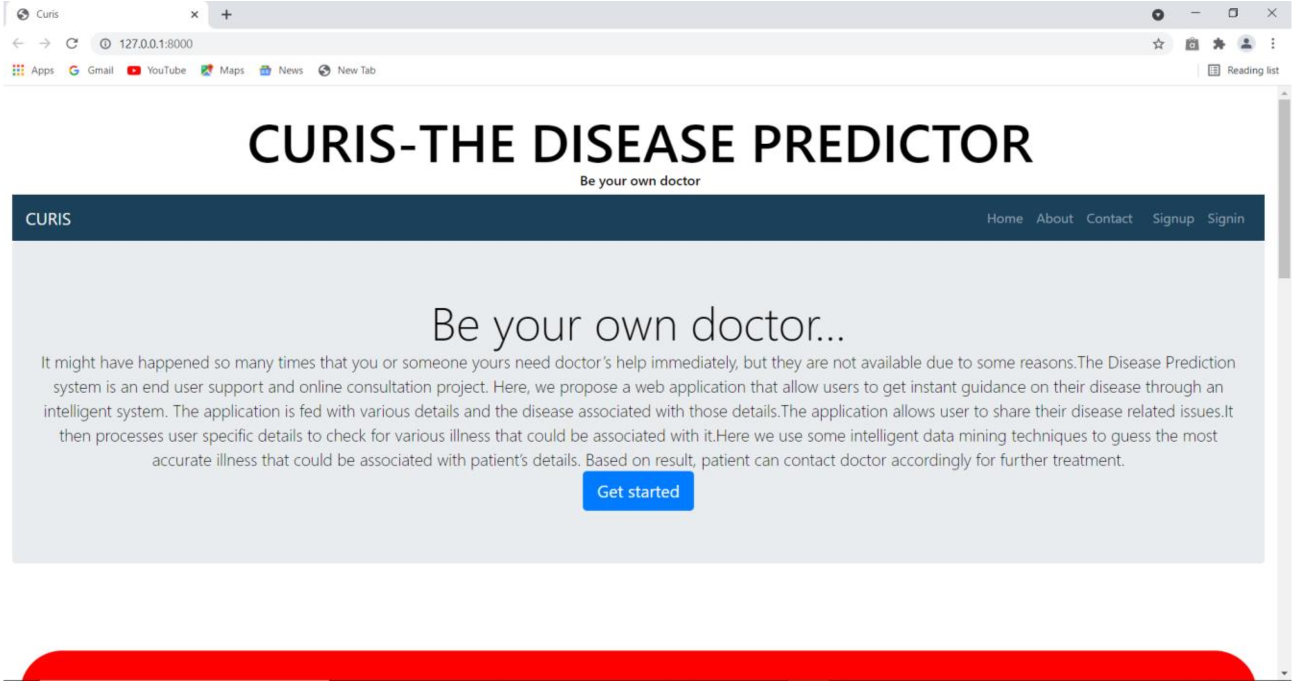
Test Case No.	3
Name of Test	Admin Login Page
Test Case Description	When admin enters correct username and password it should open the admin page.
Sample Input	Admin enters username and password.
Expected Output	It opens the admin profile page.
Actual Output	It opens the admin profile page.
Remarks	Pass
Comments	Working Properly.

Table 7.4: Disease Prediction Test Cases

Test Case No.	4
Name of Test	Checking the disease
Test Case Description	Displaying the predicted disease
Sample Input	Selecting the symptoms from the list
Expected Output	Displaying the predicted disease with confidence score
Actual Output	Displaying the predicted disease with confidence score
Remarks	Pass
Comments	Working Properly

CHAPTER 8

SNAPSHOTS



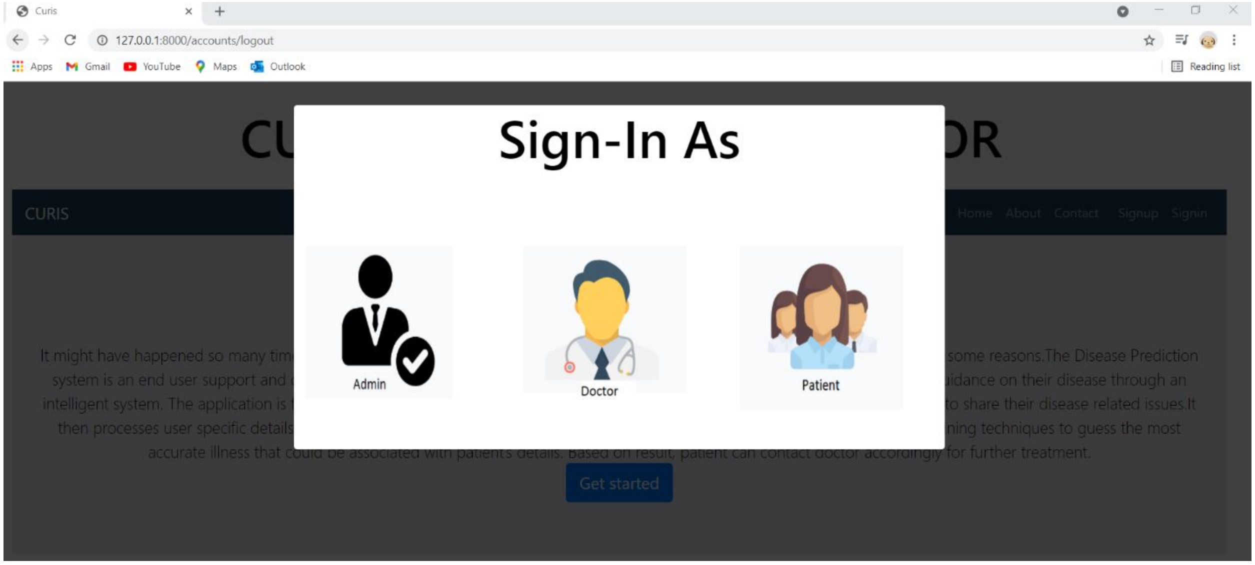
CURIS-THE DISEASE PREDICTOR

Be your own doctor

Be your own doctor...

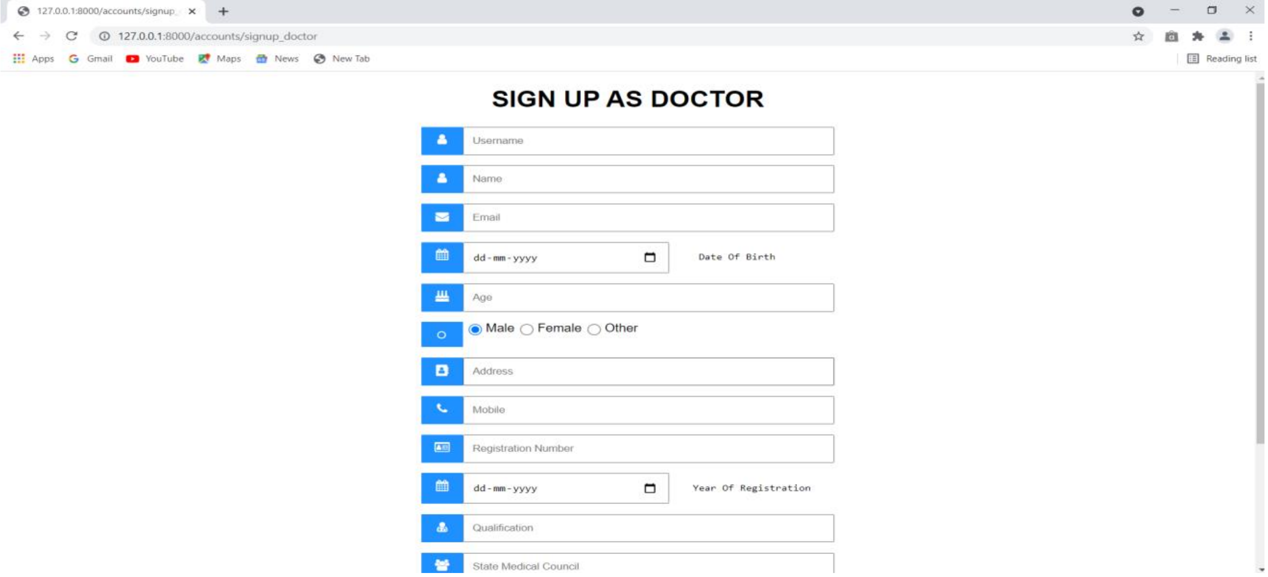
It might have happened so many times that you or someone yours need doctor's help immediately, but they are not available due to some reasons.The Disease Prediction system is an end user support and online consultation project. Here, we propose a web application that allow users to get instant guidance on their disease through an intelligent system. The application is fed with various details and the disease associated with those details.The application allows user to share their disease related issues.It then processes user specific details to check for various illness that could be associated with it.Here we use some intelligent data mining techniques to guess the most accurate illness that could be associated with patient's details. Based on result, patient can contact doctor accordingly for further treatment.

FIG 8.1: HOME PAGE



Sign-In As

FIG 8.2: SIGN IN AS ADMIN, DOCTOR AND PATIENT


127.0.0.1:8000/accounts/signup_
127.0.0.1:8000/accounts/signup_doctor
Apps
Gmail
YouTube
Maps
News
New Tab
Reading list

SIGN UP AS DOCTOR

Username

Name

Email

dd-mm-yyyy Date Of Birth

Age

● Male ○ Female ○ Other

Address

Mobile

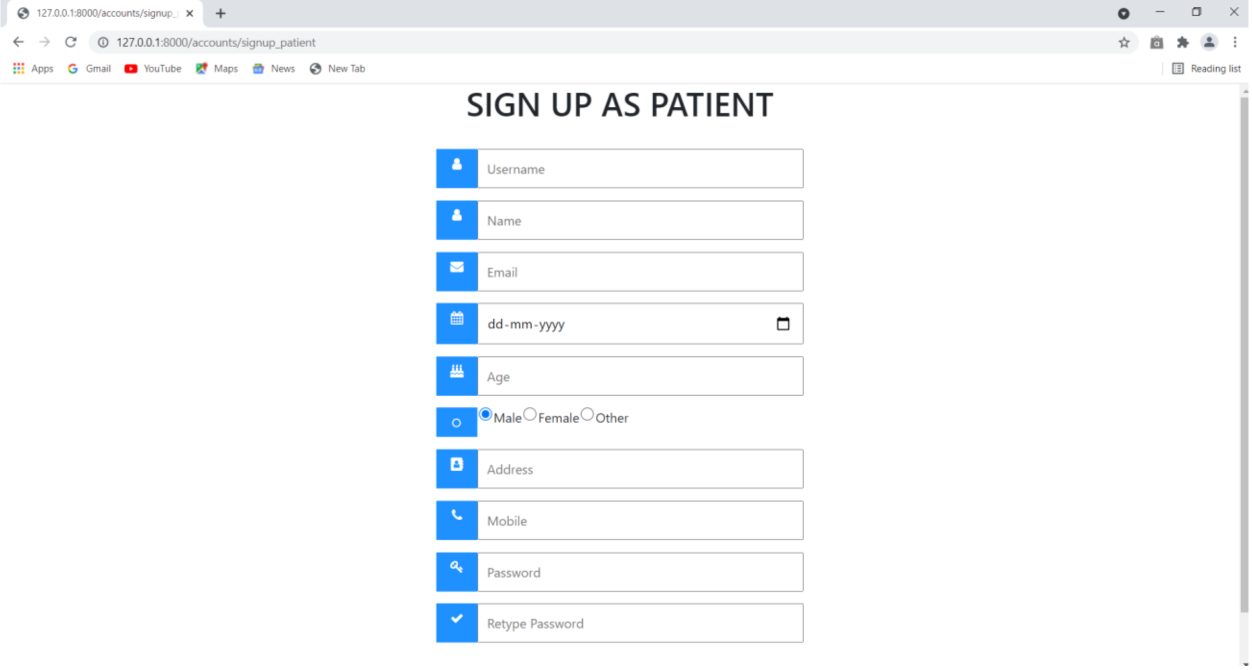
Registration Number

dd-mm-yyyy Year Of Registration

Qualification

State Medical Council

FIG 8.3: SIGNUP AS DOCTOR PAGE


127.0.0.1:8000/accounts/signup_
127.0.0.1:8000/accounts/signup_patient
Apps
Gmail
YouTube
Maps
News
New Tab
Reading list

SIGN UP AS PATIENT

Username

Name

Email

dd-mm-yyyy

Age

● Male ○ Female ○ Other

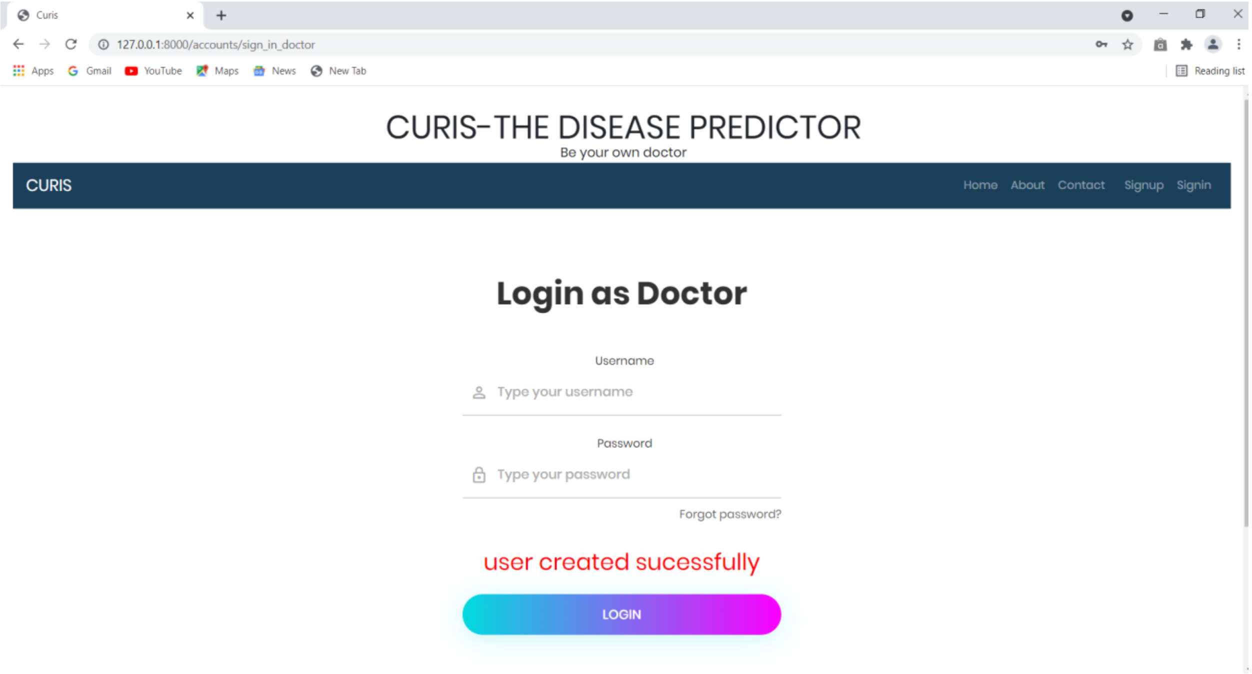
Address

Mobile

Password

Retype Password

FIG 8.4: SIGNUP AS PATIENT PAGE



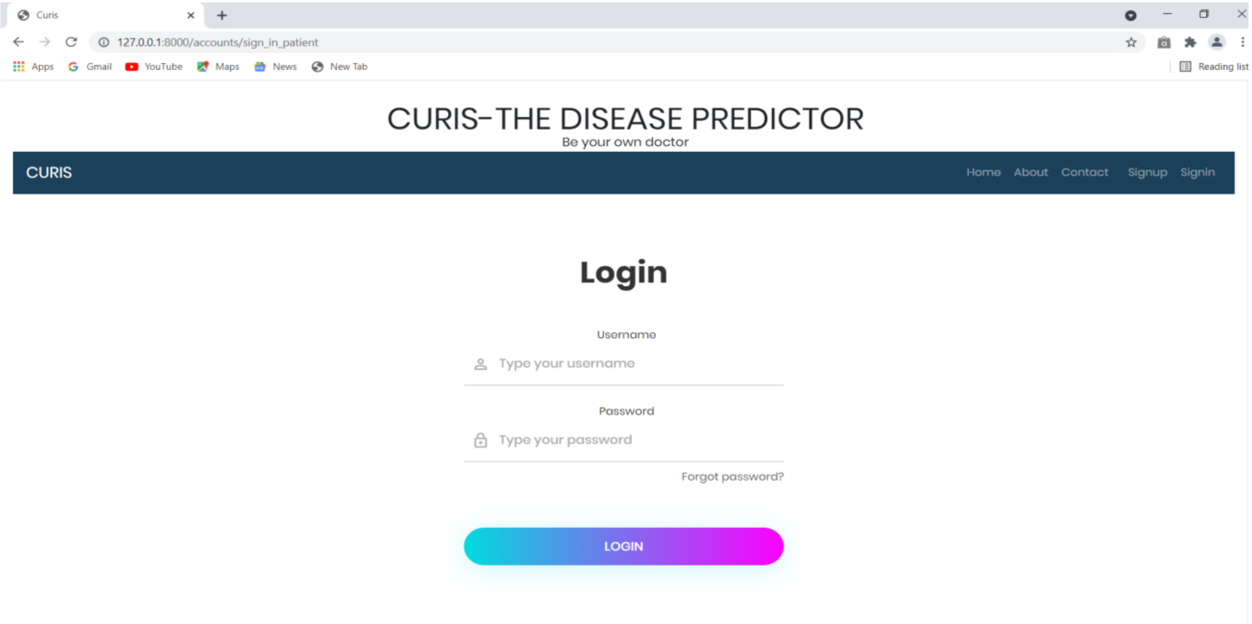
CURIS-THE DISEASE PREDICTOR

Be your own doctor

Login as Doctor

user created sucessfully

FIG 8.5: DOCTOR LOGIN PAGE

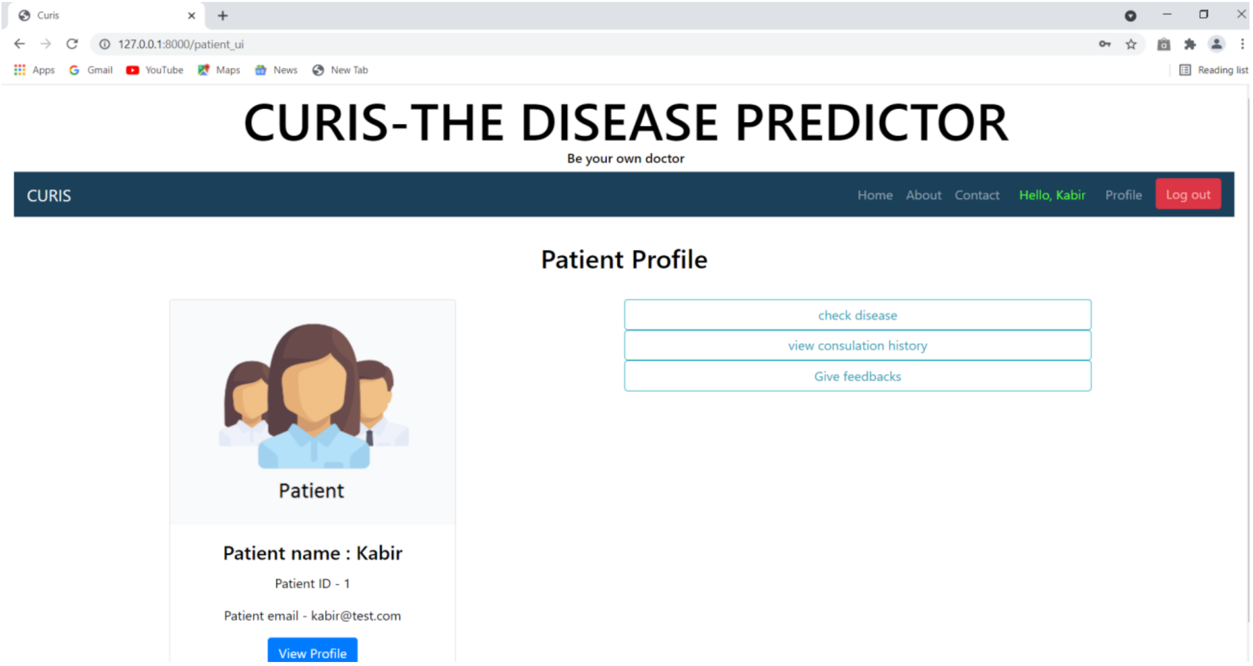


CURIS-THE DISEASE PREDICTOR

Be your own doctor

Login

FIG 8.6: PATIENT LOGIN PAGE



CURIS-THE DISEASE PREDICTOR

Be your own doctor

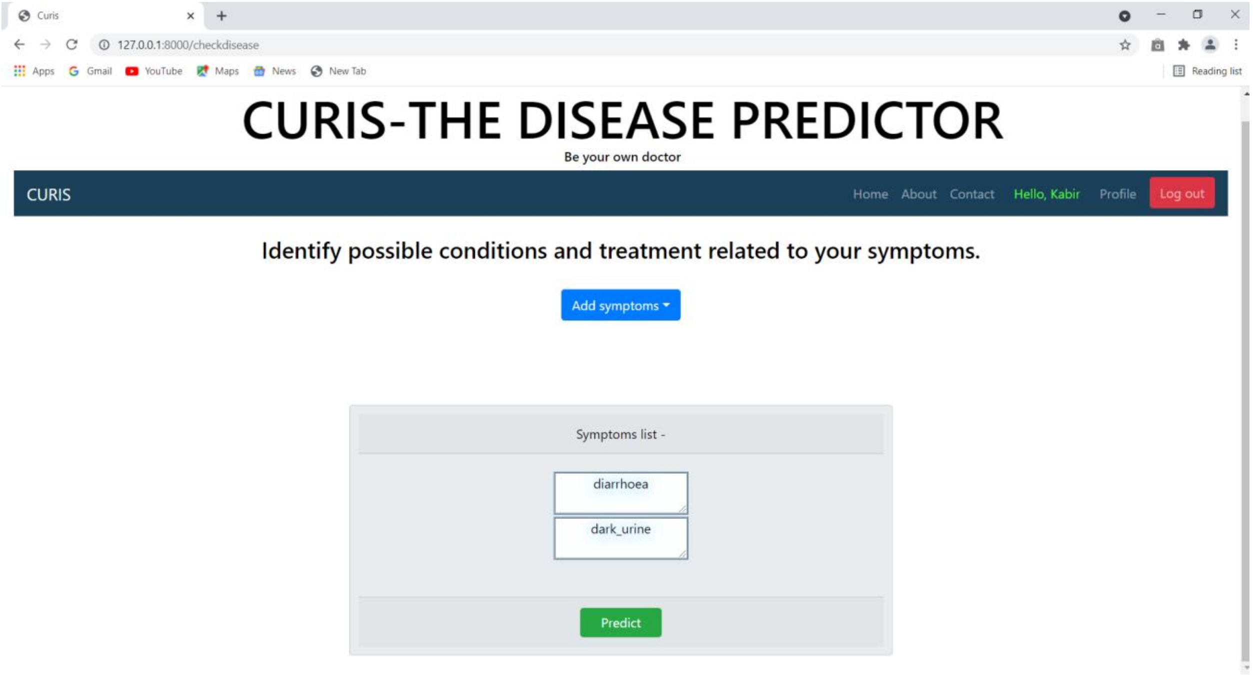
Patient Profile

Patient name : Kabir

Patient ID - 1

Patient email - kabir@test.com

FIG 8.7: PATIENT PROFILE PAGE



CURIS-THE DISEASE PREDICTOR

Be your own doctor

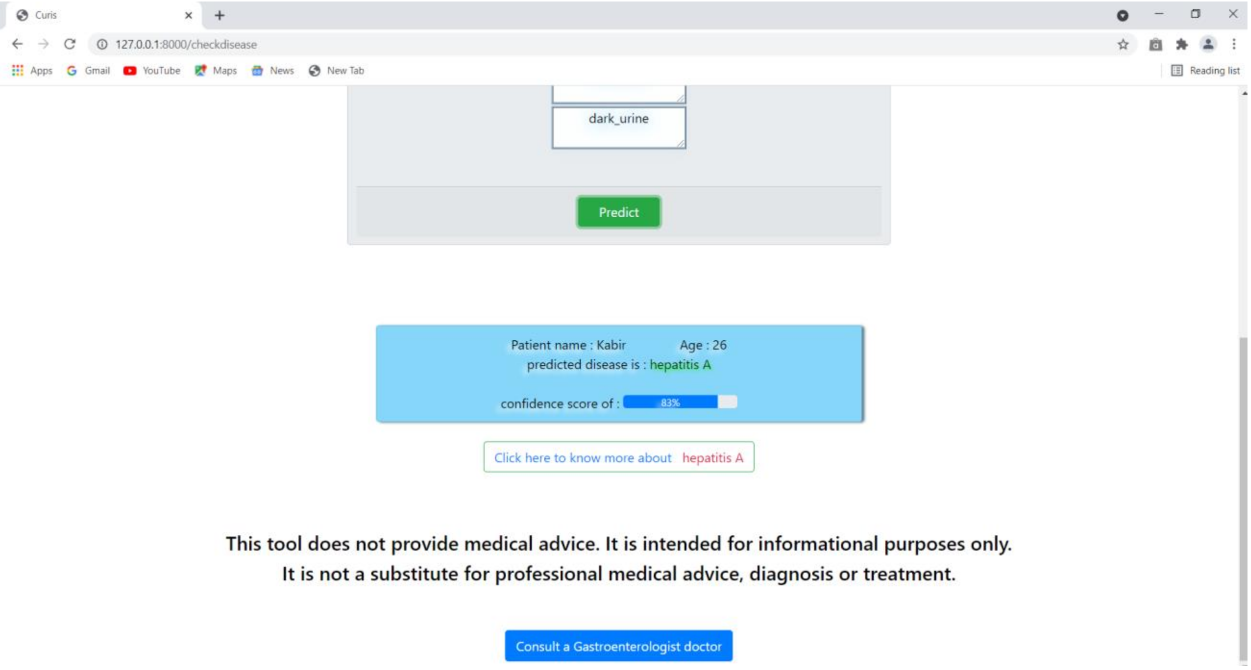
Identify possible conditions and treatment related to your symptoms.

Symptoms list -

diarrhoea

dark_urine

FIG 8.8: SYMPTOM PAGE



dark_urine

Patient name : Kabir Age : 26
predicted disease is : hepatitis A

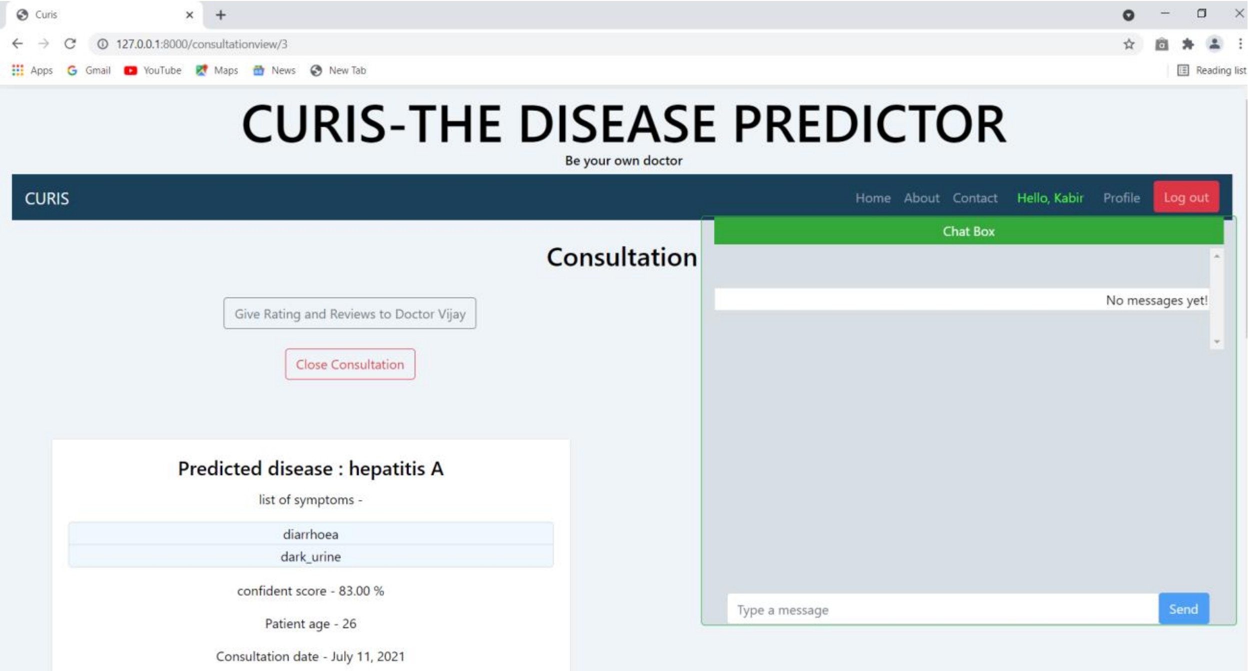
confidence score of : 83%

Click here to know more about hepatitis A

This tool does not provide medical advice. It is intended for informational purposes only.
It is not a substitute for professional medical advice, diagnosis or treatment.

Consult a Gastroenterologist doctor

FIG 8.9: PREDICTION PAGE



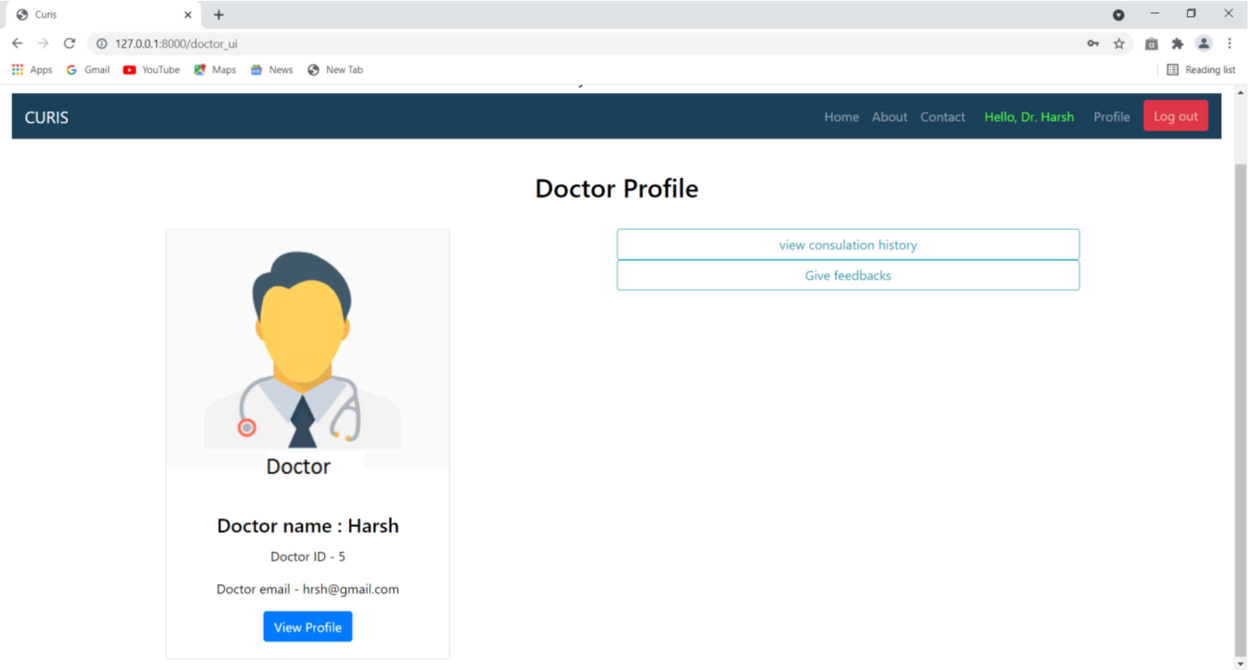
CURIS-THE DISEASE PREDICTOR
Be your own doctor

Consultation

No messages yet!

Predicted disease : hepatitis A
list of symptoms -
diarrhoea
dark_urine
confident score - 83.00 %
Patient age - 26
Consultation date - July 11, 2021

FIG 8.10: CONSULTATION PAGE



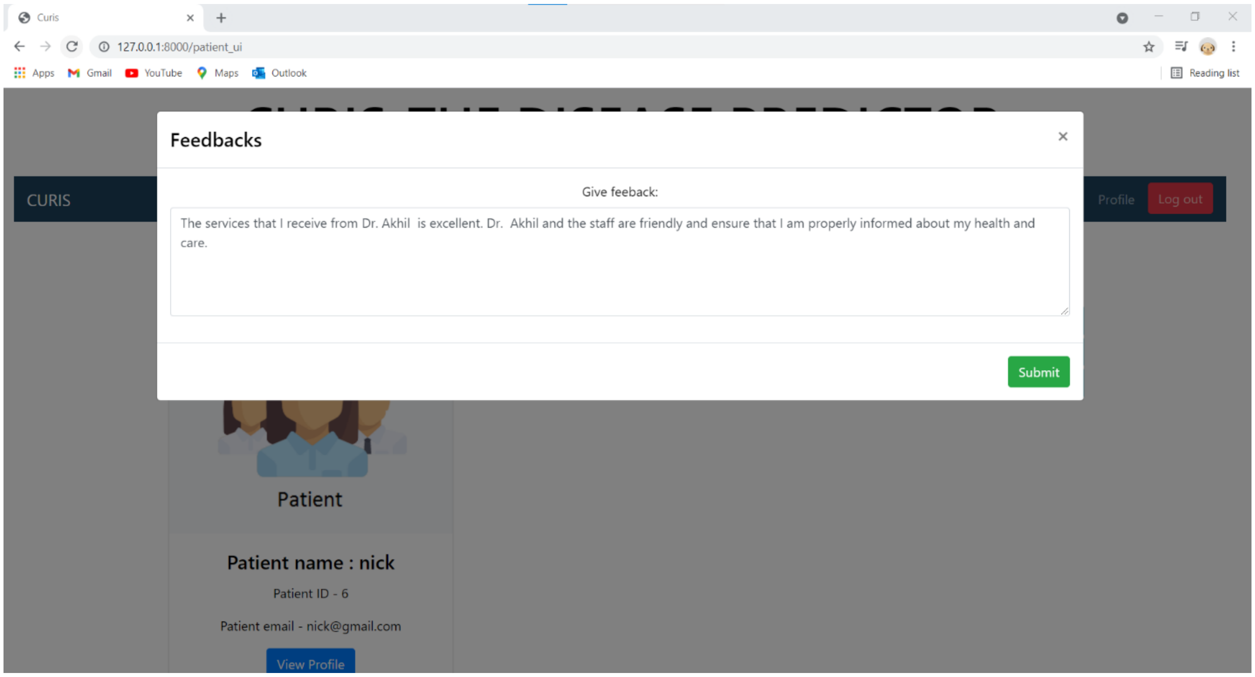
Doctor Profile

Doctor name : Harsh

Doctor ID - 5

Doctor email - hrsh@gmail.com

FIG 8.11: DOCTOR PROFILE PAGE



Feedbacks

Give feeback:

The services that I receive from Dr. Akhil is excellent. Dr. Akhil and the staff are friendly and ensure that I am properly informed about my health and care.

Patient name : nick

Patient ID - 6

Patient email - nick@gmail.com

FIG 8.12: FEEDBACK PAGE