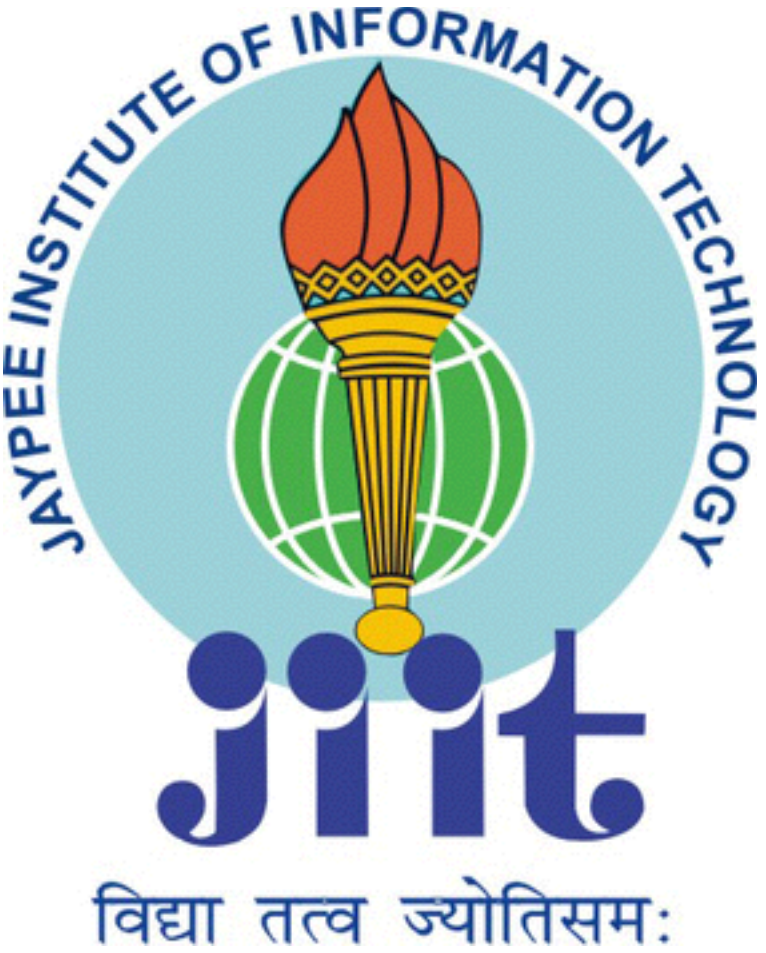


MEDEASE - *Healthcare Made Easy*

Minor Project

Nishant Srivastava - 14103033
Stuti - 14103056
Kanika Gupta - 14103271
Batch - B10

ABSTRACT

MEDEASE- Healthcare made Easy is a one stop solution to your health related queries and concerns. With practical and necessitous features like Symptom Checker, Doctor Recommendation, Hospital Locator and Medical History, it encompasses the needs of most people, the ones suffering with serious ailments to people who are not adept enough to physically visit a physician for a casual check-up or diagnosis.

Medease is a tool that can very much revolutionize how patients interact with their doctors and can bring about a massive transition in the healthcare industry.

OBJECTIVES

To develop an automated system which can do the following:

- Determine what disease a user is suffering from on the basis of what symptoms he/she is showing.
- Suggest doctors for consultation after the diagnosis of the disease.
- Display a route map to the doctor's practice.
- Record keeping of patient's medical/personal information.

SOFTWARE REQUIREMENTS

1. Python 2.7.12:

   The programming language will be mainly used for the numerous frameworks that are available for making websites.

2. PyCharm 2016.2.3:

   PyCharm is an Integrated Development Environment (IDE) used in computer programming, specifically for the Python language.

3. Django:

   Django is a free and open-source web framework, written in Python, which follows the model–view–controller (MVC) architectural pattern.

4. Pyrenn:

   A recurrent neural network toolbox for Python.

5. Numpy:

   NumPy is the fundamental package for scientific computing with Python.

6. Pandas:

   pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. We used pandas to read our dataset from the excel sheet to provide input and output data to our neural network.

7. Openpyxl:

   Openpyxl is a Python library for reading and writing Excel 2010 xlsx/xlsm/ xltx/xltm files.

8. Database (SQLite):

   To maintain the data required for the website in an organised tabular form.

9. Text editor (Textmate, Notepad++)
10. Google Chrome (Inspector)
11. HTML 5, CSS3

COMPONENTS OF MEDEASE

Medease is primarily divided into the following components :-

- **Symptom Checker:** An aid to people suffering from a disease in finding out what it is sitting at their home. The user is required to select a list of symptoms he is suffering from, from the given options, cough, headache, nausea for instance. After the user has selected all the symptoms, he/she is presented with a list of **possible diseases**.

- **Recommended Doctors:** After the user selects a disease, a list of certified doctors who can be consulted for the same on the basis of the diagnosed disease**.**

- **Hospital Locator:** The user will be displayed with a list of nearby hospitals and clinics along with a route map of the same that he can visit for treatment.

- **Medical History:** The user can create his own **profile** and enter all his/her **records** - the conditions he/she's suffering from, blood group, allergies, undergoing treatments and any other relevant information all at one place, saving the burden of carrying loads of paperwork every time he/she visits the doctor.

SYMPTOM CHECKER

THE IDEA:

Many a times we're suffering from a disease and we are hesitant in going to a doctor thinking of it to be a minor cold/cough and subject ourselves to 'self treatments', being ignorant to the fact that it can might as well be something very serious just to save the huge consulting fees and the pain of traveling.

MEDEASE provides it's users with an AI based symptom checker which suggests the user with a list of diseases he might be undergoing. Judging the possibilities, the user can than decide whether his condition requires visiting a doctor or is something too trivial, like a seasonal flu which can be treated at home.

BACKGROUND STUDY AND FINDINGS:

In order to build an accurate Symptom Checker we explored the field of Artificial Intelligence. Artificial intelligence is the branch of computer science concerned with making computers behave like humans and so to make our computer act like a doctor, AI seemed the best choice.

Artificial Learning is classified into two -

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

We, in our project, have implemented Supervised Learning.
Supervised learning is a data mining task of inferring a function from labeled training data.The training data consists of a set of training

examples. In supervised learning, each example is a pair consisting of an input object and the desired output value.
We have primarily used this type of Learning in our project.

Supervised Learning -

- Analytical learning
- Artificial neural network
- Backpropagation
- Bayesian networks
- Decision tree learning
- Learning Automata
- Nearest Neighbor Algorithm
- Support vector machines

We have implemented **Artificial Neural Network** in order to build our Symptom Checker .

Artificial Neural Networks are a computational approach which is based on a large collection of neural units loosely modeling the way a biological brain solves problems with large clusters of biological neurons connected by axons. Each neural unit is connected with many others, and links can be enforcing or inhibitory in their effect on the activation state of connected neural units. Each individual neural unit may have a summation function which combines the values of all its inputs together. There may be a threshold function or limiting function on each connection and on the unit itself such that it must surpass it before it can propagate to other neurons. These systems are self-learning and trained rather than explicitly programmed and excel in areas where the solution or feature detection is difficult to express in a traditional computer program.

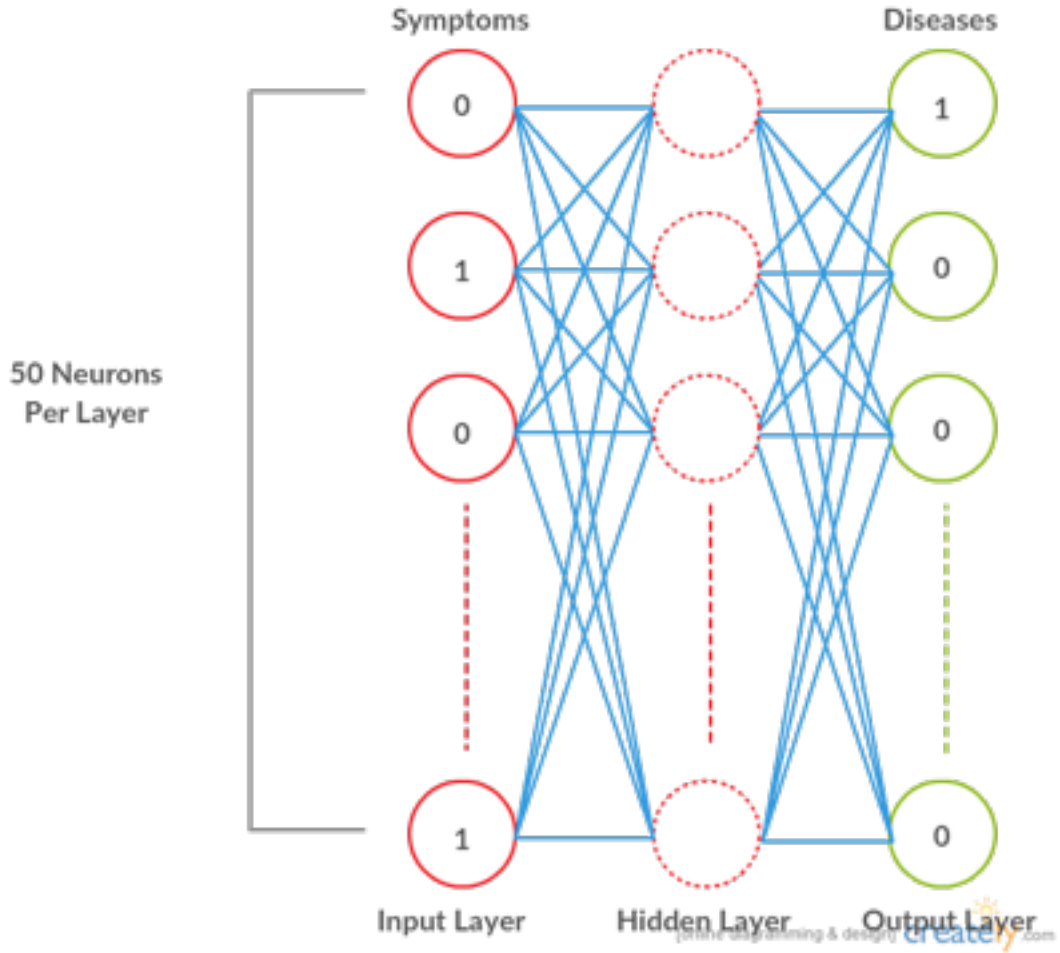
Artificial Neural Network -

- Feedforward neural network.
- Radial basis function (RBF) network.
- Kohonen self-organizing network.
- Learning vector quantization.
- Recurrent neural network.
- Modular neural networks.
- Physical neural network.

Our network is a **Feedforward neural network** - A feedforward neural network is an artificial neural network wherein connections between the units do not form a cycle. As such, it is different from recurrent neural networks. The feedforward neural network was the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.

OUR IMPLEMENTATION

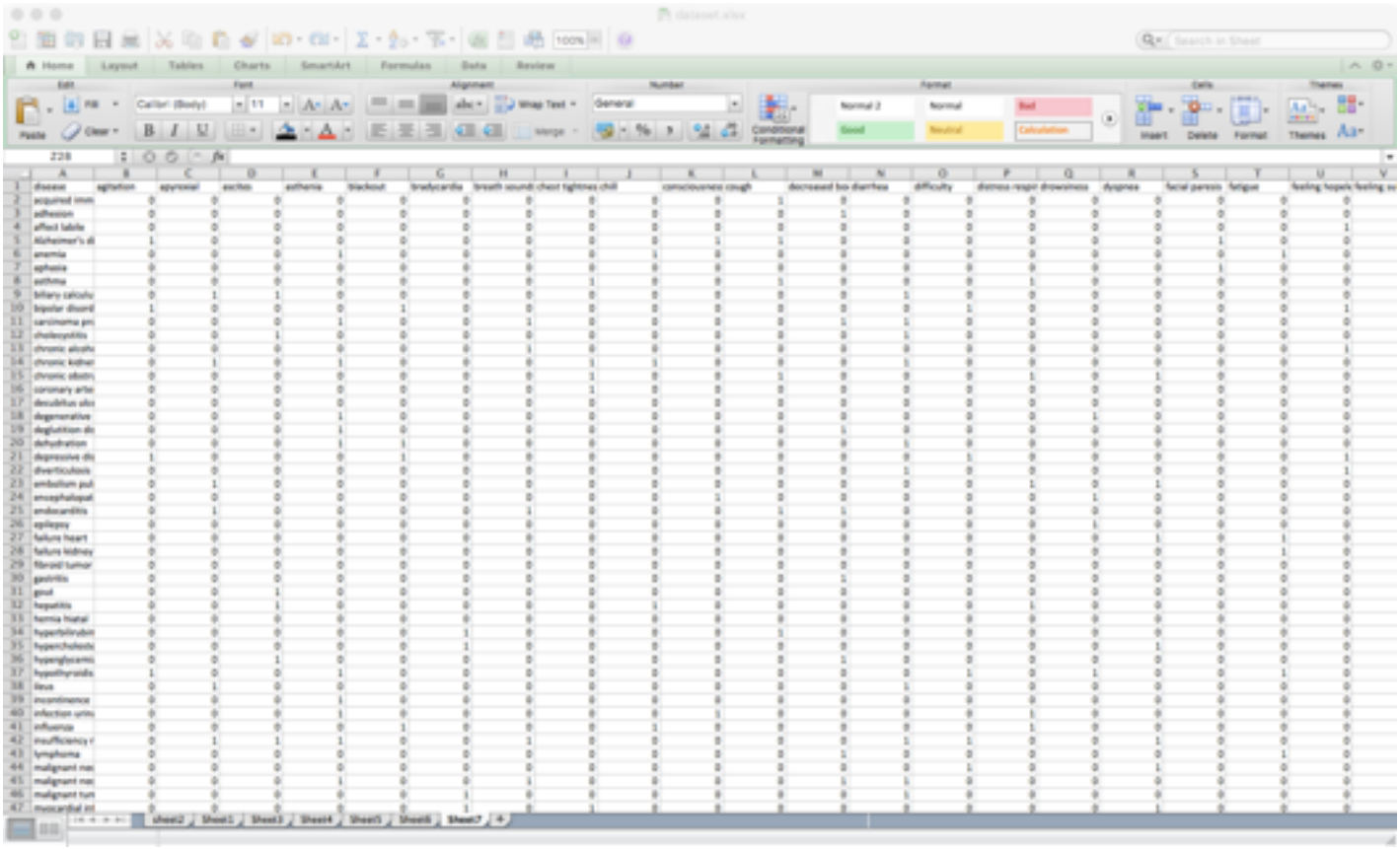
In our Neural Network we have 1 Input, Hidden and Output Layer each which we implemented using **pyrenn**. Each layer consists of 50 neurons. The 50 neurons in the input layer correspond to the symptoms which we are taking as Input from our user. The 50 neurons in the output layer correspond to 50 diseases in our dataset by which the symptom input from our user will be classified in. Following is a pictorial representation of what our network looks like:



For every symptom selected by our user, the value corresponding to that symptom in our dataset will be changed to 1 and the rest will remain 0. In the output layer the neurons with value 1 will correspond to the diseases that the symptom checker suggests the user might have.

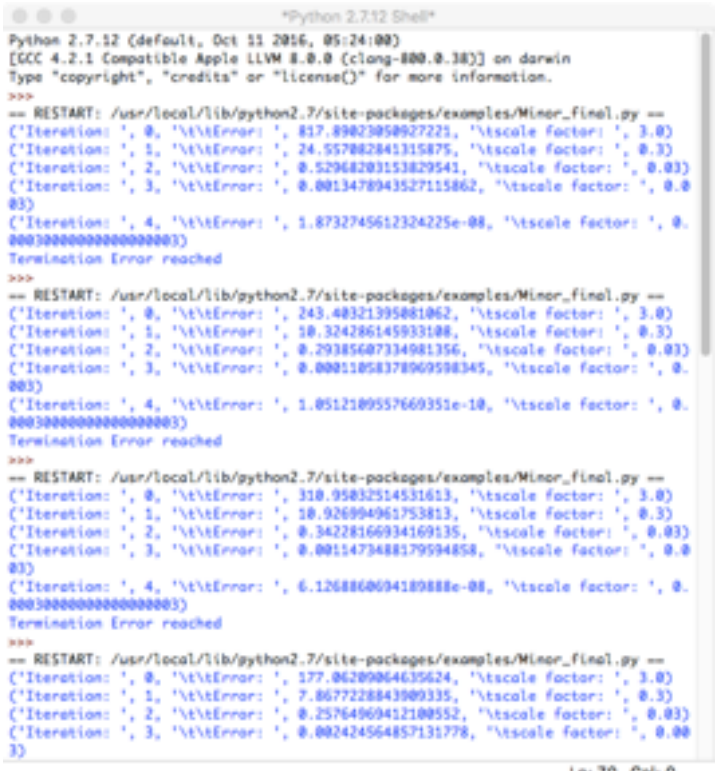
Training Our Neural Network -

Our dataset consists of 50 diseases and a set of 50 symptoms.

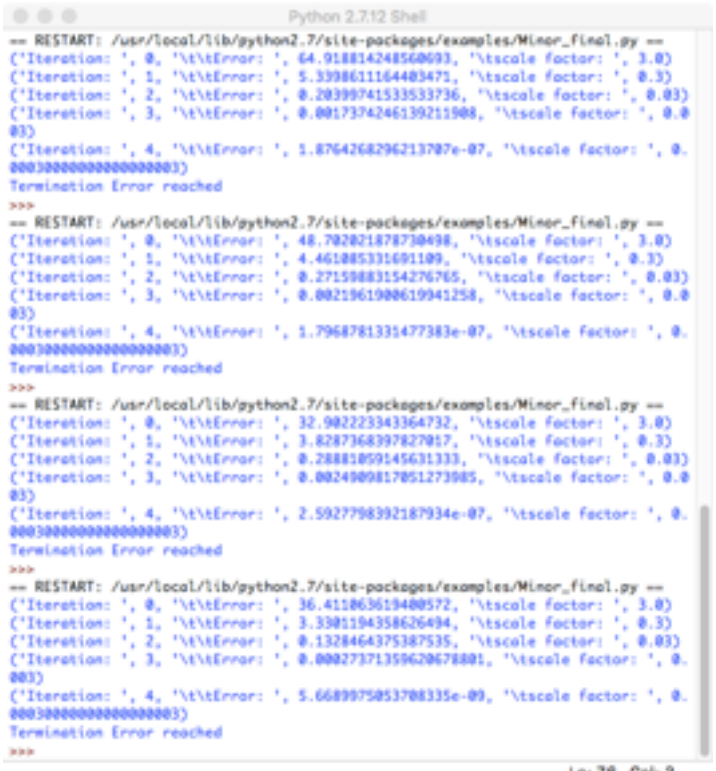


We have used **pandas** to read columns from the excel sheet to our python code and then training the data using pyrenn.

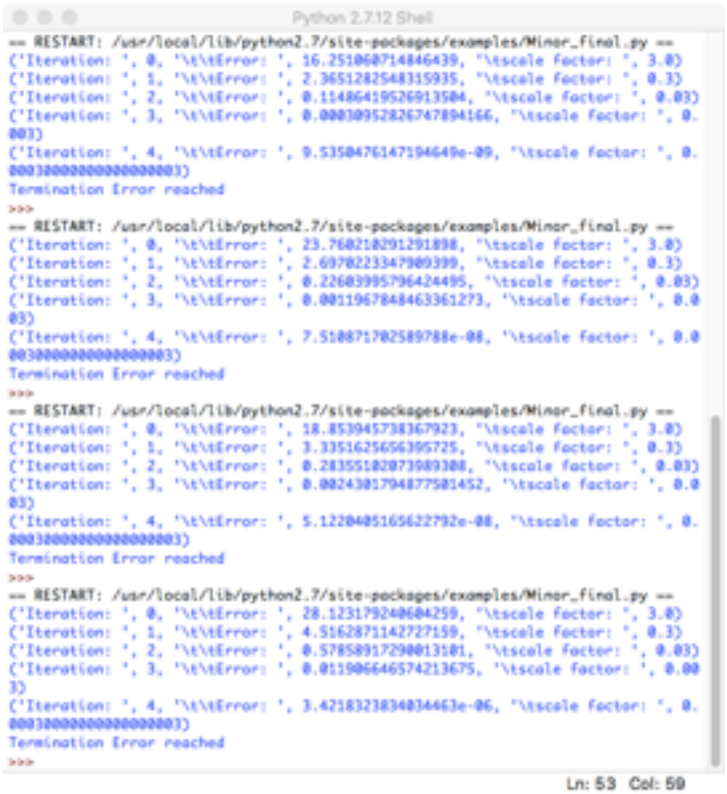
The network is trained as follows:



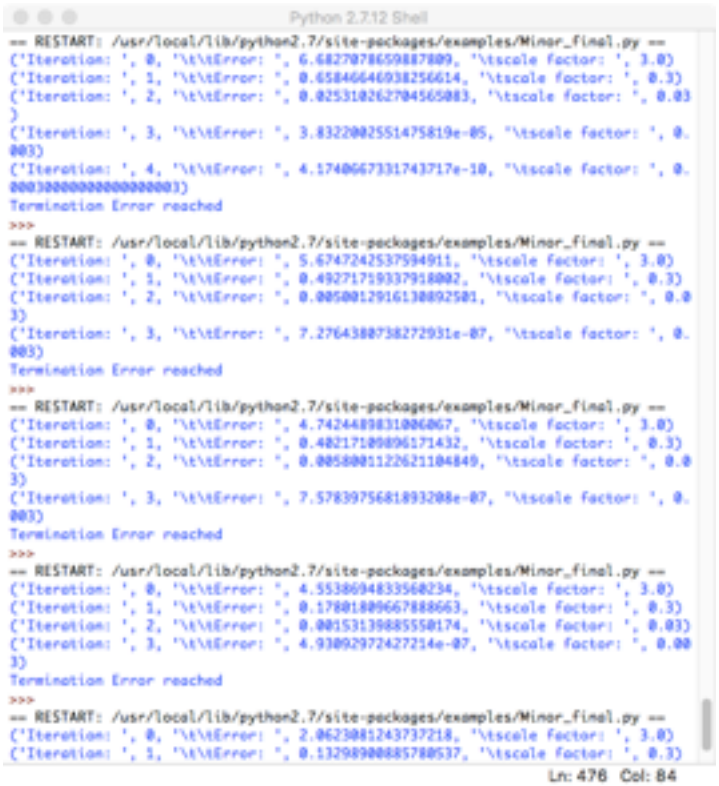
10 diseases at a time



15 diseases at a time

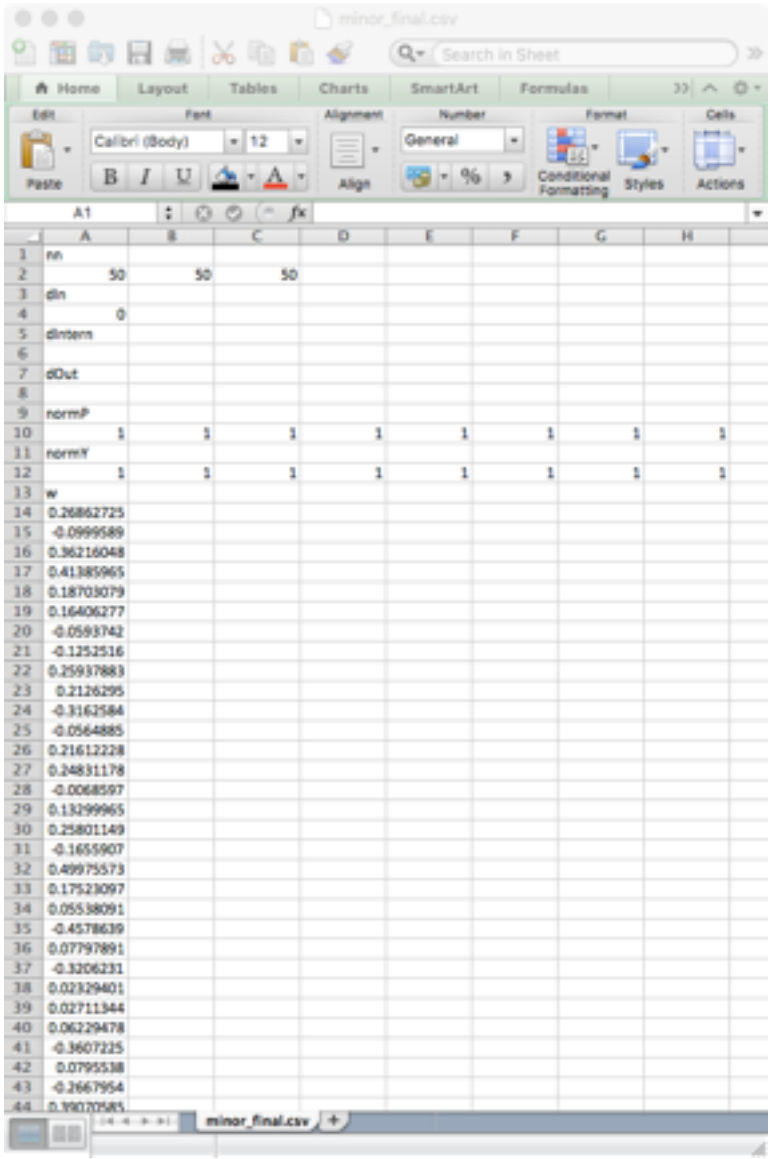


20 diseases at a time



5 diseases at a time

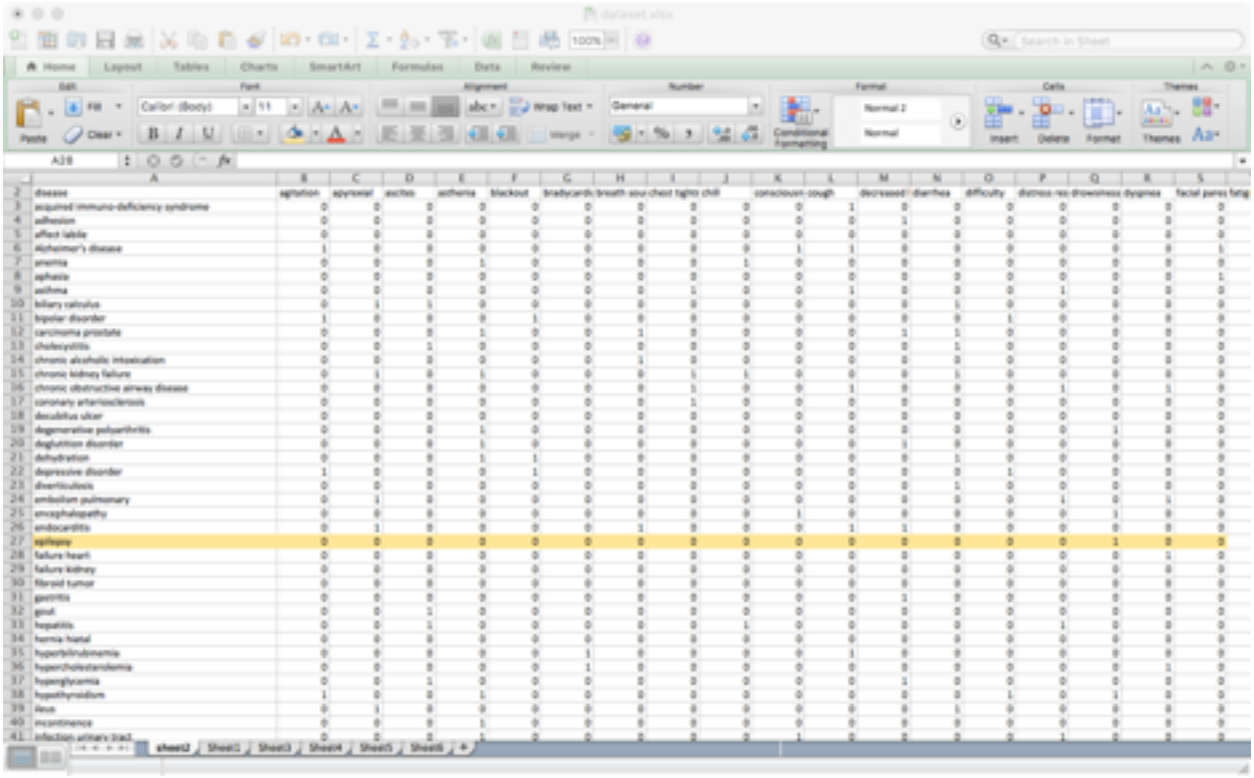
To get the most accurate results and reduce the error. This is our trained network.



	A	B	C	D	E	F	G	H
1	nn							
2	50	50	50					
3	din							
4	0							
5	dintern							
6								
7	dOut							
8								
9	normP							
10	1	1	1	1	1	1	1	1
11	normY							
12	1	1	1	1	1	1	1	1
13	w							
14	0.26862725							
15	-0.0999589							
16	0.36216048							
17	0.41385965							
18	0.18703079							
19	0.16406277							
20	-0.0593742							
21	-0.1252516							
22	0.25937883							
23	0.2126295							
24	-0.3162584							
25	-0.0564885							
26	0.21612228							
27	0.24831178							
28	-0.0068597							
29	0.13299965							
30	0.25801149							
31	-0.1655907							
32	0.49975573							
33	0.17523097							
34	0.05538091							
35	-0.4578639							
36	0.07797891							
37	-0.3206231							
38	0.02329401							
39	0.02711344							
40	0.06229478							
41	-0.3607225							
42	0.0795538							
43	-0.2667954							
44	0.98070585							

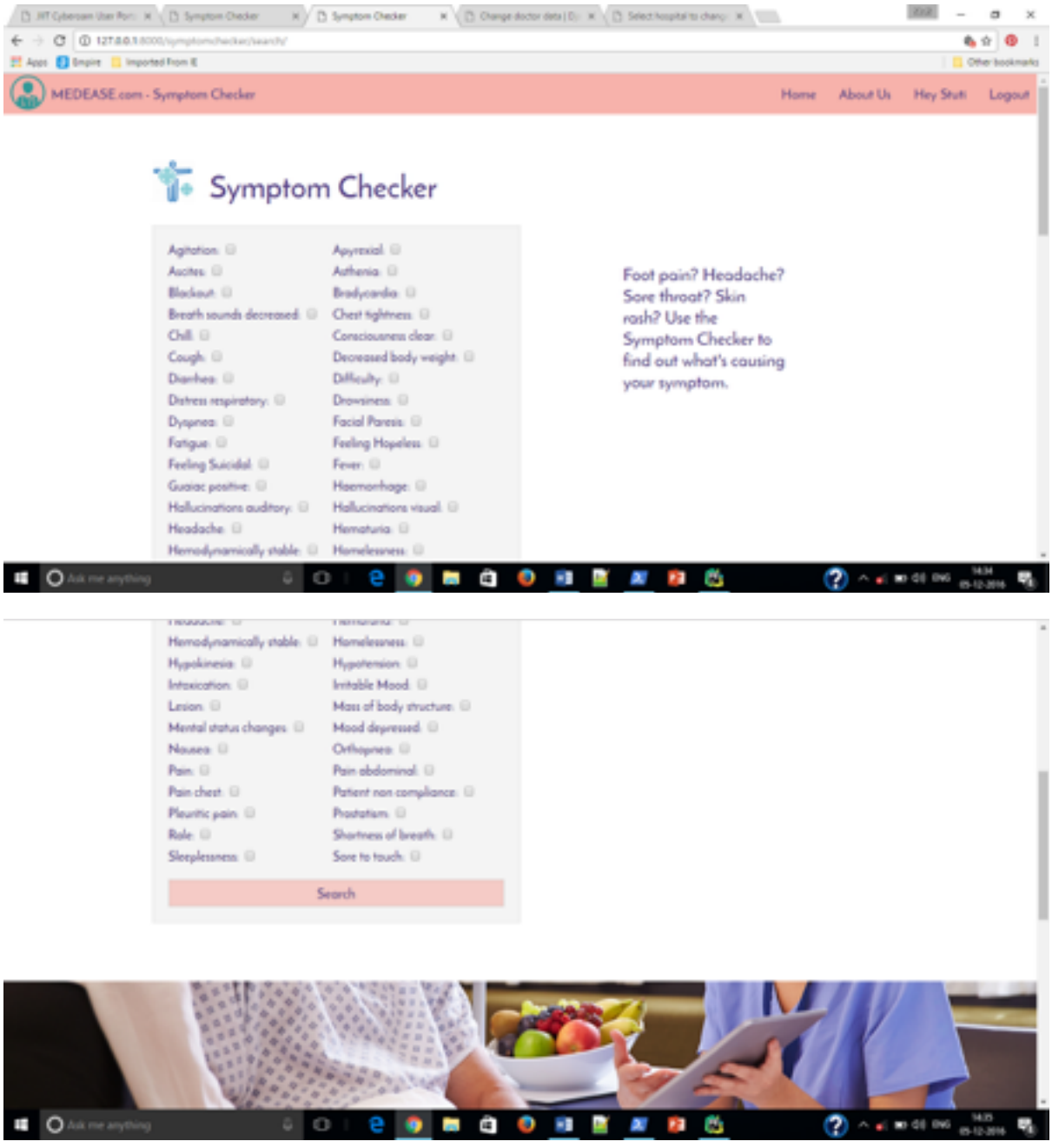
Using Our Neural Network To Get Output -

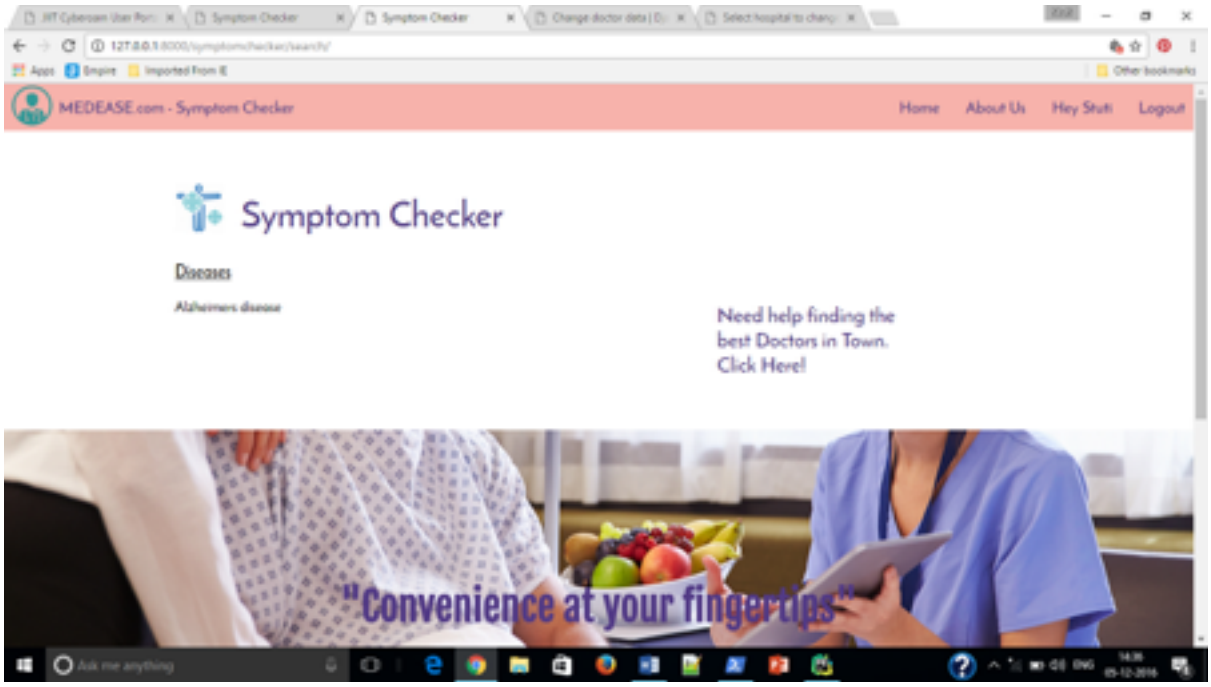
While taking input from a user, we used Django to translate those inputs into Binary and then used Openpyxl to write to our Excel sheet ‘test.xlsx’ which served as our test dataset.
This test data then gives us the required outputs on the basis of the training that we did earlier.
This output is then displayed to our user by converting it back from Binary using Django.



Below are the outputs at the output neurons which are then normalized to give values 0/1.

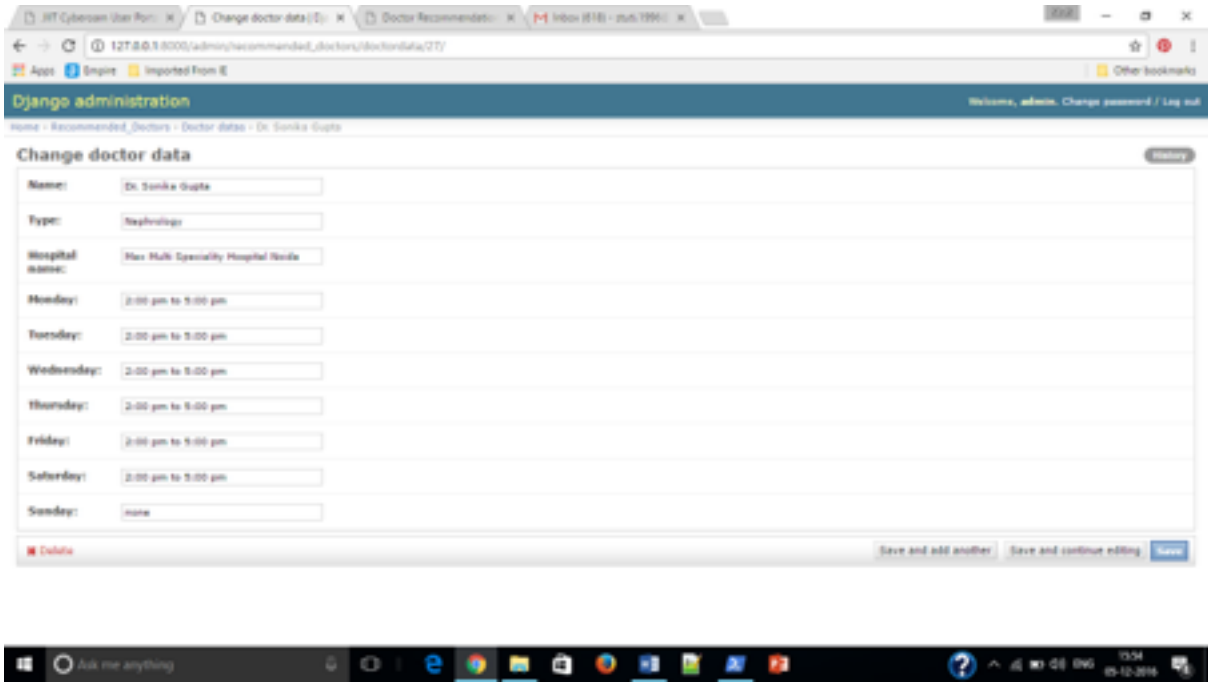
After integration with Django, the UI of the Symptom Checker is as follows:





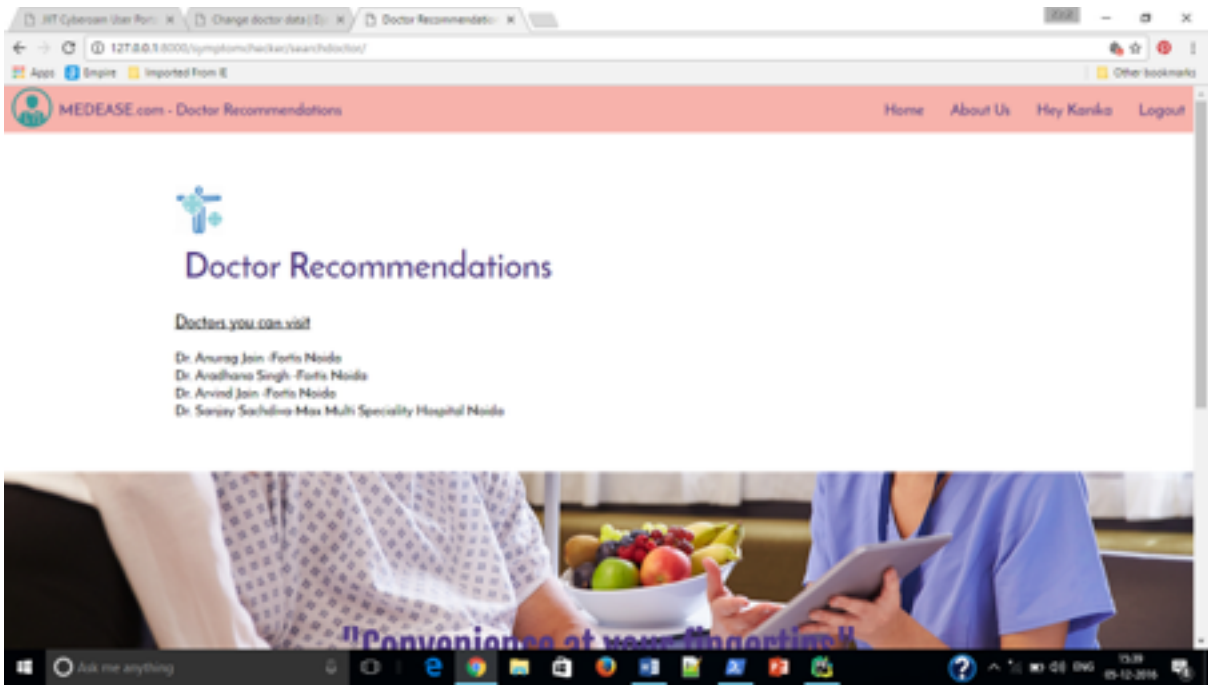
SUGGESTED DOCTORS

The table from which we retrieve our data to suggest the doctors consists of the following fields -



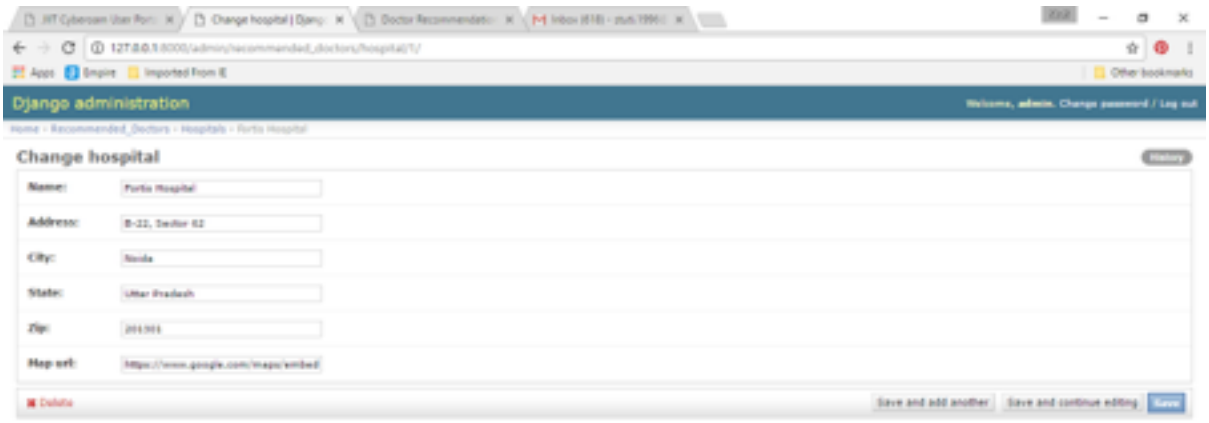
The doctors are suggested on the basis of the type of disease that resulted from our Symptom Checker thus seamlessly bounding the two applications.

The list of diseases obtained from our Symptom Checker is first mapped to its type which then helps us to map it further to the doctor that the person should consult using the field “Type” in our Doctor_data model.

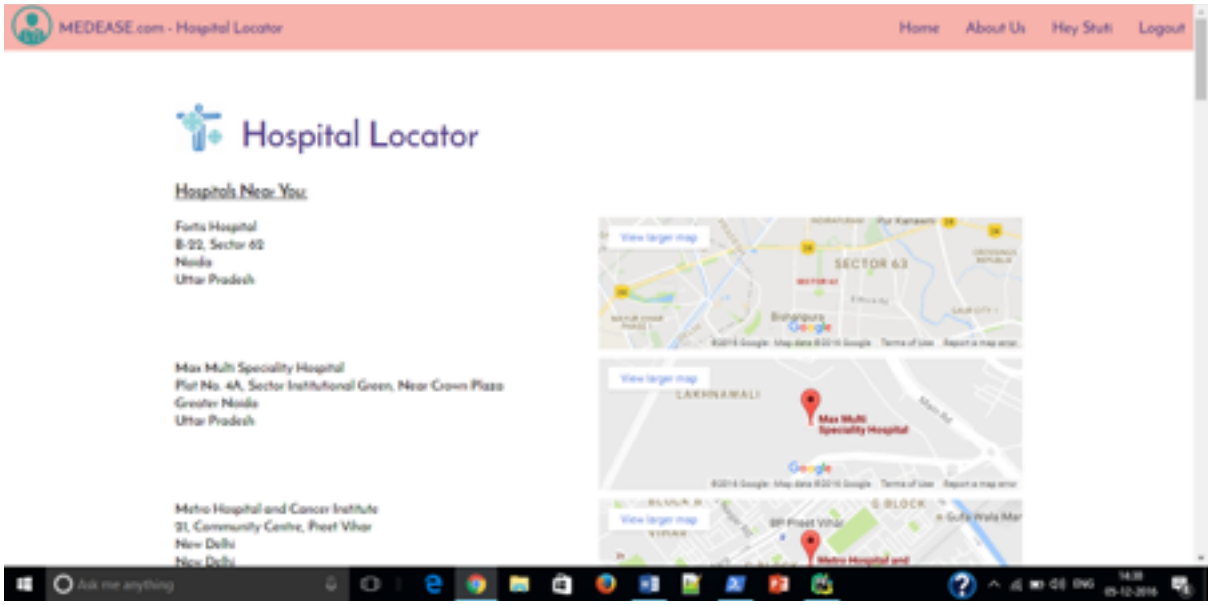


HOSPITAL LOCATOR

The table from which we retrieve our data to locate the Hospitals consists of the following fields -

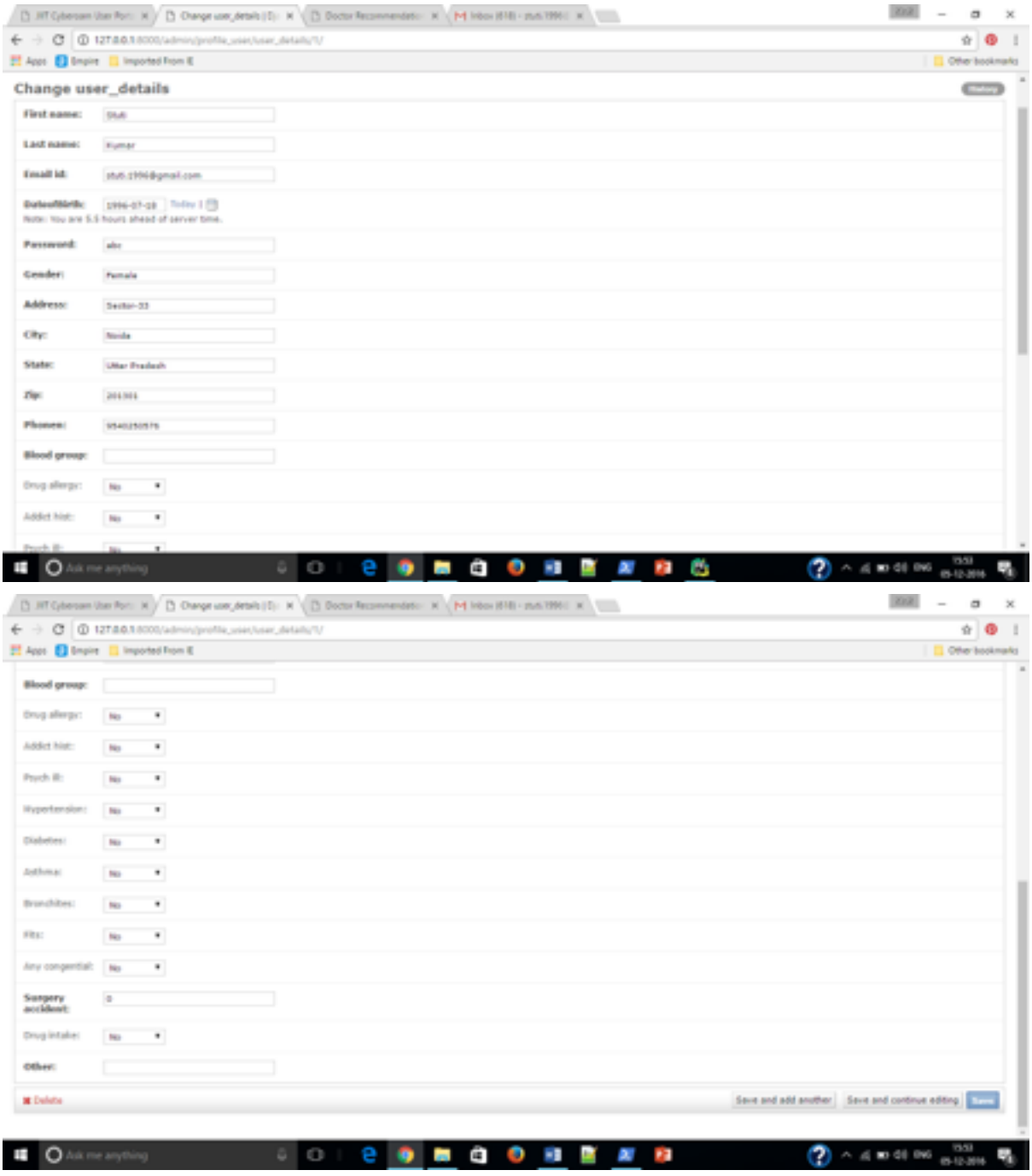


To help a user find the nearest hospitals the user must first be logged in and fill up his personal information using our medical profile app. The address , city and state fields entered by the user is used to locate the nearby hospitals .

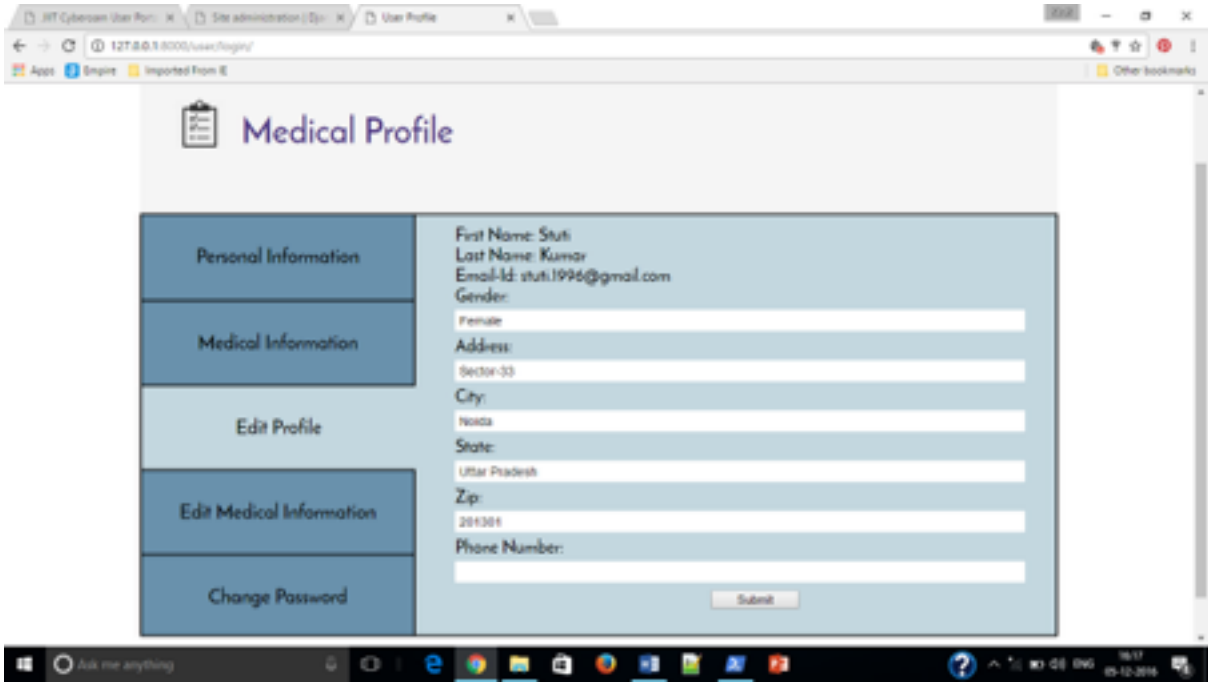


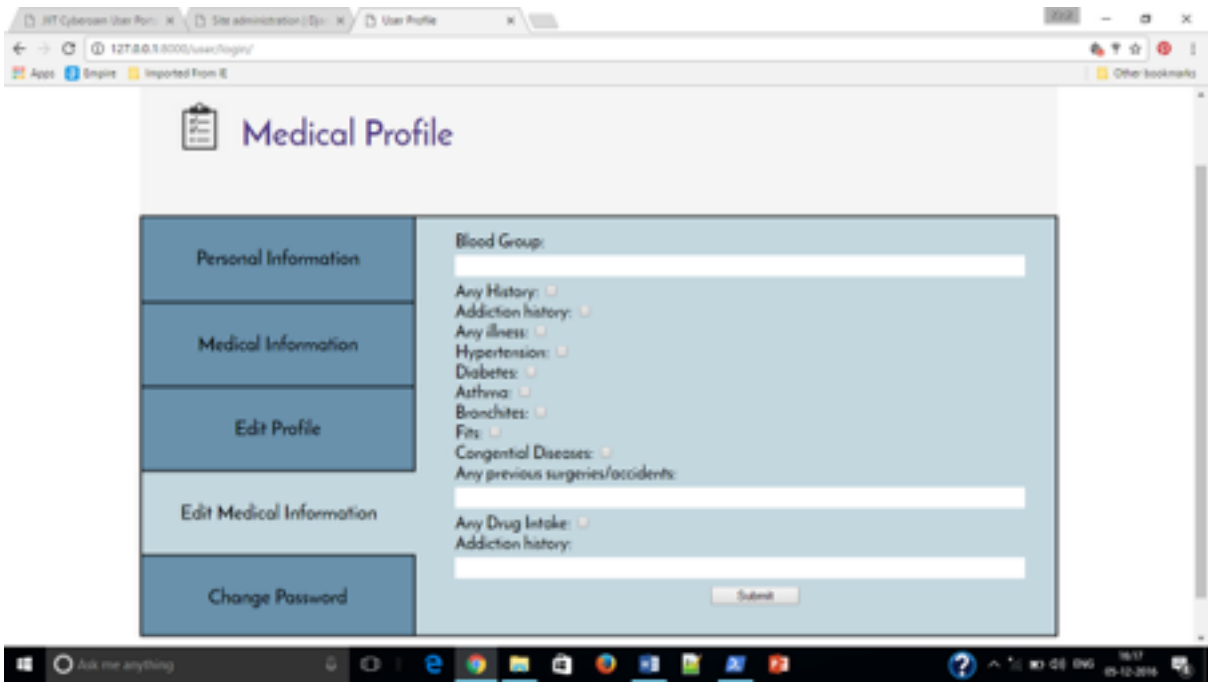
Medical History

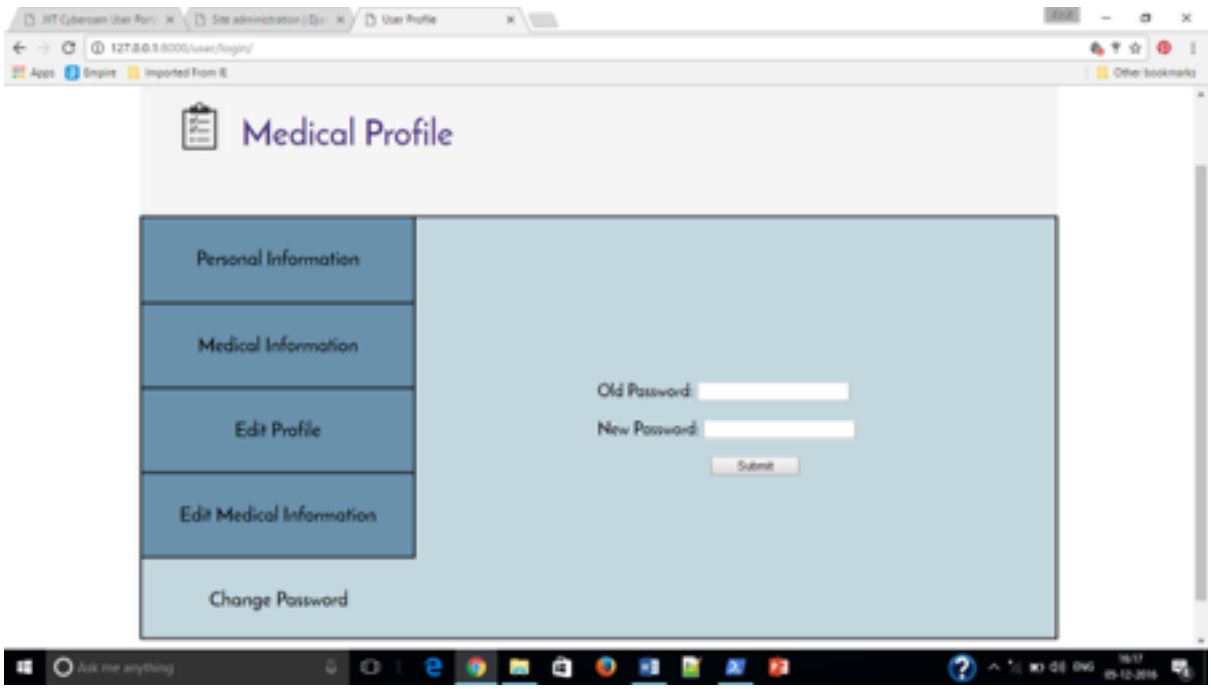
The table from which we retrieve our data to display the medical history of a user consists of the following fields -



This application is for the user to keep a track of his/her Medical history as it is very crucial for diagnosis of any other ailment which the user may contact. The choices entered in the various check boxes and character fields are stored in their respective databases and can be edited as per the user's convenience.

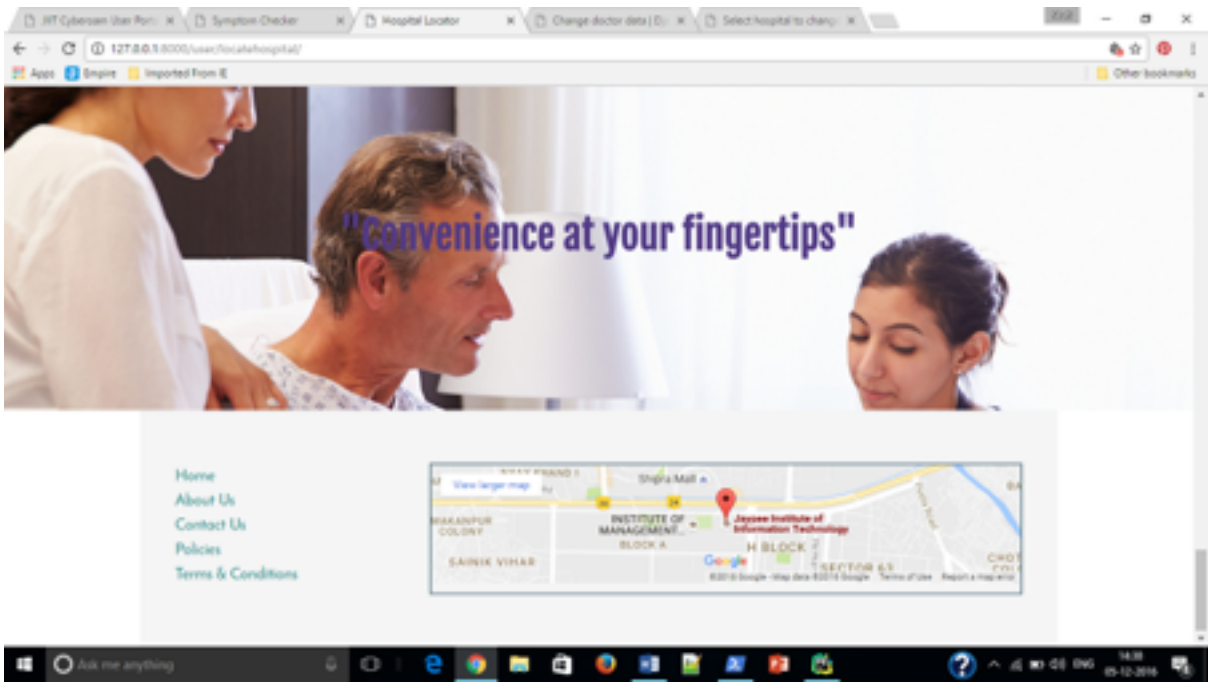

Medical Profile
Personal Information
Medical Information
Edit Profile
Edit Medical Information
Change Password
First Name: Stuti
Last Name: Kumar
Email-Id: stuti.1996@gmail.com
Gender:
Female
Address:
Sector-33
City:
Noida
State:
Uttar Pradesh
Zip:
201301
Phone Number:
Submit


Medical Profile
Personal Information
Medical Information
Edit Profile
Edit Medical Information
Change Password
Blood Group:
Any History:
Addiction history:
Any illness:
Hypertension:
Diabetes:
Asthma:
Bronchites:
Fits:
Congential Diseases:
Any previous surgeries/accidents:
Any Drug Intake:
Addiction history:
Submit


Medical Profile
Personal Information
Medical Information
Edit Profile
Edit Medical Information
Change Password
Old Password:
New Password:
Submit


Home
About Us
Login
Register
MEDEASE.com
Healthcare Made Easy


MEDEASE.com
Home
Login
Register


"Convenience at your fingertips"
Home
About Us
Contact Us
Policies
Terms & Conditions

REFERENCES

Online Documentations and Tutorials:

1. https://www.djangoproject.com/
2. www.tutorialspoint.com/django
3. https://www.youtube.com/watch?v=qgGIqRFvFFk
4. https://www.codecademy.com/learn/web
5. www.w3schools.com/html/
6. https://www.youtube.com/watch?v=bxe2T-V8XRs&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU - Neural Networks Demystified
7. pyrenn.readthedocs.io/en/latest/create.html
8. https://github.com/yabata/pyrenn
9. https://openpyxl.readthedocs.io/en/default/
10. https://docs.scipy.org/doc/numpy/reference/
11. http://pandas.pydata.org/pandas-docs/stable/

Research Papers:

- *Artifical Neural Networks (2012),G. Kumar JHA,Indian Agricultural Research Institute* - https://scholar.google.com/scholar?oi=bibs&cluster=14779615864438299204&btnI=1&hl=th
- *An Evolutionary Approach: Analysis of Artificial Neural Networks* - http://www.ijetae.com/files/Volume2Issue1/IJETAE_0112_30.pdf
- *Utilization of Neural Network for Disease Forecasting* - http://www.statistics.gov.hk/wsc/IPS063-P4-S.pdf
- *Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture* - http://ijettjournal.org/volume-3/issue-6/IJETT-V3I6P206.pdf