BLOCKCHAIN AUTHENTICATOR AND MACHINE LEARNING RECOMMENDER SYSTEM

A PROJECT REPORT

Submitted by

JASKIRAT SINGH BHATIA [Reg No: RA1511003010011]

Under the Guidance of

Dr. M. PRAKASH

(Associate Professor, Department of Computer Science and Engineering)

In Partial Fulfillment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM UNIVERSITY, KATTANKULATHUR- 603 203

APRIL 2019

Department of Computer Science and Engineering
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **BTech / Computer Science Engineering(CSE)**

Student Name : **Jaskirat Singh Bhatia, Anoop Singh Bhatia**

Registration Number : **RA1511003010011, RA1511003010011**

Title of Work : **BlockChain Authenticator and Machine Learning Recommender System**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s).
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalised in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM UNIVERSITY, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this thesis titled "**BLOCKCHAIN AUTHENTICATOR AND MCHINE LEARNING RECOMMENDER SYSTEM"** is the bonafide work of **Mr. JASKIRAT SINGH BHATIA AND Mr. ANOOP SINGH LABANA** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. M. PRAKASH
SUPERVISOR
Associate Professor
Department of Computer Science and Engineering

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Department of Computer Science and Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

The biggest concern in this era is trust; No one is willing to trust anyone. The latest trend in development of trust is the Blockchain technology and we now bring it into the medical branch. EHRs (Electronic Health Records) are the basis of storing the health records for the patients these days, but they are not decentralised and they can be modified or deleted by an authorised person or by a hacker easily. The base for this project is to give all the power to the patient by implementing the EHRs System on blockchain, making the records unmodifiable and not deletable show-ing transparency with whomsoever we want to share our records. The other part of our project is based on Machine Learning which focusses on predicting a disease based on the symptoms shown by the patient. We will be using SVM, Logistic Regression and Decision Tree algorithm to predict the disease and in the end, we will develop a voting system which takes the votes of the disease predicted by the 3 algorithms and select the majority. Then as per the disease predicted, a doctor for the same will be recommended using a recommender system.

ACKNOWLEDGEMENT

We express our humble gratitude to Dr. Sandeep Sancheti, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dr. C. Muthamizhchelvan, Director, Faculty of Engineering and Technology, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank Dr. B. Amutha, Professor & Head, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Academic Advisor Dr. K. Annapurani, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for her great support at all the stages of project work.

We would like to convey our thanks to our Panel Head, Dr. M. Prakash, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for his / her inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, T. Senthil Kumar, Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, Dr. M. Prakash, Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, for providing me an opportunity to pursue my project under his mentorship. He / She provided me the freedom and support to explore the research topics of my interest. Her / His passion for solving the real problems and making a difference in the world has always been inspiring.

We sincerely thank staff and students of the Computer Science and Engineering Department, SRM Institute of Science and Technology, for their help during my research. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Jaskirat Singh Bhatia
Anoop Singh Labana

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

EHR	Electronic Health Record
ML	Machine Learning
RSA	Rivest-Shamir-Adleman
UI	User Interface
API	Application Program Interface

CHAPTER 1

INTRODUCTION

1.1 Blockchain and Machine Learning Recommending System

Now a day's Blockchain and Machine Learning (ML), plays a great role in different fields or areas among the Health Care System. This leads to various studies and re-searches being conducted to selected health care accomodations. It is necessary to ensure a technologically appropriate system designed to fully employ the Blockchain and Ma-chine Learning technology for the maximal benefit in the health care society.

Computers here have a great applicability to store data robustly and ease the access on them in a brief burst of time

In order to store the consumer information securely, a blockchain is being build. The blockchain is a robust, integrated, secure and trustworthy technology. The blockchain system encrypts the data stored in each layer with the data stored in the next layer, making it next to impossible to crack it without the key which brings us to the next technology in Blockchain that is the public/private key system. The public/private key of Blockchain works like any other system, that is all the hospitals have the public key of the patient and can use it to enter the details of the patient, and the patient will have his private key which he and only he can use to access his data, hence making it very secure.

ML using the algorithms of Deep Learning will be used to a high level of extent in our project. Deep Learning algorithms on clustering to figure out the type of disease the patient has and informing the patient about it. Also using Machine Learning Recom-mending algorithms to recommend the most suitable doctor for that particular disease nearest to the patient.

1.2 Problem Statement

Data is one thing in this world everyone is after. The ley man is afraid to give up his personal details into the world for privacy reasons. The lay man in this world is also very busy and doesn’t has enough time to spend on his health and take care of it properly. Thus, our aim is to develop a system which is secure and also one which saves the time of the user by predicting his disease from his symptoms and recommend the most suitable doctor accordingly.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Survey

Human services might be an immense application circumstance of blockchain, and blockchains used in help region unit known as wellbeing blockchain. As a rule, blockchain squares zone unit open and consequently the exchanges in them region unit open. On the off chance that some security information are engaged with these exchanges, they will be spilled. Inferable from help framework including a phenomenal arrangement of protection data, certain security components ought to be intended to shield this protection data in wellbeing blockchain. Besides, in light of the fact that the center of insurence components is that the key administration plots, the reasonable key oversee ment plans should be planned well before blockchains might be used to help framework. Here, as per the highlights of wellbeing blockchain, the creators utilize a body identifier net-work to style a light-weight reinforcement and prudent recuperation topic for keys of wellbeing blockchain. The creator's examinations demonstrate that the plan has huge insurence and perfor-mance, and it tends to be utilized to ensure protection information on wellbeing blockchain adequately and to showcase the apparatus of wellbeing blockchain.

In view of the customary lexicon learning techniques, that ignores the connection be-tween the example and subsequently the wordbook iota, we tend to propose the weighted component to join the example with the wordbook molecule amid this dad per. Then, the customary word reference learning strategy is inclined to cause over-tting for patient classication of the constrained preparing informational collection. Subsequently, this paper embraces l2-standard regularization requirement, which understands the confinement of the model space, and upgrades the speculation capacity of the model and maintains a critical distance from over-tting somewhat. Contrasted and the past shallow word reference learning, this paper proposed the insatiable profound lexicon learning.

In light of the conventional lexicon learning techniques, that ignores the connection be-tween the example and in this manner the wordbook molecule, we tend to propose

the weighted component to connect the example with the wordbook molecule amid this dad per. In the mean time, the customary word reference learning strategy is inclined to cause over-tting for patient classication of the restricted preparing informational index. Subsequently, this paper receives l2-standard regularization imperative, which understands the constraint of the model space, and improves the speculation capacity of the model and maintains a critical distance from over-tting somewhat. Contrasted and the past shallow lexicon learning, this paper proposed the avaricious profound word reference learning.

Persistent similitude learning means to discover fitting separation measurements to quantify dad tient sets for a particular errand. To catch the verifiable data of patient' record, a legitimate method to speak to longitudinal EHR is important. Additionally, we need an approach to gain proficiency with the likeness degree or separation between each pair of patients. We propose two patient comparability learning systems on EHR dataset. The crude EHRs are feed into a CNN model which catches the back to back consecutive data to get familiar with a vector portrayal. At that point delicate max based administered grouping strategy and triplet misfortune based separation metric learning technique are utilized to get familiar with the comparable ity of patient sets. Exploratory outcomes on sickness forecast and patient bunching demonstrate that CNN can more readily speak to the longitudinal EHR successions, and our endtoend closeness structures beat best in class separate measurement learning strategies.

The huge capability of this innovation shows up wherever, as of recently, a believed outsider was fundamental for the settlement of market administrations. With Blockchain, direct exchanges abruptly conceivable, whereby a focal on-screen character, who controlled the information, earned commission or even interceded in a blue penciling design, can be killed. This troublesome character, which underlies Blockchain innovation, will firmly influence the level of influence between existing business sector players in human services. It will likewise advance new computerized plans of action and advanced wellbeing activities. Because of the way

that, later on, (information) go-betweens can be kept away from, this innovation opens new entryways concerning how showcase communications in social insurance can be directed. Blockchain in this manner has an enormous potential for the future and will indicate troublesome changes in the social insurance industry.

2.2 Inference From The Survey

Blockchain and Machine Learning Recommending System Now a day's Blockchain and ML, plays a big role in different fields or areas among the Health Care System. This eventually leads to many studies being conducted in selected medicine care facilities. It is im-portant to ensure a technologically afordable, eficient, appropriate, and environmen-tally adaptable and customer friendly system, designed to fully utilize the Blockchain and Machine Learning technology for the greatest benefit in the health care soci-ety. Here computer have great relevance on storing data's securely and ease access on them in brief burst of time. In order to store the consumer information securely, a blockchain is being build. The blockchain is a robust, integrated, secure and trustwor-thy technology.

The blockchain system encrypts the data stored in each layer with the data stored in the next layer, making it almost impossible to crack it without the key which brings us to the next technology in Blockchain that is the public/private key sys-tem. The public/private key of Blockchain works like any other system, that is all the hospitals have the public key of the patient and can use it to enter the details of the patient, and the patient will have his private key which he and only he can use to access his data, hence making it very secure. Healthcare may be a huge application situation of blockchain, and blockchains utilized in aid area unit known as medicine blockchain. In general, blockchain blocks area unit open and therefore the transaction in them area unit public. If some private data is/are involved in these transaction, there is a high chance for them to be leaked. Matured to aid system involving an excellent deal of privacy information, certain preservation instruments should be designed to shield this privacy information in medicinal blockchain.

Moreover, because the essence of security measurments is the public – private key man-agement schemes, the suitable key management schemes ought to be constructed before blockchains may be utilized in aid system. Here, as reported by the features of medicinal blockchain, the authors use a frame detector network to style a light-weight backup and economical comeback theme for keys of medicinal blockchain.

CHAPTER 3

BLOCKCHAIN

With the quick advancement in innovation, new creations continue creating and we should misuse the new advances to do useful for the world. The Blockchain tech-nology centers around a conveyed record, henceforth decentralizing the information. The upside of this is in the event that one of the records is adulterated or hacked, we don't lose any information as rest of the records bolster the uncorrupted information. Blockchain additionally scrambles the following square of the chain with its forerunner, making the information unmodifiable; the star of this is insurance agencies would trust on the patient more as the entirety of his records would be orig-inal and appearing with the insurance agencies. As the Health Record will be open by the patient not at all like the current framework where the medical clinics and the specialists get to the records for the patient and every emergency clinic has a different record, thus giving a typical record of the patient for any emergency clinic.d, we do not lose any data as rest of the ledgers support the uncorrupted data. Blockchain also encrypts the next block of the chain with its predecessor, making the data unmodifiable; the pro of this is that insurance companies would trust on the patient more as all his records would be orig-inal and showing transparency with the insurance companies. As the Health Record will be accessible by the patient unlike the existing system where the hospitals and the doctors access the records for the patient and each hospital has a separate record, hence providing a common record of the patient for any hospital.

3.1 Transaction

An electronic coin is named as a gathering of computerized marks. Every holder, when he needs to exchange the coin to the following individual, carefully signs the prior exchange with the open ey of the later holder, adding all these as far as possible of the coin. The chain of proprietorship can be effectively confirmed by the representative which should be possible by just checking the marks appended in the exchange

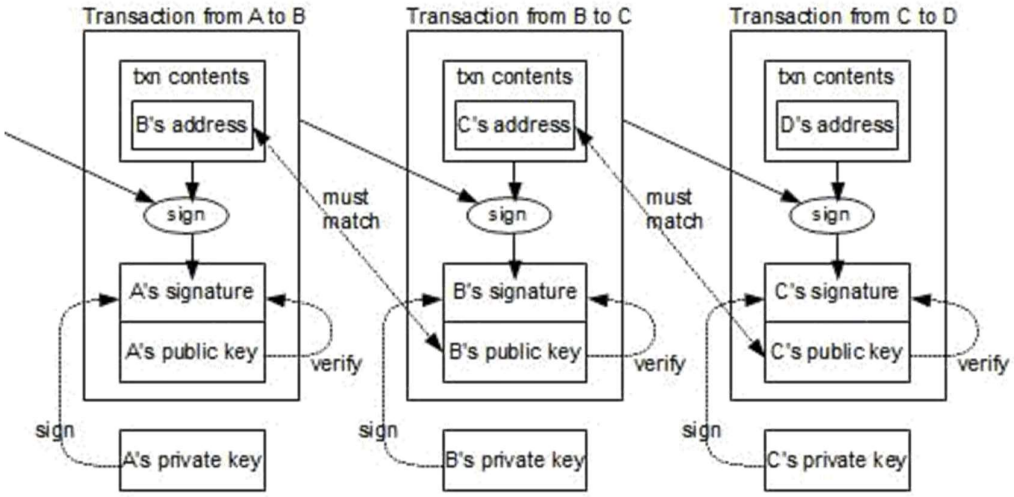

Figure 3.1: Transaction

3.2 Proof Of Work

The verification of work is utilized to make sense of if the square has really being mined or not. It involves an esteem which when hashed with a hashing calculation, the hash starts with a required least number of zeros. The middle work required is the time required to mine a specific square which is the exponentials of the quantity of zeros compulsory and can be confirmed by executing a solitary hash. When the measure of CPU control which should be a great deal has been used to influence it to fulfill the required confirmation of-work, at that point the substance of the square can't be changed without playing out that very work once more. As the squares which arrive later are attached onto it, the work done to change the square would consolidate re-trying all the work (or mining) obstructs it which aggregates to be a great deal.

Evidence of work is additionally in charge of the blockchain to just proceed in the first or unmodified chain. The choice of which chain to choose s made by the longest chain; as that will be the chain with the most extreme verification of work used in it. On the off chance that most of the hubs in the chain are straightforward hubs, the valid chain will become the most quickly and will be certainly quicker than any contending chains. To refresh a past square, a programmer would have to mine the square he is assaulting alongside all squares which are after it and after that connect with and outperform crafted by all the legit hubs considering the way that the legitimate hubs become all around quickly; the likelihood of which is by zero.

3.3 Transparency and Privacy

As Blockchain uses Public-Private key cryptography, each user has his own pair of keys. When a transaction is made, the user can share his public key and make his transaction visible to the person with whom he shared his key. If the user wishes privacy, he can choose to keep his public key anonymous and all the other people will notice that a transaction took place. Hence giving the user full power for his transparency or privacy.

CHAPTER 4

DISEASE PREDICTOR

Our Disease Predictor will take in various amounts of symptoms as input and predict a disease based on the symptoms. It will be based on three machine learning algorithms : 1) Logistic Regression

2) Naive Bayes

3) Decision Tree Classification

4.1 Creating the Data-Set

We had a data-set which mapped each symptom to a disease along with its TF-IDF (Term Frequency âAS¸ Inverse Document Frequency) with 321 symptoms and 4219 diseases. So, in order to get the symptoms for a particular disease, we had to run a for loop on all the symptoms looking which one had the corresponding disease; and once we found out all the symptoms, we made a new dataset which had the symptoms for each disease.

Once we had created our dataset, it was time to convert it into a dataset which the machine could understand. The new dataset had the rows as all the diseases and the columns as all the symptoms. So if a particular disease had a particular symptom, it was marked as 1 and if not, it was marked as 0. In the end we had a dataset of dimensions: 4921 X 321 (diseases X symptoms).

4.2 Logistic Regression

Strategic relapse can be utilized to envision the likelihood of any result which can just have two values(0 or 1, an or b). The expectation depends on the utilization of at least one indicators (they can be numerical or clear cut). A direct regressor isn't convenient

to envision the estimation of a two esteem variable for two reasons: A straight regressor can anticipate values which are outside the acknowledgment extend and subsequently not fitting to use(e.g. anticipating values outside the scope of 0 to 100).

Since the activities can just have one of two conceivable values(0 or 1, an or b) for every task, the residuals and results won't be typically or gausianlally dispersed around the anticipated line.

Then again, a strategic relapse delivers a calculated bend or a sigmoid bend, whose qualities are inside 1 and 0. Calculated relapse is fairly like direct relapse, however the bend is built/made utilizing the common logarithm or the sigmoid of the "chances" of the objective variable, as opposed to the estimation. Besides, the indicators don't should be gausianally dispersed or typically or even have equivalent fluctuations in every single gathering.

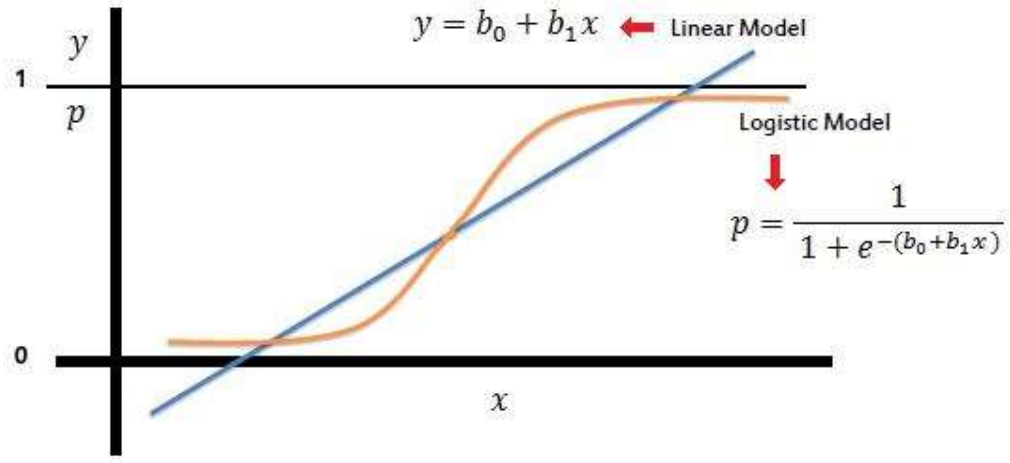

Figure 4.1: Logistic Regression

4.3 Decision Tree

Decision tree is used to create classification or regression models/algorithms in the fashion of a distributed tree structure. It divides a data-set into smaller and even smaller subsets wheras at that very time a correlated decision tree is developed which keeps on increasing as the nodes are developed. The final result is a decision tree with nodes and leaf leaves (decision nodes and nodes). A decision tree node (e.g., Outbook) can have more than two branches (e.g., Left, Right and Top), each representing corresponding values for the respective attribute tested. Leaf node (e.g., Time Played) representing a decision for the respective analytical target. The topmost/root node decision node generally corresponds to the best predictor in the tree is called the root node. Decision trees, generally can handle categorical data easily and in some cases even numerical data is handled easily.

Figure 4.2: Decision Tree

4.3.1 Algorithm

The core-deep design for building decision trees is called ID-3 by R. Quinlan which employs a top to down, greedy search algorithm through the area of possible branches without any kind of back-tracking. The ID-3 design can be efficiently used to construct a decision tree in a very small time for regression by replacing Standard Deviation Reduction with Information Gain.

4.4 Naive Bayes

The Naive Bayes is somewhat of a classifier which utilizes the well known Bayes Theorem and the Naive suspicion. It predicts the investment estimations for each class. The class which has the most noteworthy estimation or plausibility is viewed as the best class or the no doubt class. This is known as Maximum A Posteriori or (MAP). Guide of the speculation is:
MAP (H)
= maximum (P(H|E))
= maximum ((P(E|H) * P(H)) / P(E))
= maximum (P(E|H) * P(H))

P(E) is likelihood of proof, and it is generally used to standardize the outcome. It continues as before in this way, evacuating it doesn't generally influence the yield, yet transforms it somewhat which is irrelevant. Gullible Bayes classifier anticipates that none of the highlights should be similar to one another or one and another. Closeness or nonattendance of one or a couple of highlights does not impact the closeness or nonappearance of some other component.

4.4.1 Example

A vegetable may be treated to be a Beet if it is maroon, spherical, and about 5" in diameter. Considering the fact that these features bank on another or even upon the inclusion of any other feature, the naive Bayes classifier acknowledges all of these attributes to individually devote to the estimation that this particular vegetable is an beet. In real data-sets, we test an assumption where we are given many evidences(features). So, calculations become a little conglomerated. To streamline the work, the feature to individualize approach is used to separate many evidence and treat all of them as an individual one.
P(H|Many features) = P(E-1| H)* P(E-2|H)...*P(E-n|H) * P(H) / P(Many Evidences)

4.5 Support Vector Machine

A Support Vector Machine (SVM) is a classification algorithm performs classification by finding a hyperplane that exaggerate the margins between two leagues. Support Vectors are the vectors that are used to define the hyperplane. SVM is very effective in classifying datasets which have 2 or more categories. It generally takes more tome to train a dataset on an SVM algorithm, but the answer is very crisp and accurate.

4.5.1 Algorithm

1) Determine an excellent hyperplane to axeggerate the margin
2) Use the definition above for dilemmas which are not separable linearly that is have a penalty class for classifications not under any group.
3) Transform data to a higher dimensional space(if the data is not linearly separable) where it is relatively easier to perform classification on the data with linear

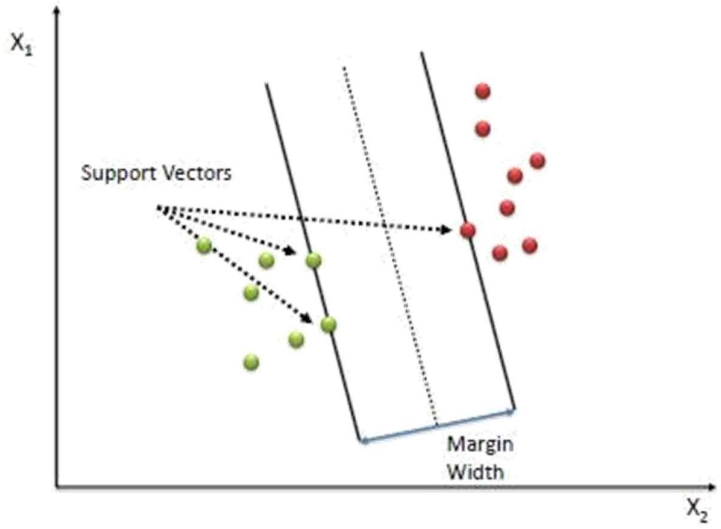

Figure 4.3: SVM

decision surfaces of the svm: redefine the problem so that data now explicitly maps to this space.

In the end, the data is separated linearly with a linear plane. The data can be classi-fied as 1) In the current category or 2) Not in current category.

4.6 Training and Testing Data

The dataset developed in is entirely used as the sole training data for out algorithms. This training data is fit in each of the algorithms separately and then the trained algo-rithm is pickled to save it. Pickling is used so that we do not have to train the algorithm again and again. And can be used directly from the hard disk of my PC which saves a lot of time. Each algorithm has its own pickle. When we want to load the data, we just read the pickle and load tit as our model which can then be used to predict diseases normally. Our testing data in this case was the same as our training data as an algorithm can not predict diseases which it doesn't know already.

4.7 Final Result

Whenever the symptoms were input in our user interface, we first had to convert it into a format which our algorithm could read. So, we used binary just as we did in our dataset. If the symptom entered was in the list, it was entered as 1 and if it was not in the list, it was entered as 0. We had a list of 0s and 1s as a list of length 321 (number of symptoms). Then this data was used to get our predictions for each algorithm.

CHAPTER 5

RECOMMENDER SYSTEM

A recommender System will be utilized to suggest a specialist for the patient who is best appropriate for that specific infection.

A recommender framework or a suggestion framework (in some cases "framework" is re-put with a comparable, for example, "stage" or "motor") is a tract of data permeating frameworks which try to appraise the "rating" or "inclination" a client would provide for a curious perspective. Recommender frameworks for the most part works with a few kinds of general data: Idiosyncratic data: This is the data about things (classes, watchwords, and so forth.) and clients (profiles, inclinations, and so forth.) which exceptionally characterize every client. Client - Item connections: This is data, for example, likes, number of buys, appraisals, and so forth which are exceptionally given by every client. In view of the data above, we touch base at a first sort of prescribing frameworks: content-based, which utilizes quirky data, and the second kind of proposal frameworks: collective separating, which depends on client - thing cooperations. The third type is a blend of both: Hybrid frameworks join the two sorts of learning with the point of turning away entanglements which are produced when working with only one kind.

5.1 Content Based Systems

These procedures make suggestions utilizing a client's thing and profile highlights. They theorize that if a client was keen on a specific thing previously, they will most presumably be keen on it at some point later on. Comparative things are normally gathered dependent on their component - set. Client

profiles are created utilizing authentic between activities or by unequivocally examining clients regarding their excitement. One sort of frameworks use the information from the client's close to home or public activity; they are not viewed as absolutely content - based. One issue which emerges is making apparent suggestions for a client as a result of over the top specialization (client 1 is just intrigued by classes 2, 3, and 4,

also, the framework is unfit to suggest things outside the scope of those classifications, despite the fact that those classifications couldn't be truly fascinating to client 1). Another normal issue which emerges is that new clients come up short on an itemized profile except if they are expressly requested increasingly more data. In any case, it is generally easy to add new things or classifications to the framework. We simply need to guarantee that we appoint them an applicable gathering as indicated by their list of capabilities.

5.2 Collaborative Filtering

These frameworks utilize some unique sorts of strategies to consider the client collaboration with a specific gathering of things. A network can be utilized to effortlessly picture the arrangement of collaborations where every section (i,j)(i, j)(i,j) is utilized to speak to the cooperation between client iii and thing jjj separately. A fascinating way we can pick up understanding on cooperative separating is by not taking a gander at it as a proposal issue, however by thinking of it as a speculation of characterization and relapse. By and large cases we mean to anticipate a variable which is depended straightforwardly on different factors or highlights, though in synergistic sifting, refinements, for example, include factors or class factors don't exist.

5.3 Data-Set

We had successfully created a data set of all the doctors in Delhi. Our dataset comprises of 3000 doctors, their speciality, degrees and their addresses. After transforming the data-set into an excel file, we had used google Application Program Interface (API) to find the latitude and longitude of each doctors address so that it could be plotted on the map and also so that we could find the distance of the doctor from the user. And we had given a random rating (out of 10), and a random Cost per Meeting to each doctor.

5.4 Implementation

After finishing the dataset, we had created a user interface where the user would enter the speciality of the doctor he needed. Then our recommendation algorithm would filter out results as per the user requirements. The user could choose whether to filter each doctor by Rate, Rating or Distance. The resulting Doctors were then displayed on the map along with a table showing the information about each doctor.

CHAPTER 6

ENCRYPTION-DECRYPTION

Rivest-Shamir-Adleman (RSA) is a public - private key cryptosystem. RSA s based

on the factoring problem that is the difficulty of factorizing the product of two very large prime numbers; making it almost impossible to crack. In RSA, two keys are generated that is the public and private key such that neither can be derived from one of them. The private key is kept a secret and no one else knows it but the user, on the other hand, the public key is known to anyone or everyone. During encryption, the public key is used to encrypt the document and the private key to decrypt it. For a Digital Signature, the private key is used to digitally signa document.

In our App, we have two different classes a Patient and a Doctor. The size of the keys of a doctor is 2048 bits and the size of the keys of a Patient is 2048*4 bits. The size of the doctor keys is small because he will be signing data which is relatively small in size whereas the Patient will be encrypting large sized data.

Whenever a patient or a doctor signs in, his public- private keys are generated, his private key is encrypted and both are stored in our database and can be accessed by the user upon login.

6.1 Implementation

The keys when generated, are stored in a database in a string format. Whenever we had to load a particular key set, The User ID was required to decrypt the private key of the user and the public key could be read directly from the database. Whenever a Doctor entered an EHR for a Patient, each block of the EHR was digitally signed by the private key of the user and then it was encrypted by the public key of the user. The encrypted data was then

stored in the database. Whenever we wanted to extract data from the database to show on the user's screen, the data was first decrypted using the private key of the user, then the digital signature of the doctor was verified and if the signature was valid, it was displayed on the screen

CHAPTER 7

CONCLUSION

To conclude, We made a web User Interface (UI) using Flask (Python). The UI had a connection with a database with tables: 1)Patient 2)Doctor

3)Keys
4)EHR
5)Blockchain
Whenever a Doctor or a Patient Signed Up, A new entry was made in the respected table and a pair of keys were generated which were stored in the keys table. Whenever the Doctor created a new Electronic Health Record (EHR), it was stored in the EHR table and a new block in the blockchain was created linked to the previous block. Only the Doctor has access to this page. The Disease Predictor, which used Logistic Regression, Decision Tree, SVM and Naive Bayes was 87 % accurate. The Doctor Recommender System produced excellent results in recommending a doctor which could be sorted by distance, Rate or Rating.

The Problems of having all the records safe, encrypted, unmodifiable, undeletable were succesfully resolved by the blockchain and RSA encryption.

REFERENCES

[1] Rui Guo, Huixian Shi, Qinglan Zhao, and Dong Zhengi, "Secure Attribute-Based Signature Scheme With Multiple Authorities for Blockchain in Electronic Health Records Systems", IEEE Access (Special Section on Research Challenges and Opportunities in Security and Privacy of Blockchain Technologies), pp. 11676-11686, 2018.

[2] Qi Liu, Kenli Li, "Decentration Transaction Method Based on Blockchain Technology", International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), pp. 416-419, 2018.

[3] Mingjun Dai, Shengli Zhang, Hui Wang and Shi Jin, "A Low Storage Room Requirement Framework for Distributed Ledger in Blockchain", IEEE Access, pp. 22970-22975, 2018.

[4] Mertz L, "(Block) Chain Reaction: A Blockchain Revolution Sweeps into Health Care, Offering the Possibility for a Much-Needed Data Solution", IEEE Pulse, vol. 9, no. 3, pp. 4-7, 2018.

[5] Matthias Mettler, "Blockchain technology in healthcare: The revolution starts here", IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), 2016.

[6] Huawei Zhao, Peidong Bai, Yun Peng and Ruzhi Xu, "Efficient key management scheme for health blockchain", CAAI Transactions on Intelligence Technology, vol. 3, no. 2, pp. 114-118, 2018.

[7] Zyskind G, Nathan O, Pentland A.S, "Decentralizing privacy: using blockchain to protect personal data", IEEE Security and Privacy Workshops, San Jose, CA, pp. 180-184, 2015.

[8] Chunxue Wu, Chong Luo, Naixue Xiong, Wei Zhang, Tai-Hoon Kim, "A Greedy Deep Learning Method for Medical Disease Analysis", IEE Access(Special Section on Mobile Multimedia for Healthcare) 2018, pp. 11676-11686, 2018.

[9] S. M. Taheri, Iman Irajian, "DeepMovRS: A Unified framework for deep learning-based movie recommender system", 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS), 2018

[10] Kraff, D, "Difficulty control for blockchain-based consensus systems", Journal of Peer-to-Peer Networking and Applications, vol. 9, no. 2, pp. 397–413, 2016.

APPENDIX - 1

8.1 account.html

```
{% extends "layout.html" %}
{% block content %}

  <div class="content-section" style="background: #2F4F4F;color: #FFFACD">
    <div class="media">
      <img class="rounded-circle account-img" src="{{ image_file }}">
      <div class="media-body">
        <h2 class="account-heading" style="color: #FFFACD">{{ current_user.username }}</h2>
        <p class="text-secondary" style="color: #FFFACD">{{ current_user.email }}</p>
      </div>
    </div>
      <form method="POST" action="" enctype="multipart/form-data">
          {{ form.hidden_tag() }}
          <fieldset class="form-group">
              <legend class="border-bottom mb-4">Account Info</legend>
              <div class="form-group">
                  {{ form.username.label(class="form-control-label") }}

                  {% if form.username.errors %}
                      {{ form.username(class="form-control form-control-lg is-invalid") }}
                      <div class="invalid-feedback">
                          {% for error in form.username.errors %}
                              <span>{{ error }}</span>
                          {% endfor %}
                      </div>
                  {% else %}
                      {{ form.username(class="form-control form-control-lg") }}
                  {% endif %}
```

```
            </div>
            <div class="form-group">
                {{ form.email.label(class="form-control-label") }}
                {% if form.email.errors %}
                    {{ form.email(class="form-control form-control-lg is-invalid") }}
                    <div class="invalid-feedback">
                        {% for error in form.email.errors %}
                            <span>{{ error }}</span>
                        {% endfor %}
                    </div>
                {% else %}
                    {{ form.email(class="form-control form-control-lg") }}
                {% endif %}
            </div>
            <div class="form-group">
                {{ form.picture.label() }}
                {{ form.picture(class="form-control-file") }}
                {% if form.picture.errors %}
                    {% for error in form.picture.errors %}
                        <span class="text-danger">{{ error }}</span></br>
                    {% endfor %}
                {% endif %}
            </div>
        </fieldset>
        <div class="form-group">
            {{ form.submit(class="btn btn-outline-info") }}
        </div>
    </form>
</div>
{% endblock content %}
```

8.2 create_post.html

```
{% extends "doc_layout.html" %}
{% block content %}
<div class="content-section" style="background: #2F4F4F;color: #FFFACD">
```

```
<form method="POST" action="">
    {{ form.hidden_tag() }}
    <fieldset class="form-group">
        <legend class="border-bottom mb-4">{{ legend }}</legend>
        <div class="form-group">
            {{ form.userid.label(class="form-control-label") }}
            {% if form.userid.errors %}
                {{ form.userid(class="form-control form-control-lg is-invalid") }}
                <div class="invalid-feedback">
                    {% for error in form.userid.errors %}
                        <span>{{ error }}</span>
                    {% endfor %}
                </div>
            {% else %}
                {{ form.userid(class="form-control form-control-lg") }}
            {% endif %}
        </div>
        <div class="form-group">
            {% for i in range(7) %}
                {{ form.test_or_med[i].label(class="form-control-label") }}


                {{ form.test_or_med[i](class="form-control form-control-lg") }}

                {{ form.causes[i].label(class="form-control-label") }}


                {{ form.causes[i](class="form-control form-control-lg") }}
            {% endfor %}
        </div>
        <div>
            {% for disease in form.diseases %}
                {{ disease.label(class="form-control-label") }}


                {{ disease(class="form-control form-control-lg") }}
```

```
            {% endfor %}
          </div>
        </fieldset>
        <div class="form-group">
          {{ form.submit(class="btn btn-outline-info") }}
        </div>
    </form>
</div>
{% endblock content %}
```

8.3 layout.html

```
<!DOCTYPE html>
<html>
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">

    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='main.css') }}">

    {% if title %}
        <title>eClinic - {{ title }}</title>
    {% else %}
        <title>eClinic</title>
    {% endif %}
  <style>
    #map{
```

```
    height:400px;
    width:100%;
   }
  </style>
</head>
<body class="text-steel" style="background: #FFFACD">
   <header class="site-header">
    <nav class="navbar navbar-expand-md fixed-top" style="background:   #2F4F4F">
      <div class="container">
        <a class="navbar-brand mr-4" href="{{ url_for('main.home') }}"><span style="color:
red">e+</span><span style="color: white">Clinic</span></a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarToggle" aria-controls="navbarToggle" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarToggle">
          <div class="navbar-nav mr-auto">
            <a class="nav-item nav-link" href="{{ url_for('main.home') }}" style="color:
#FFFACD">Home</a>
            <a class="nav-item nav-link" href="{{ url_for('main.about') }}" style="color:
#FFFACD">About</a>
          </div>
          <!-- Navbar Right Side -->
          <div class="navbar-nav">
            {% if current_user.is_authenticated %}
              <a class="nav-item nav-link" href="{{ url_for('predictor.disease_predictor') }}"
style="color: #FFFACD">Disease Predictor</a>
              <a class="nav-item nav-link" href="{{ url_for('recommend.doctor_recommender') }}"
style="color: #FFFACD">Doctor Recommender</a>
              <a class="nav-item nav-link" href="{{ url_for('users.user_posts') }}" style="color:
#FFFACD">My EHRs</a>
              <a class="nav-item nav-link" href="{{ url_for('users.account') }}" style="color:
#FFFACD">Account</a>
              <a class="nav-item nav-link" href="{{ url_for('users.logout') }}" style="color:
#FFFACD">Logout</a>
```

```
            {% else %}
              <a class="nav-item nav-link" href="{{ url_for('bchain.show_chain') }}" style="color:
#FFFACD">BlockChain</a>
              <a class="nav-item nav-link" href="{{ url_for('doc_main.home') }}" style="color:
#FFFACD">Switch To Doctor's Page</a>
              <a class="nav-item nav-link" href="{{ url_for('users.login') }}" style="color:
#FFFACD">Login</a>
              <a class="nav-item nav-link" href="{{ url_for('users.register') }}" style="color:
#FFFACD">Register</a>
            {% endif %}
          </div>
        </div>
      </div>
    </nav>
  </header>
  <main role="main" class="container">
    <div class="row">
      <div class="col-md-8">
        {% with messages = get_flashed_messages(with_categories=true) %}
          {% if messages %}
            {% for category, message in messages %}
              <div class="alert alert-{{ category }}">
                {{ message }}
              </div>
            {% endfor %}
          {% endif %}
        {% endwith %}
        {% block content %}{% endblock %}
        <div class="jumbotron jumbotron-fluid" style="background: #FFFACD">
          <h2 class="text-center">WELCOME TO THE FUTURE</h2>
          <h1 class="text-center" style="text-shadow: 3px 2px grey"><span style="color: red;font-
size: 60px">e</span><span style="color: red;font-size: 50px">+</span><span style="color:
#2F4F4F">CLINIC </span></h1>
        </div>
      </div>
      {% if current_user.is_authenticated %}
```

```
        <div class="col-md-4">
          <div class="content-section" style="background: #2F4F4F">
            <h3 style="color: #FFFACD">Currently Logged In as A Patient</h3>
            <p class='text-muted'>Your General Information.
              <ul class="list-group">
                <li class="list-group-item" style="background: #FFFACD">Name: {{
current_user.name }}</li>
                <li class="list-group-item" style="background: #FFFACD">Username: {{
current_user.username }}</li>
                <li class="list-group-item" style="background: #FFFACD">Email: {{
current_user.email }}</li>
              </ul>
            </p>
          </div>
        </div>
      {% else %}
        <div class="col-md-4">
          <div class="content-section" style="background: #2F4F4F">
            <h3 style="color: #FFFACD">Currently not Logged In</h3>
            <p class='text-muted'>Login to get all the benefits of our App.
              <ul class="list-group">
                <li class="list-group-item" style="background: #FFFACD">Saving EHRs on a
BlockChain</li>
                <li class="list-group-item" style="background: #FFFACD">Predicting Diseases</li>
                <li class="list-group-item" style="background: #FFFACD">High Level Security</li>
                <li class="list-group-item" style="background: #FFFACD">Doctor
Reccomendations</li>
              </ul>
            </p>
          </div>
        </div>
      {% endif %}
    </div>
  </main>
```

```
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
</body>
</html>
```

DISEASE PREDICTOR

8.4 forms.py

```
from flask_wtf import FlaskForm
from flask_wtf.file import FileField, FileAllowed
from wtforms import StringField, PasswordField, SubmitField, BooleanField, FieldList,
SelectField
from wtforms.validators import DataRequired, Length, Email, EqualTo, ValidationError
from rabadiom.disease_predictor.utils import unique_symptoms


class SymptomForm(FlaskForm):
    symptom1 = SelectField('Symptom - 1', choices=unique_symptoms)
    symptom2 = SelectField('Symptom - 2', choices=unique_symptoms)
    symptom3 = SelectField('Symptom - 3', choices=unique_symptoms)
    symptom4 = SelectField('Symptom - 4', choices=unique_symptoms)
    symptom5 = SelectField('Symptom - 5', choices=unique_symptoms)
```

```
    symptom6 = SelectField('Symptom - 6', choices=unique_symptoms)
    submit = SubmitField('Submit')

    def validate_symptom1(self, idd):
      if idd.data == None:
        raise ValidationError("First Symptom can not be Empty")

class SymptomFinalForm(FlaskForm):
  #symptoms = StringField[]
  #symptom = StringField('Symptom')
  #symptoms.append(symptom)

  symptom = FieldList(StringField('Symptom'), min_entries=6, max_entries=6)
  add_symptom = SubmitField("Add another Symptom")
  submit = SubmitField("Submit")
```

8.5 Routes.py

```
from flask import render_template, Blueprint, flash, request
from rabadiom.disease_predictor.forms import SymptomForm, SymptomFinalForm
from flask_login import login_user, current_user, logout_user, login_required
from rabadiom import login_manager
from rabadiom.disease_predictor.utils import predict_disease, predict_using_ml
from wtforms import StringField
from wtforms.validators import DataRequired
from functools import wraps
import math


def login_required(role):
    def wrapper(fn):
        @wraps(fn)
```

```
        def decorated_view(*args, **kwargs):
            if not current_user.is_authenticated:
              return login_manager.unauthorized()
            if (current_user.role != role):
                return login_manager.unauthorized()
            return fn(*args, **kwargs)
        return decorated_view
    return wrapper
predictor = Blueprint('predictor', __name__)
@predictor.route("/Disease Predictor", methods=['GET', 'POST'])
@login_required("User")
def disease_predictor():
        form = SymptomForm()
        if form.validate_on_submit():
                if form.symptom1.data == "Select":
                        return render_template('show_diseases.html', title='Disease Predictor
Result')
                symptoms = [form.symptom1.data, form.symptom2.data,
form.symptom3.data, form.symptom4.data, form.symptom5.data, form.symptom6.data]
                diseases = predict_using_ml(symptoms)
                #diseases = ',\r\r\n'.join(["disease :" + d[0] + " \nscore :" + str(math.ceil(d[1]))
for d in diseases])
                #flash(diseases,"success")
                return render_template('show_diseases.html', title='Disease Predictor Result',
diseases=diseases)
        elif request.method == "GET":
                form.symptom1.data = "None"
                form.symptom2.data = "None"
                form.symptom3.data = "None"
                form.symptom4.data = "None"
                form.symptom5.data = "None"
                form.symptom6.data = "None"
        return render_template('disease_predictor.html', title='Disease Predictor', form=form)
```

8.6 utils.py

```
from flask import render_template, Blueprint, flash, request
from rabadiom.disease_predictor.forms import SymptomForm, SymptomFinalForm
from flask_login import login_user, current_user, logout_user, login_required
from rabadiom import login_manager
from rabadiom.disease_predictor.utils import predict_disease, predict_using_ml
from wtforms import StringField
from wtforms.validators import DataRequired
from functools import wraps
import math
def login_required(role):
    def wrapper(fn):
        @wraps(fn)
        def decorated_view(*args, **kwargs):
            if not current_user.is_authenticated:
              return login_manager.unauthorized()
            if (current_user.role != role):
                return login_manager.unauthorized()
            return fn(*args, **kwargs)
        return decorated_view
    return wrapper



predictor = Blueprint('predictor', __name__)

@predictor.route("/Disease Predictor", methods=['GET', 'POST'])
@login_required("User")
def disease_predictor():
        form = SymptomForm()
        if form.validate_on_submit():
                if form.symptom1.data == "Select":
                        return render_template('show_diseases.html', title='Disease Predictor
Result')
```

```
                symptoms = [form.symptom1.data, form.symptom2.data,
form.symptom3.data, form.symptom4.data, form.symptom5.data, form.symptom6.data]
                diseases = predict_using_ml(symptoms)
                #diseases = ',\r\r\n'.join(["disease :" + d[0] + " \nscore :" + str(math.ceil(d[1]))
for d in diseases])
                #flash(diseases,"success")
                return render_template('show_diseases.html', title='Disease Predictor Result',
diseases=diseases)
        elif request.method == "GET":
                form.symptom1.data = "None"
                form.symptom2.data = "None"
                form.symptom3.data = "None"
                form.symptom4.data = "None"
                form.symptom5.data = "None"
                form.symptom6.data = "None"
        return render_template('disease_predictor.html', title='Disease Predictor', form=form)
```

POSTS

8.7 Routes.py

```
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField, TextAreaField, FieldList
from wtforms.validators import DataRequired, ValidationError
from rabadiom.models import User




class PostForm(FlaskForm):

    userid = StringField('User ID', validators=[DataRequired()])
    diseases = FieldList(StringField('Probable Disease'), min_entries=3, max_entries=3)
    test_or_med = FieldList(StringField('Suggested Medicine or Test'), min_entries=7,
max_entries=7)
    causes = FieldList(StringField('Why?'), min_entries=7, max_entries=7)
```

```
submit = SubmitField('Post')



def validate_userid(self, userid):
    user = User.query.filter_by(username=userid.data).first()
    if not user:
        raise ValidationError('No Such User Exists')
    elif user.role == "Doctor":
        raise ValidationError('The ID Provided is the ID of a Doctor')
```

APPENDIX-2

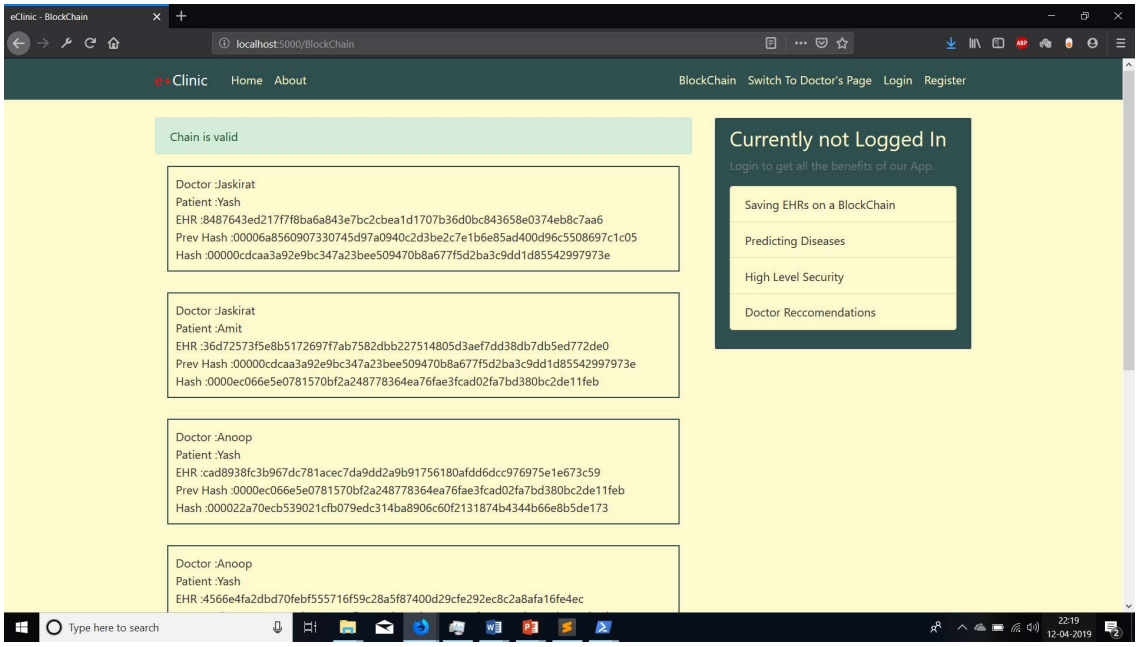

Fig a2.1 - Blockchain

- This is one of the patient pages
- This is the page to display the Blockchain:
- As we can see, on the top it informs us if the BlockChain is valid or not and it is valid in this case.
- We can access this page only if we are logged out as a patient or as a doctor.
- In the navigation bar, we can see options to switch to doctor's page, login(as a patient) and register(as a patient).

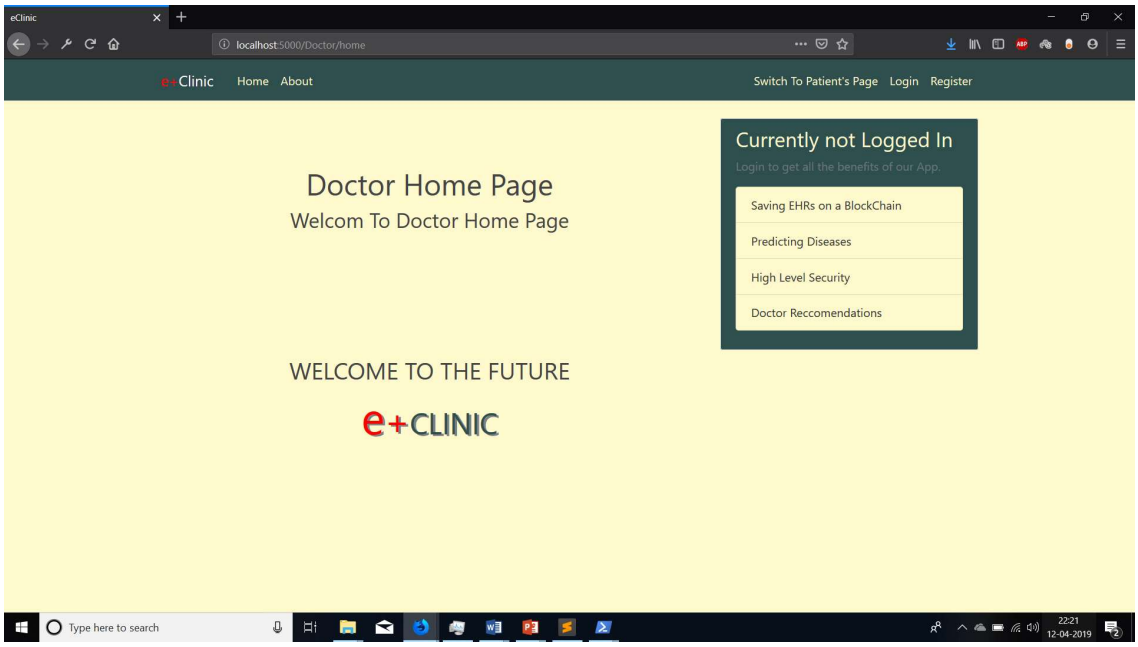

Fig a2.2 – Doctor Home

- This is one of the doctor pages.
- This is the doctor home page.
- On the navbar, we can see that we have options to go to the home(doctor home) page, about(doctor about) page, Switch to patient's page, Login(doctor login), Register(doctor register)
- The side bar tells us information like there is currently no one logged in and other benefits of our website.

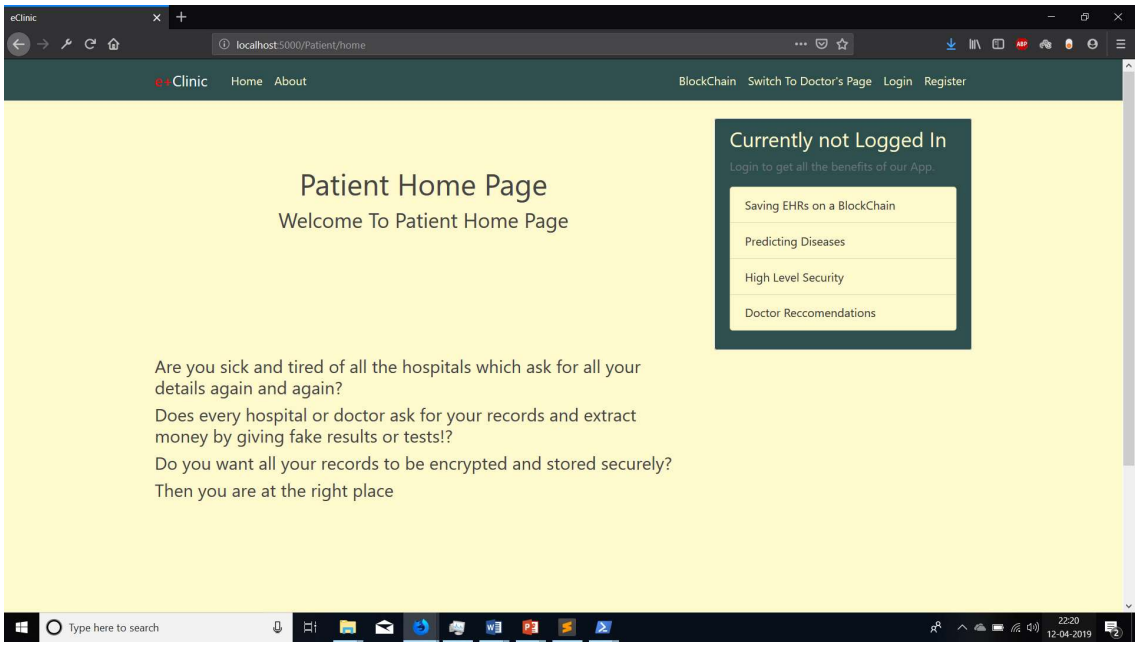

Fig a2.3 – Patient Home

- This is one of the patient pages.
- This is the patient home page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Switch to patient's page, Login(patient login), Register(patient register)
- The side bar tells us information like there is currently no one logged in and other benefits of our website.

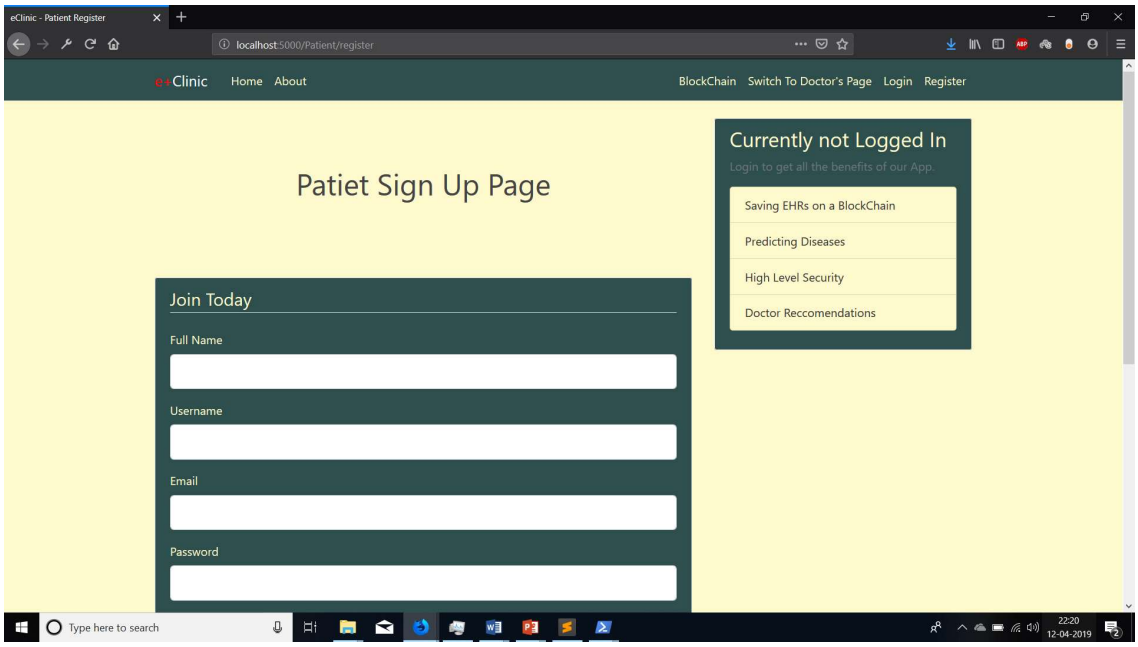

Fig a2.4 – Patient Register

- This is one of the patient pages.
- This is the patient register page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Switch to patient's page, Login(patient login), Register(patient register)
- The side bar tells us information like there is currently no one logged in and other benefits of our website.
- On the bottom, we have option to sign up as a doctor in case a doctor is trying to use this page.
- We also have a link to go to the login page in case the patient is already registered and would like to log in.

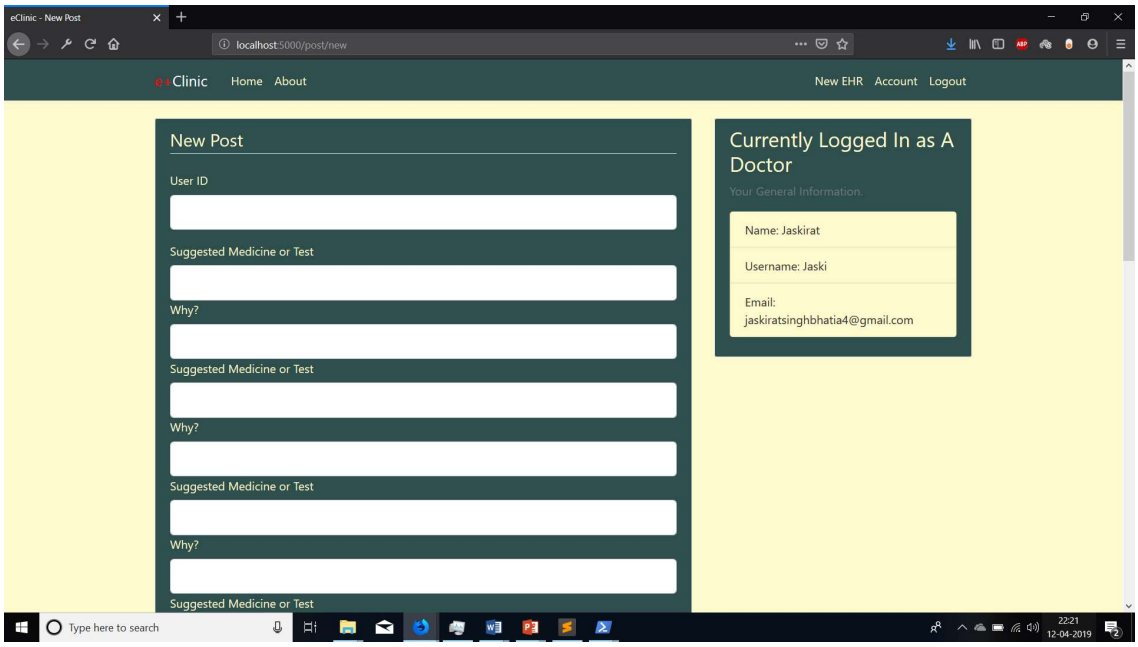

Fig a2.5 – New EHR

- This is one of the doctor pages.
- Only a doctor can access this page
- This is the doctor new EHR page.
- On the navbar, we can see that we have options to go to the home(doctor home) page, about(doctor about) page, Switch to patient's page, Login(doctor login), Register(doctor register)
- The side bar tells us information like there is currently a doctor named Jaskirat logged in and his other details.
- The doctor fills in this form and then submits it to enter a new entry for the patient

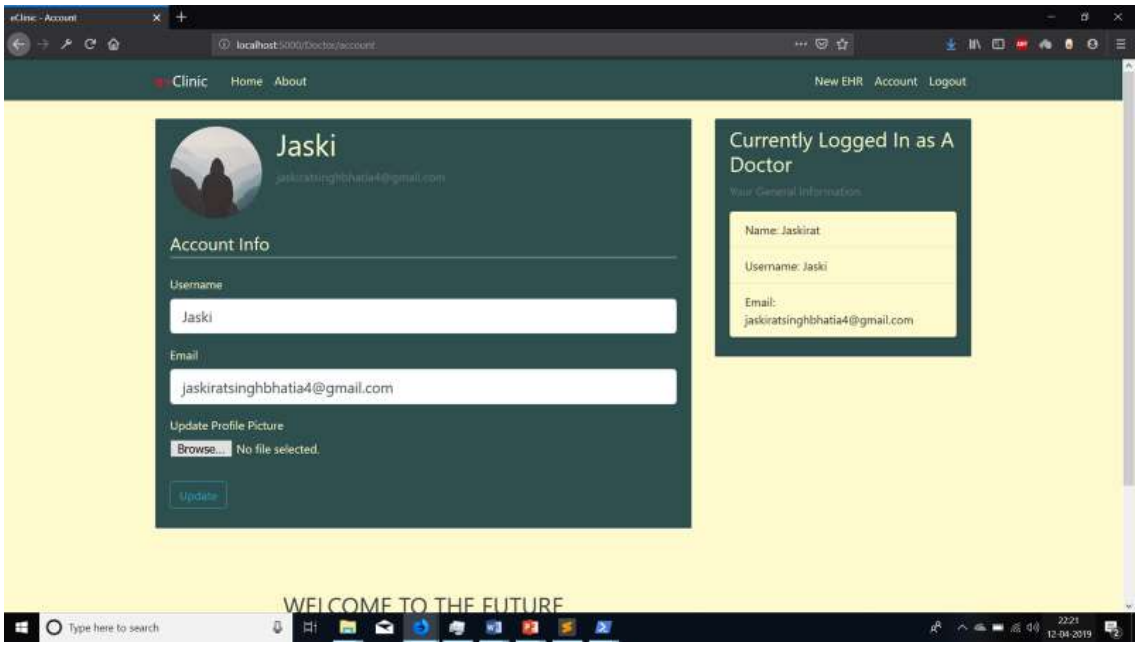

Fig a2.6 - Account

- This is one of the doctor pages.
- Only a doctor can access this page
- This is the doctor Account page.
- On the navbar, we can see that we have options to go to the home(doctor home) page, about(doctor about) page, Switch to patient's page, New EHR(doctor), Logout(doctor logout)
- The side bar tells us information like there is currently a doctor named Jaskirat logged in and his other details.
- The doctor can use this form to update his personal information.

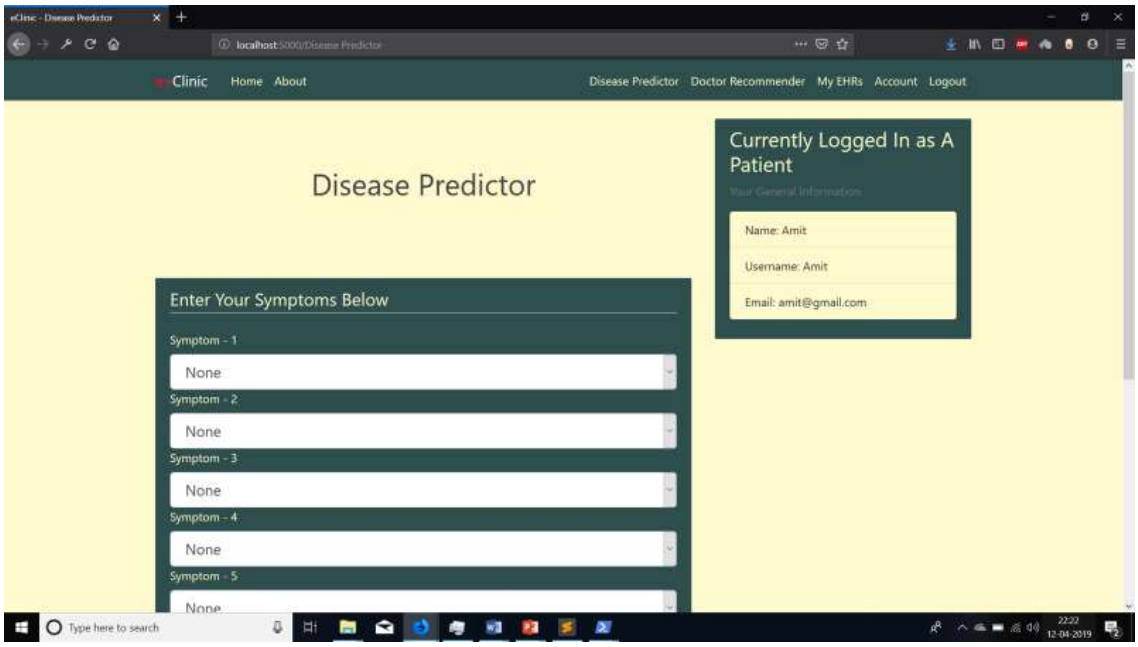

Fig a2.7 – Disease Predictor

- This is one of the patient pages.
- Only a patient can access this page
- This is the patient Disease Predictor page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Doctor Recommender Page, Disease Predictor page, Logout(doctor login)
- The side bar tells us information like there is currently a patient named Amit is logged in and his other details.
- The patient fills in this form and then submits it to go to the diseases page where all the possible diseases are present.

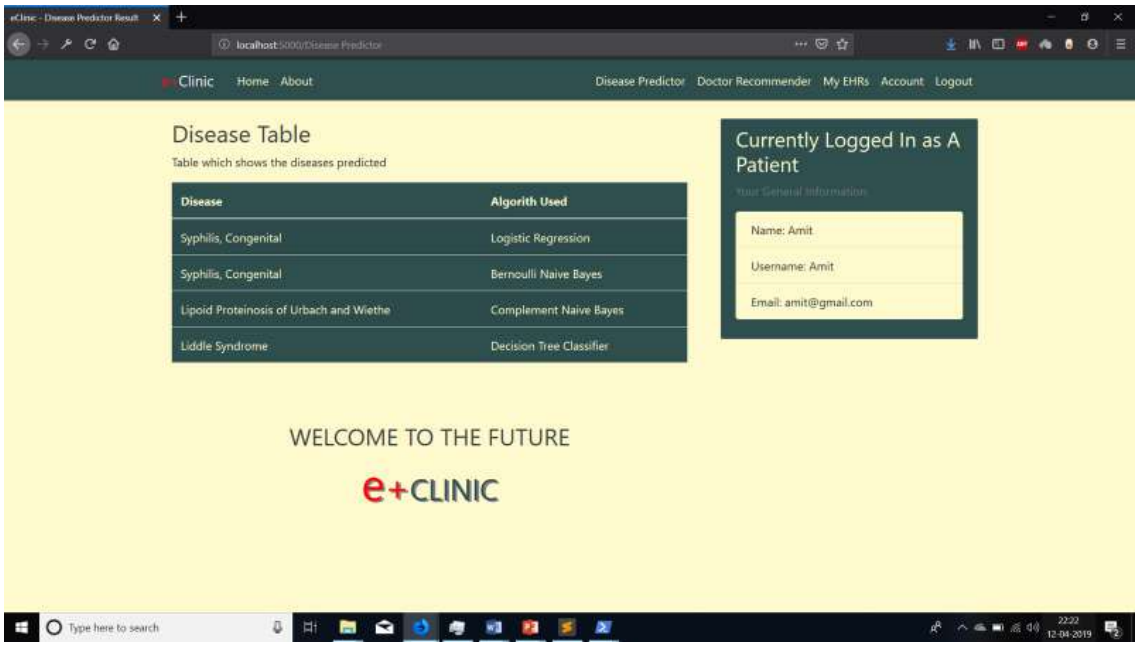

Fig a2.8 – Show Diseases

- This is one of the patient pages.
- Only a patient can access this page
- This is the patient show diseases page.
- On the navbar, we can see that we have options to go to the home(patent home) page, about(patient about) page, Doctor Recommender Page, Disease Predictor page, Logout(doctor login)
- The side bar tells us information like there is currently a patient named Amit is logged in and his other details.
- This page shows all the diseases which are possible which are predicted using the symptoms entered before.

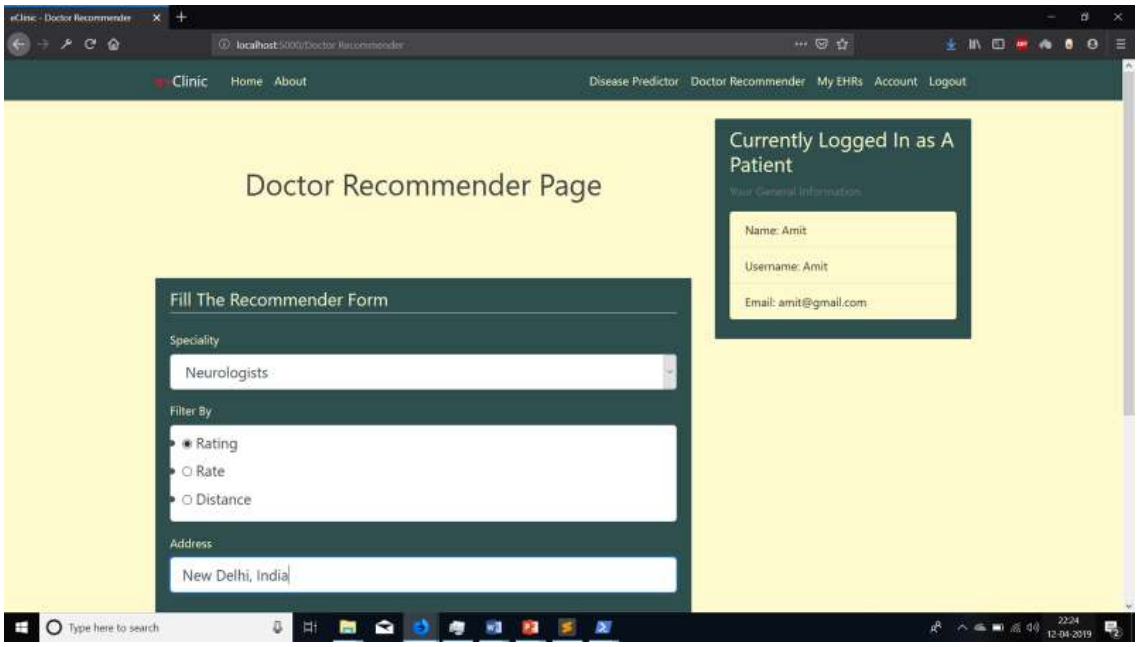

Fig a2.9 – Doctor Recommedner

- This is one of the patient pages.
- Only a patient can access this page
- This is the patient Doctor Recommender page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Doctor Recommender Page, Disease Predictor page, Logout(doctor login)
- The side bar tells us information like there is currently a patient named Amit is logged in and his other details.
- The patient fills in this form and then submits it to go to the show doctors where all the recommended doctors are present.

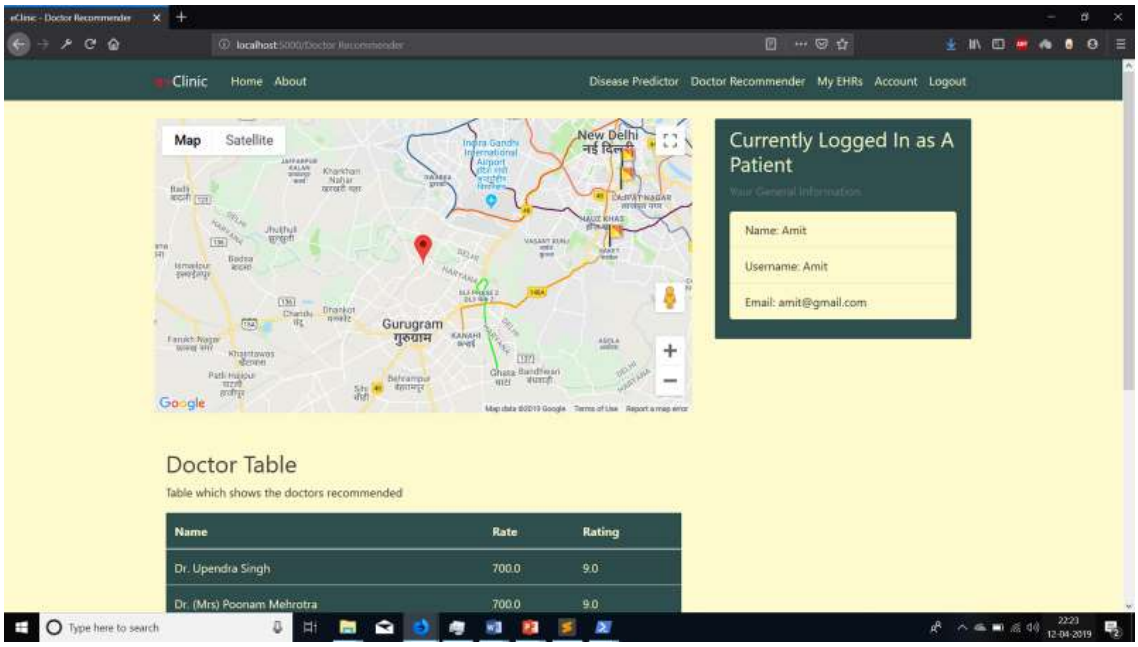

Fig a2.10 – Show Doctors

- This is one of the patient pages.
- Only a patient can access this page
- This is the patient show doctors page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Doctor Recommender Page, Disease Predictor page, Logout(doctor login)
- The side bar tells us information like there is currently a patient named Amit is logged in and his other details.
- This is the result of the form which we submitted in the doctor recommender page and all the recommended doctors are listed along with a few more details.

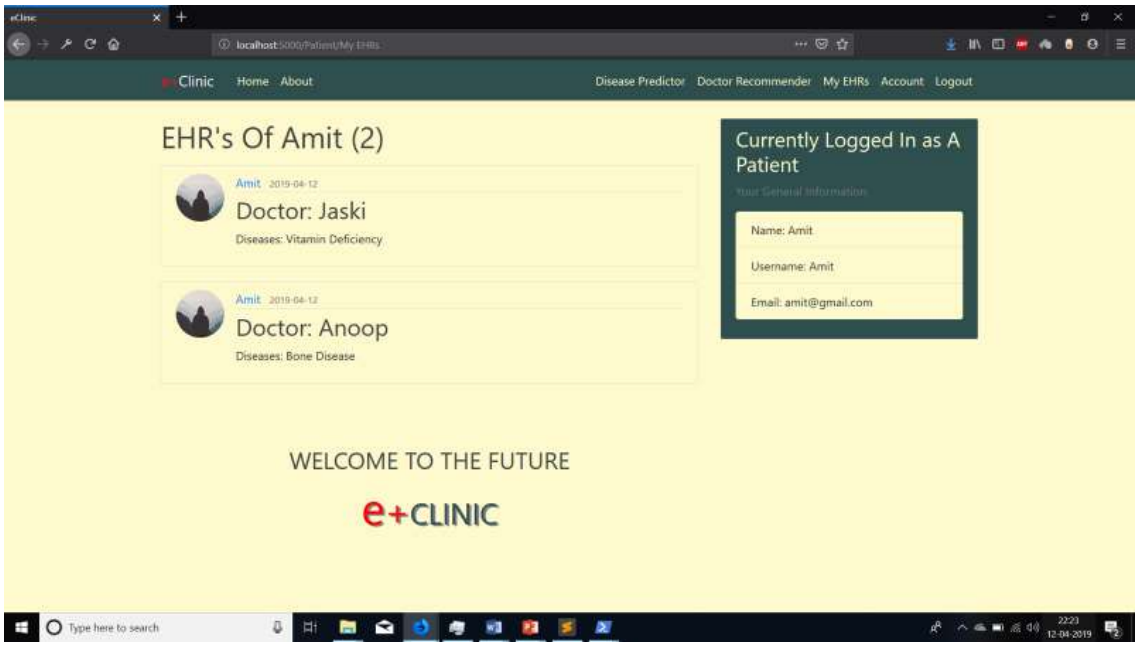

Fig a2.11 – My EHR's

- This is one of the patient pages.
- Only a patient can access this page
- This is the patient My EHR's page.
- On the navbar, we can see that we have options to go to the home(patient home) page, about(patient about) page, Doctor Recommender Page, Disease Predictor page, Logout(doctor login)
- The side bar tells us information like there is currently a doctor named Jaskirat logged in and his other details.
- This page shows us all the EHR's which were entered by the various doctor for patient Amit.

SUBMISSION/ ACCEPTANCE OF PAPER

Title: Blockchain based Electronic Health Record Maintenance System

Authors: Mr. Jaskirat Singh Bhatia*, Mr. Anoop Singh Labana, Dr. M. Prakash

*Corresponding author

Accepted and Submitted to: International Conference on Clinical4.0, Healthcare4.0, Medical4.0 & Pharma4.0 (IC2HMP'2019)