CHARLOTTE

Smart Health Care System

Term Project Report

Group 5

Team Members:
Eshani Ghosh
Shreya Biswas
Amirthavarshini Dhanavel
Lakshmi Srujitha Gali
Nimal Kumarr Arunkumar

Table of Contents

Abstract:
The development of a smart health prediction system is done to identify a patient's likely diseases based on their symptoms. This algorithm seeks to identify the most likely condition for a given collection of symptoms by examining historical information offered for various diseases. The suggested system makes predictions based on the given symptoms and learns from prior data using machine learning methods and methodologies. The technology is anticipated to deliver trustworthy and accurate disease forecasts, assisting patients and medical professionals in acting quickly and enhancing the efficacy of treatment.
Introduction:

Introduction to the Problem Statement:
The conventional approach to identifying diseases based on a patient's symptoms is arbitrary and prone to mistakes, which results in misdiagnosis, delayed treatment, and higher healthcare expenses. To solve this issue, it is necessary to create a smart health prediction system that can identify likely diseases based on the patient's symptoms. The proposed method uses machine learning algorithms and techniques to examine historical datasets of various diseases and determine the most likely diagnosis for a given collection of symptoms.

The system will assist doctors in providing rapid and accurate diagnoses, enhancing patient outcomes, and lowering healthcare expenditures. The creation of such a system would also aid in bridging the gap between the demand for healthcare services and the supply of medical resources, particularly in rural and underdeveloped areas. Therefore, the problem statement concentrates on the requirement for an accurate, efficient, and reliable approach to identify diseases based on symptoms in order to enhance the efficacy of healthcare services and lower costs.

1. Project Team Details:

Our team, the "ANTI-KRYPTONITE," which has members with backgrounds in computer science and a shared interest in helping the healthcare sector, is working to create a smart health prediction system.

Team Members:

Lakshmi Srujitha Gali: 801307126

She is a motivated learner who works hard in school with the goal of improving self and achieving professional success so that she can fulfill her aim of resolving the sustainability problems that nations all around the world confront. Sustainable, aspirational, and career-minded individual pursuing her Master of Science in Computer Science at UNCC currently. She has had professional work experience at Accenture Organization as an Associate Software Engineer before pursuing her higher studies for greater exposure in the same field of interest. She has a proactive personality which has been proven by her extra-curricular achievements during her undergraduate days. Attached are links to her LinkedIn and GitHub profiles:

Srujitha's linkedin profile

Srujitha's GitHub profile

Shreya Biswas: 801287328

She is currently pursuing masters in computer science at University of North Carolina Charlotte. Her interests include Machine learning and Artificial Intelligence. Her hobbies include traveling, reading and music. Other than that she is proficient at python and java. She is good at problem solving and brainstorming new ideas. She plans to work in software development and machine learning related fields in the future. Her soft skills include team work, hardworking, enthusiastic learner and leadership qualities. Shreya can also handle tasks and situations that are not easily manageable very efficiently. She knows to push herself beyond her limits and is not afraid of trying new things. She has a very professional attitude towards her work and takes her work seriously and dedicatedly. She is fun to be around and always up for a challenge.

Eshani Ghosh: 801288444

She is currently pursuing a Master's degree in Computer Science from UNC Charlotte with a concentration in Data Science. She has always been passionate about different domains related to computers and that zeal has forced her to learn more about this field. After her Bachelor’s degree in Information Technology, she worked as a software engineer for close to two years. There she got to learn a lot about software development life cycle and how real world projects operate. Now, as a part of coursework in ITCS-6112 she feels delighted for the opportunity to learn more about software and their implementations, and she is excited to work in this project on “smart health predictor”. She believes this project can help healthcare sectors to progress more by reducing the time taken to list out the symptoms of the patients and their probable diseases.

Amirthavarshini Dhanavel: 801318124

She is a MS in Computer Science student who started in the fall of 2022. Before this, she completed a bachelor's degree in Computer Science Engineering at Anna University in India. She learned about operating systems, databases, cloud computing, and machine learning. It also inspired her to collaborate with professors on various papers and work on initiatives for web development. After that, her interest in computer science expanded, and she joined UNC Charlotte to pursue it further. Deep learning, computer vision, artificial intelligence, and mobile and web app development all pique her attention. Her Github username is Amirthavarshini-Dhanavel.

Nimal Kumar Arunkumar: 801317922

He is a computer science student at the University of North Carolina at Charlotte, pursuing a Master of Science degree. Previously, he earned a Bachelor of Technology in Computer Science and Engineering from the SRM Institute in India. Nimal showcases his skills through various projects in the following fields: web development, AI, and data science projects, such as a student trading website, a healthcare chatbot, and a local sports prediction project. He is proficient in programming languages such as C, Python, JavaScript, HTML, and C++, and has experience in data structures, front-end development, databases, and operating systems. Nimal has successfully completed various certificate courses from prestigious universities.

Team Agreement:

After holding a couple of team meetings during the past two weeks post class hours, team Anti- Kryptonite has come together with some rules and regulations that must be followed by all its team members till the end of this project.

- methods of communication - Email for formal communication and a WhatsApp group for informal conversations.
- communication response times – Email: within 2-4 hours, phone, messenger, text: Immediately after checking.
- meeting attendance- When to meet: Every Tuesday after class & Thursday at 9 pm on Zoom, All meetings are mandatory.
- division of work – During every stage of the project we will reflect on our performance as a team and share feedback. The team leaders Srujitha and Shreya will combinedly decide on the submissions as to who will submit a group assignment. We will make sure that every team member participates equally.

2. Project Proposal:

Individual Qualification and Strengths:

Eshani Ghosh : Her strengths include Programming and developing the models in machine learning and working with frontend. She also makes clear and organized documentations.

Shreya Biswas: She is very insightful in designing the project and the related ideas including abstraction and maintenance of the modularity of the project. She also has interests in programming and developing modules for the project.

Lakshmi Srujitha Gali: She is very dedicated to managing the team and organizing all the team activities including allocation of work. She is a good presenter and also contributes to the designing of the application with unique ideas.

Amirthavarshini Dhanavel: Her strengths include organizing all the related documentations and keeping track of the changes if induced within the software development life cycle. She also has the ability to ensure all the scenarios are considered when developing an application and hence is insightful in testing applications.

Nimal Kumar Arunkumar: His strengths include programming and organizing the team activities. He is also a good presenter and is very efficient in making the project ideas understandable to any level of audience.

Apart from their qualifications and strengths, each member of the group are eager to work in all the phases of the software development life cycle including documentation, presentation, programming and organizing the team activities.

Facilitators:

Srujitha and Shreya will be the two facilitators from the team and will be managing all the team related activities and keeping track of all the project related activities.

Statement of the goal:

This project on developing a smart health predictor system is aimed to predict the probable diseases of a patient based on the_inputs of the symptoms given. This system is expected to predict the diseases based on the past datasets provided for the diseases and predict the most likely disease for the given symptoms based on the previous data.

Services to be provided by the System:

A smart health prediction system is a system that may precisely identify probable illnesses based on patient symptoms and deliver prompt probable disease with the aid of machine learning algorithms like decision tree classifier, random forest classifier, and others.

This system will have two functional modules- the backend and the frontend. The backend will be responsible for training pre-built machine learning algorithms with some past data about the illness of the patients and the related symptoms. The testing dataset will be used to check if the system is precisely able to predict the diseases with great accuracy. Two predefined algorithms will be used to predict the illness and the accuracy of both can be compared:

With the help of the decision tree classifier method, a hierarchical tree-like structure can be created, with each node standing for a feature and each branch for a decision based on that feature. The algorithm discovers the ideal tree structure to reduce the training data set's classification error. Using the generated decision tree and a patient's list of symptoms, the disease can then be predicted.

Another machine learning algorithm that can be used to create a prediction model for the smart health-based system is the random forest classifier. To produce a final forecast, this algorithm builds numerous decision trees and merges them. To avoid overfitting and increase the precision of the final prediction, the algorithm builds each decision tree by randomly choosing a portion of the features and the training data set.

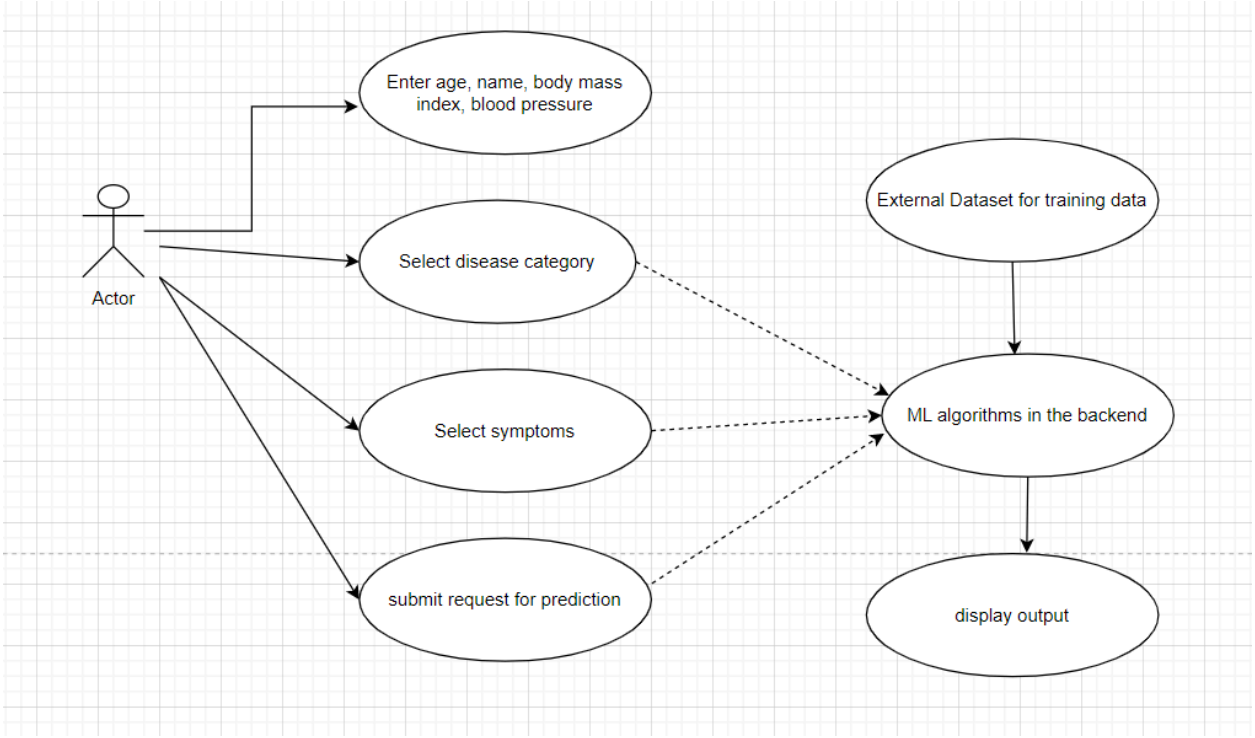
To make the models accessible to the users a frontend will be developed using a GUI and that can be used by the users to give their symptoms as input and the probable illness will be displayed on the screen. The use of machine learning algorithms like decision tree classifier, random forest classifier, and others, can help to build accurate prediction models. With proper training and testing datasets, the system can be fine-tuned to provide precise and timely medical interventions, resulting in better patient outcomes.

System Customers:

The patients will be the expected system customers whose data will be required as input to the system. This system can be used in healthcare facilities for initial screening of the patients with their symptoms. This can help in making the diagnosis process faster for the doctors and reduce the wait times for the treatments and medical tests of the patients.

3. System Requirements:

Use Case Diagram:



Use Cases:

1. As a user, I want to give the symptoms I have as an input so that this system can predict the disease I am suffering from.

a. Pre condition: Entering name

b. Post Condition: 2

c. Is Epic: No

2. As a User, I will have to choose any given way here given as algorithm to predict my disease.

a. Pre condition: 1

b. Post Condition:

c. Is Epic: No

3. As a User, I will be able to see the predicted disease I am suffering from.

a. Pre condition: 1,2,4,5,6

b. Post Condition:

c. Is Epic: No

4. As a developer, I will take input of the user name, three symptoms the user is having to predict the disease.

a. Pre condition: 1,2

b. Post Condition:

c. Is Epic: No

5. As a developer, I will use the choice of the machine learning algorithm given by the user and use it to predict the disease and also determine the accuracy of the algorithm used.

a. Pre condition: 1,2,4

b. Post Condition:

c. Is Epic: Yes

6. As a developer, I will be displaying the result in the frontend.

a. Pre condition: 4,5

b. Post Condition: 3

c. Is Epic: No

Non-Functional Requirements:

1. UI should be easier to use for the user who are not technically equipped.
2. The maximum turnout time should be less than a minute.
3. Our target audience are health patients.

Functional Requirements:

1. The UI/UX should be able to take inputs from the user like age, gender, symptoms.
2. The API framework should be able to take the input from the frontend and convert them into the data format required by the backend.
3. Machine learning models need to be trained and tested with past datasets to help in predicting diseases.
4. The models should be correctly able to predict the disease based on the user inputs.

4. Product Backlog

Updated User Stories:

Team : Anti-Kryptonyte

1. As a user, I should be able to provide my name.

a. Pre condition:

b. Post Condition:

c. Size: 1

d. Is Epic: No

e. Priority: Low

2. As a user I should be able to select the scrollable list of symptoms I am suffering from so that the system can predict the disease.

Pre condition:

Post Condition: 3

Size: 3

Is Epic:

No

Priority:

High

3. As a user I should be able to choose the way of algorithm to be used for predicting the disease.

a. Pre condition:

b. Post Condition:

c. Size: 3

d. Is Epic: No

e. Priority: High

4. As a developer, I should be able to take input of the user name and the symptoms of the user.

a. Pre condition: 1, 2

b. Post Condition: 5

c. Size: 2

d. Is Epic: No

e. Priority: High

5. As a developer, I should be able to execute the classifier algorithm that the user provides.

a. Pre condition: 3

b. Post Condition:6

c. Size: 1

d. Is Epic: Yes

e. Priority: High

6. As a developer, the results should be shown on the frontend for the user.

a. Pre condition: 2,3

b. Post Condition:

c. Size: 1

d. Is Epic: No

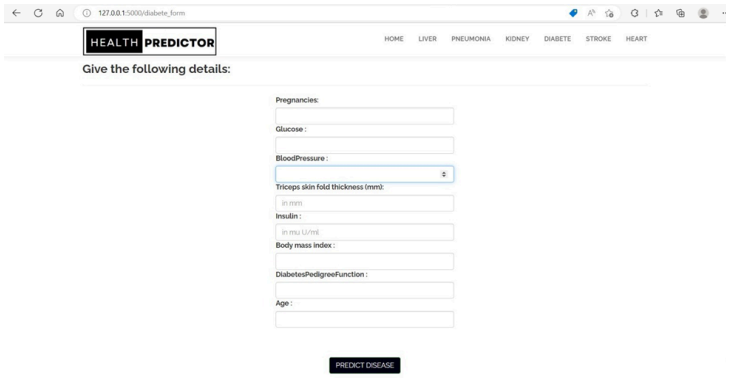
e. Priority: High

Sprint 1:

In our first sprint we are working on UI components and making the UI more accessible for the user. The frontend will take the username as the input and the symptoms of the user. Using these we will be predicting the disease using the classifier algorithms.

Our targeted user stories are 1,2 and 3.

UI Design :



5. Sprint Reports:

Sprint 1

For our project our 1st sprint plan would be to achieve our target and solve the user stories requirements and problems. At present users can log in and predict their disease based upon their inputs.

Implemented User Stories

- Story 1 - As a user, I want to have drop down boxes.
- Story 2 - As a user, I want the page to be interactive.
- Story 3 - As a developer, I require buttons with better labeling for prediction algorithms.
- Story 4 - As a developer, I want the page to have more input fields to map the symptoms.
- Story 6 - As a developer, I want a smooth scrolling page which will make it more interactive.

Changes

In order to fulfill our user requirements based upon the story We have to make some upgrades to our targeted user stories. We will be choosing the support side of the application after we solve the issues.

The following new stories were added to the product backlog to be worked on in future sprints.

- Story 10 - As a user, I want to input my age, gender, and symptoms to the smart health prediction system so that it can predict my health condition.
  - Pre-conditions: 8
  - Post- conditions:
  - Is Epic: No
  - Size: 3
  - Priority: High

- Story 11 - As a developer, I want to use the Flask web framework to create a user interface for the smart health prediction system so that users can input their data
  - Pre-conditions: 1,2,4
  - Post- conditions: 13
  - Is Epic: No
  - Size: 3
  - Priority: High

- Story 12 - As a developer, I want to use the Pandas library to read the user input data from the Flask web interface and store it in a data frame so that it can be processed and analyzed by the smart health prediction system.
  - Pre-conditions: 1,4
  - Post- conditions:13
  - Is Epic: No
  - Size: 3

- Priority: High

    - Story 13 - As a user, I want to see the predicted health condition and recommended actions based on my input data so that I can take appropriate actions to improve my health.
- Pre-conditions: 4
- Post- conditions: 12
- Is Epic: No
- Size: 3
- Priority: High

    - Story 14 - As a developer, I want to use the Scikit-learn library to train and test machine learning models for health prediction based on user input data.
- Pre-conditions: 15
- Post- conditions:
- Is Epic: No
- Size: 3
- Priority: High

    - Story 15 - As a developer, I want to use the Matplotlib library to visualize and display the predicted health condition and recommended actions to the user interface.
- Pre-conditions: 12
- Post- conditions:
- Is Epic: No
- Size: 3
- Priority: High

Reflection

Our key point from this sprint cycle is that some of our user stories might want to be further decomposed because some of them required implementations that were too extensive for us to complete in the allotted time. The front end created which involves HTML, CSS, JavaScript is

now further divided to have Jquery some functionalities like smooth scrolling and custom fonts and to interact with HTML.

Pending User Stories

- Story 5 - As a developer, I want to implement JQuery through which communication with HTML is easier.
  - Pre-conditions:
  - Post- conditions:1,6
- Story 7 : As a developer, I want to add a dataset in the backend for the models to be trained.
  - Pre-conditions:
  - Post- conditions:7,8,15
- Story 8 - As a developer, I want to integrate the smart health prediction system with a database to store user input data and predict health conditions for future analysis and improvement of the system.
  - Pre-conditions: 7
  - Post- conditions:15
- Story 9 - As a developer, I want to improve the accuracy of the machine learning models by adding more data sources and features to the system.
  - Pre-conditions: 15
  - Post-conditions:
- Story 10 - As a developer, I want to add security features to the system to ensure that user input data is protected and not accessible by unauthorized users.
  - Pre-conditions: 8

- Post- conditions:

- Story 11 - As a user, I want to input my age, gender, and symptoms to the smart health prediction system so that it can predict my health condition.
  - Pre-conditions: 1,2,4
  - Post- conditions:13

- Story 12 - As a developer, I want to use the Flask web framework to create a user interface for the smart health prediction system so that users can input their data
  - Pre-conditions: 1,4
  - Post- conditions: 13
- Story 13 - As a developer, I want to use the Pandas library to read the user input data from the Flask web interface and store it in a data frame so that it can be processed and analyzed by the smart health prediction system.
  - Pre-conditions: 4,12
  - Post- conditions:15

- Story 16 - As a developer, I want to use the Matplotlib library to visualize and display the predicted health condition and recommended actions to the user interface.
  - Pre-conditions: 13
  - Post- conditions:2

The important user stories to be implemented from sprint 1 are user stories 7, 11, 12 and 13 . These stories represent showing prediction output in a smooth manner. After Sprint 1, the user can choose symptoms from given drop down boxes and click the button to choose which prediction algorithm they need.

Sprint 2

In our second sprint of our Smart Health Predictor System we implemented usage of Python libraries to read data from the frontend and show output to the frontend:

Implemented User Stories

User Input and Data Reading

- Story 11- As a user, I want to input my age, gender, and symptoms to the smart health predictor system so that it can predict my health condition.
- Story 12- As a developer, I want to use the Flask web framework to create a user interface for the smart health prediction system so that users can input their data.
- Story 13- As a developer, I want to use the Pandas library to read the user input data from the Flask web interface and store it in a data frame so that it can be processed and analyzed by the smart health prediction system.

Health Prediction and Output Display

- Story 14- As a user, I want to see the predicted health condition and recommended actions based on my input data so that I can take appropriate actions to improve my health.
- Story 15- As a developer, I want to use the Scikit-learn library to train and test machine learning models for health prediction based on user input data.
- Story 16- As a developer, I want to use the Matplotlib library to visualize and display the predicted health condition and recommended actions to the user interface.

Changes

We are dividing the programming part into two sections, one for reading data and input values from the user and the other being predicting and displaying the results. We have downloaded and utilized python libraries to read and input data from the users. We wish to visualize the data results of our project to create a general understanding among the users developers.

- Story 17: As a developer, I want to make the data collected from the frontend to be stored in a dataframe so that it can be used directly by the machine learning algorithms directly.

Pre-conditions: 11

Post- conditions:

Is Epic: No

Size: 2

Priority: Medium

Reflections

This sprint was completely focused on processing the data in the backend and the frontend. Usage of Flask python framework with Pandas library made it possible to correctly collect data from the user. Also, we tried on making the data to be added to the dataset we will be adding such that the models can work more accurately in the future based on the inputs given in the past.

Pending Stories

- Story 5 - As a developer, I want to implement JQuery through which communication with HTML is easier.
  - Pre-conditions:
  - Post- conditions:1,6
- Story 7 - As a developer, I want to add a dataset in the backend for the models to be trained
  - Pre-conditions: 7
  - Post- conditions:8,15S
- Story 8 - As a developer, I want to integrate the smart health prediction system with a database to store user input data and predict health conditions for future analysis and improvement of the system.
  - Pre-conditions: 7
  - Post- conditions:15

- Story 9 - As a developer, I want to improve the accuracy of the machine learning models by adding more data sources and features to the system.
    - Pre-conditions: 15
    - Post-conditions:

- Story 10 - As a developer, I want to add security features to the system to ensure that user input data is protected and not accessible by unauthorized users.
    - Pre-conditions: 8
    - Post- conditions:

In the next sprint, we will be working on developing machine learning models that can predict the diseases correctly based on the inputs obtained in this sprint.

Sprint 3

In our third sprint, we target to develop the models for processing the user symptoms and coming out with a predicted result for the disease they might be having. The functionality of the application has been implemented with this backend and now this application can predict some diseases based on the user inputs.

Implemented User Stories

- Story 7 - As a developer, I want to add a dataset in the backend for the models to be trained
- Story 8 - As a developer, I want to integrate the smart health prediction system with a database to store user input data and predict health conditions for future analysis and improvement of the system.
- Story 9 - As a developer, I want to improve the accuracy of the machine learning models by adding more data sources and features to the system.
- Story 15 - As a developer, I want to use the Scikit-learn library to train and test machine learning models for health prediction based on user input data.
- Story 18 - As a developer, I want to test the accuracy of the system developed for predicting a disease using the R-squared mean library.

Changes

The dataset used did not include all the possible symptoms that are required for the model to work accurately. So we added more data to the dataset.

- Story 19 - As a developer, I added more data into the dataset and checked the accuracy of the system for a given prediction based on the user input.
  - Pre-conditions:
  - Post- conditions: 18
  - Is Epic: No
  - Size: 1
  - Priority: High

Pending User Stories

- Story 20 - As a developer I want to implement random forest classifier and naive bayes Machine learning algorithms in the backend of this system for better accuracy and precision
  - Pre-condition:7,8
  - Post-Condition: 18
- Story 21 - As a developer, I want to make the output of the prediction model to be displayed to the frontend for the user.
  - Pre-Condition: 15
  - Post-Condition:

Reflection

Implementing the third sprint was challenging because we needed to make models with good accuracy such that the application can predict the diseases accurately. Using multiple

algorithms can help to remove discrepancies and so we decided to use Decision Tree Classifier, random forest classifier and Naïve Bayes algorithm that can work by taking the inputs and based on the training and the testing sets can predict the disease. The algorithm which has already been implemented has an accuracy of greater than 90%. We are trying to replace the data that is used for the initial working of these algorithms to be replaced by more specific and real life data sets such that they can work more accurately and predict the diseases correctly.

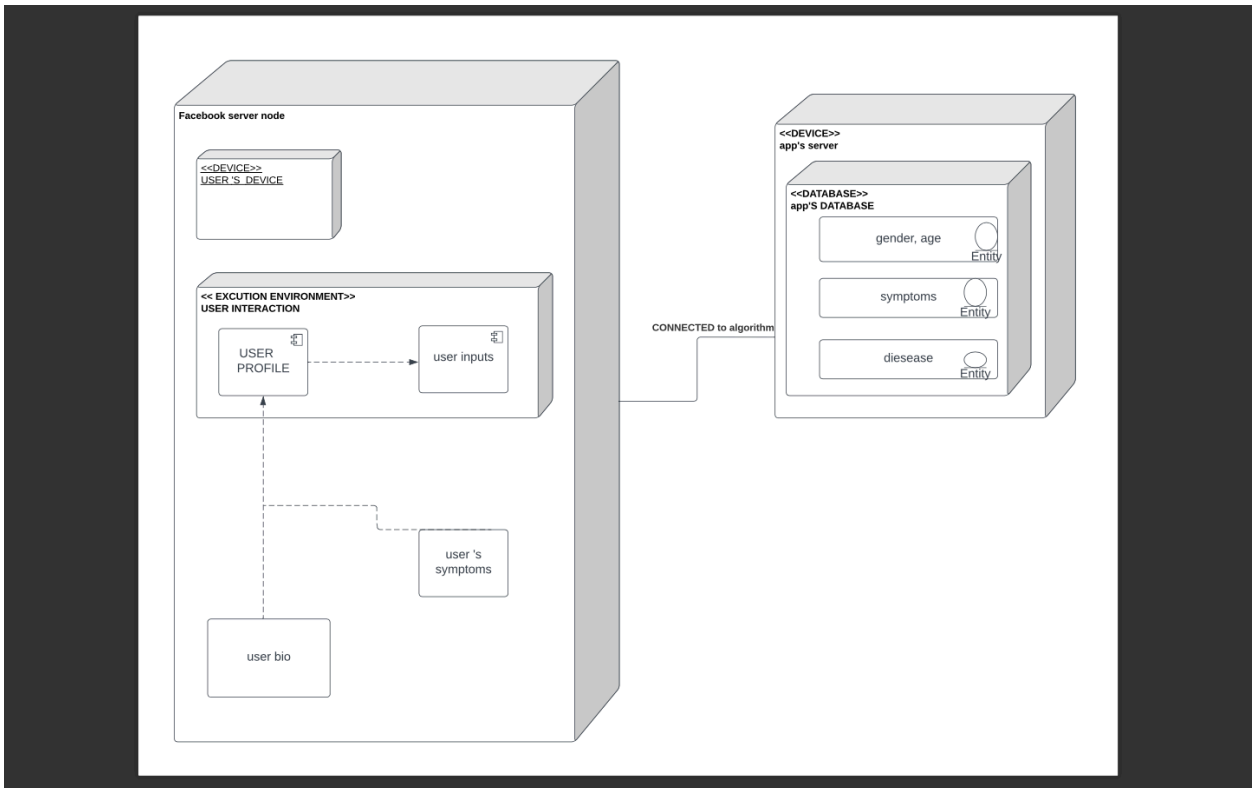
For the next sprint, we will be working on including the other two machine learning algorithms and checking the accuracy of the entire system. We will be testing the system and refactoring on the code to make it more modular and understandable.

6. Reused Components and Program

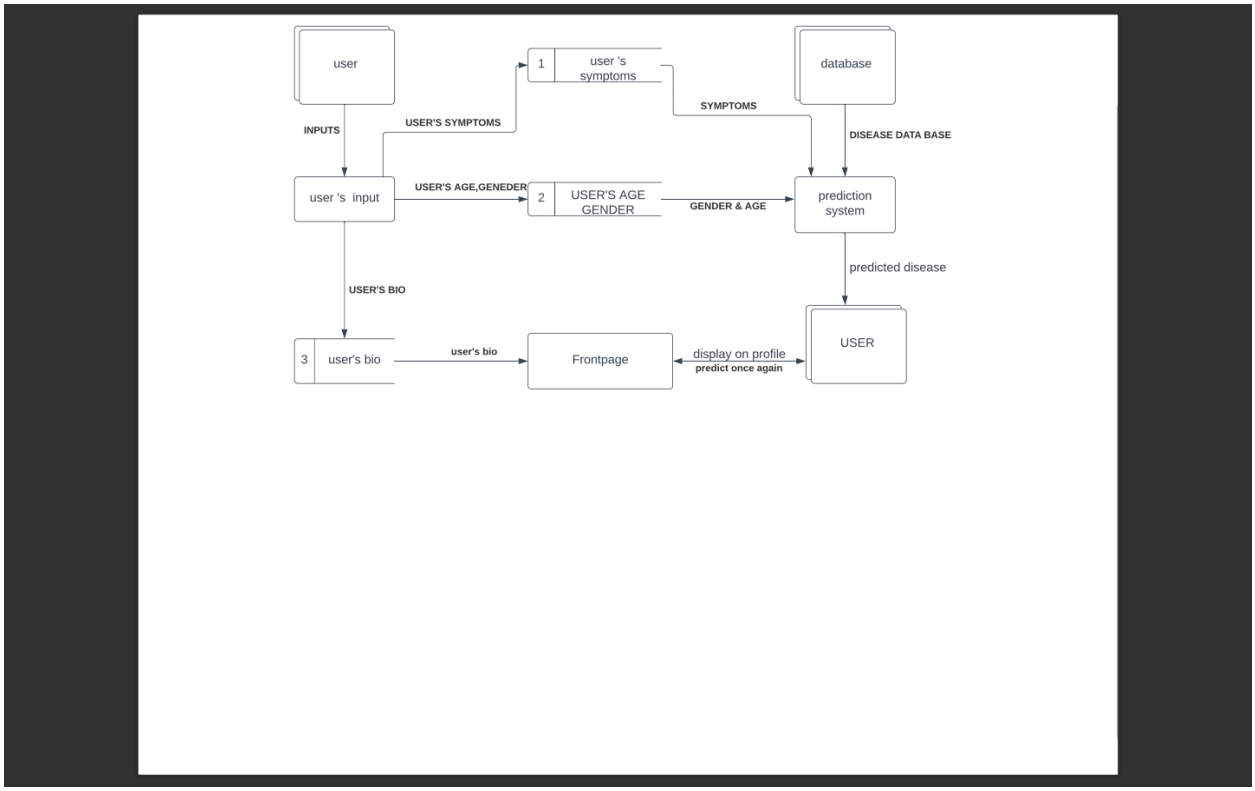
The machine learning models that were used in our application were predefined models which were taken from Kaggle. We reused these models in our application by connecting it with a front end using Flask and launching it in the form of a webpage application.

7. UML Diagrams

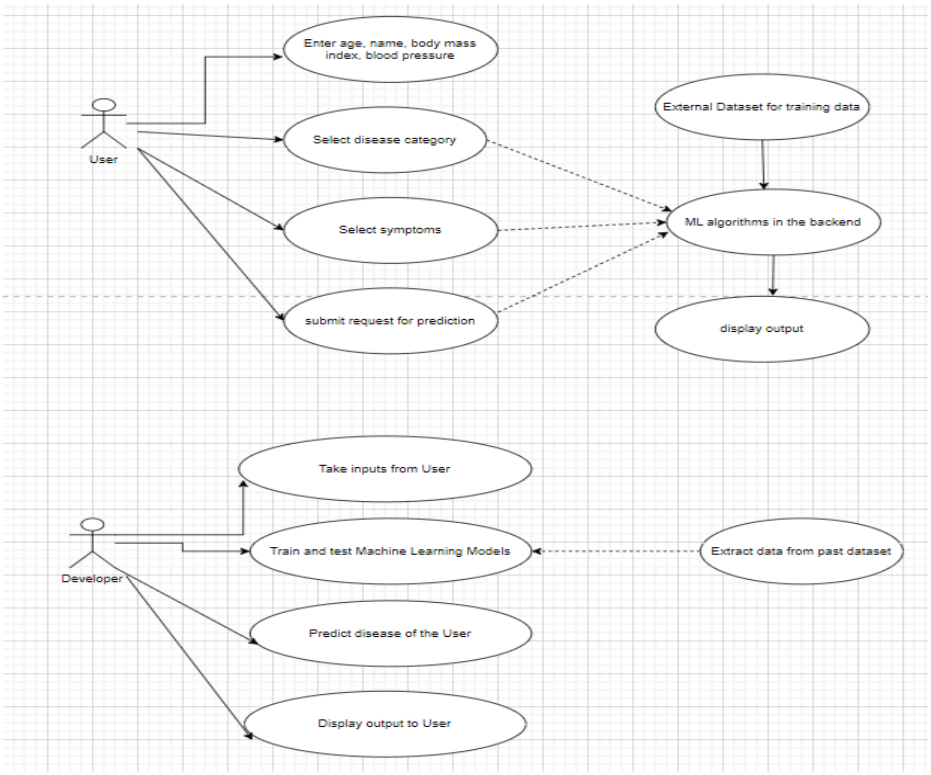
1. System Architecture diagram:



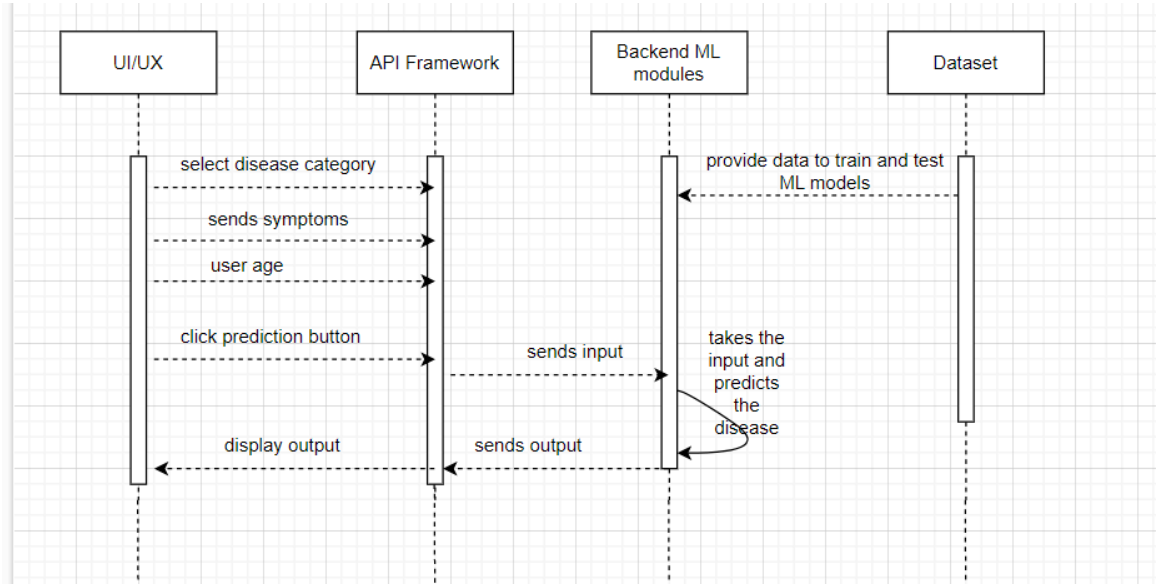
2. Data Flow Diagram:



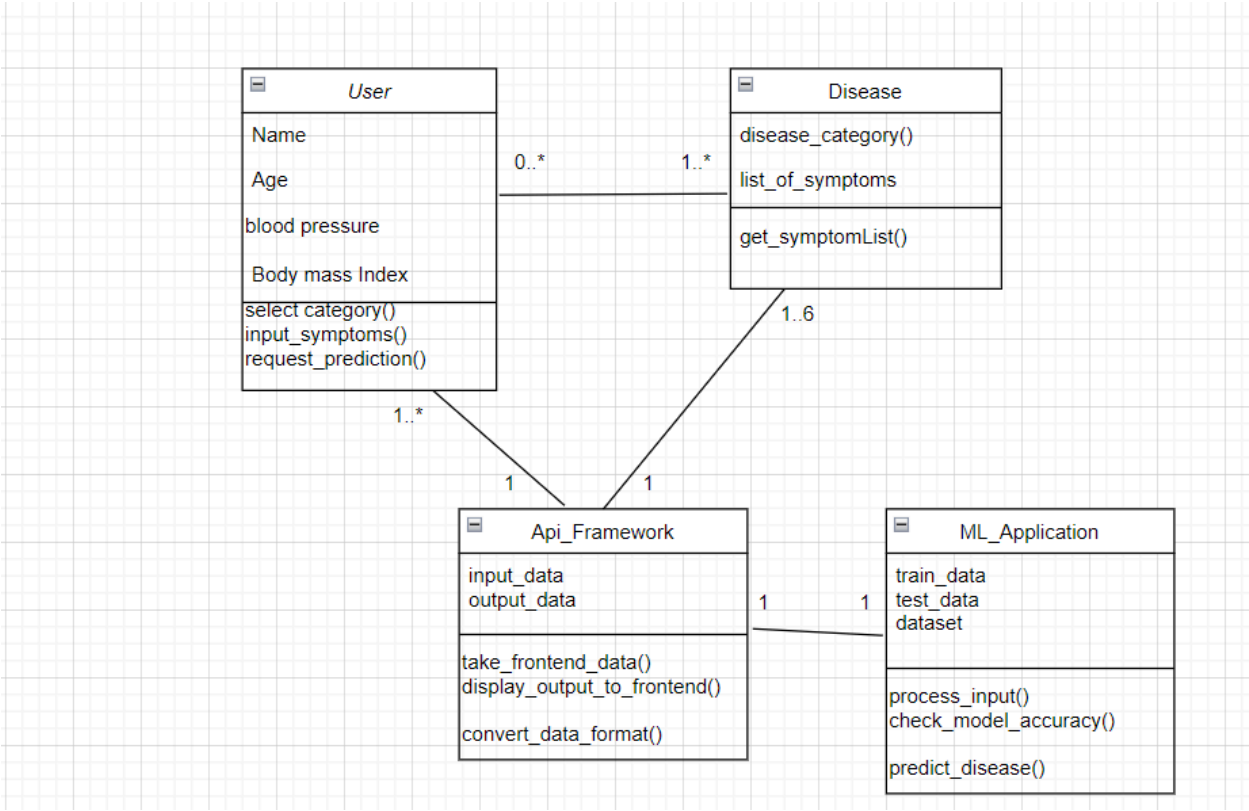
3. Use Case Diagram:



4. Sequence Diagram:



5. Class Diagram:



8. Programming Languages

For the frontend, i.e, for making the UI design and the webpage, we have used HTML, CSS and Javascript. For the backend, i.e., for all the machine learning algorithms, we have used python programming language and its libraries.
Predefined machine learning classification algorithms and models were used for this application. The algorithms used were based on decision tree classifiers, random forest classifiers, and convolutional neural networks, which were integrated with the Pickle library of Python for implementation. Separate models were used for each of the disease categories.

9. Tools and Environment

Jupyter Notebook is used for all the backend coding, i.e the python code
Notepad++ is used for all the frontend coding, i.e HTML , CSS and Javascript codes

10. Datasets Used

Predefined machine learning models for different types of diseases were used for our application. The datasets used to train and create these models were taken from Kaggle.

11. User Manual

Deployment

Our application, called smart healthcare system, allows a patient to predict the disease they are facing or may be facing by taking in inputs of the symptoms. The target audience will be health care patients. This is meant to give quick results to give an idea about the disease before running all the laboratory tests which may take further time. The users have to select the symptoms from the list provided in a drop down menu. The user can also input their age. There is a button to predict the disease.

The application consists of two components, front end and back end. Both of them are independently deployed and implemented.

The front end is implemented using HTML, CSS, and Javascript. It makes and designs the webpage of our application where the main prediction system is deployed.

The backend uses Python programming language. Specifically, machine learning classifier models are implemented using scikit library and tensorflow. Furthermore, the backend development is handled by using the Flask web framework to create a user interface to input data in it.

Algorithms used in the different models:

liver=sklearn library , ensemble learning RandomForestClassifier

kidney=sklearn library, ensemble learning RandomForestClassifier

heart=sklearn library, LogisticRegression

lungs=pytorch, convolutional neural network

stroke=sklearn library , LogisticRegression

diabetes=sklearn library, ensemble learning RandomForestClassifier

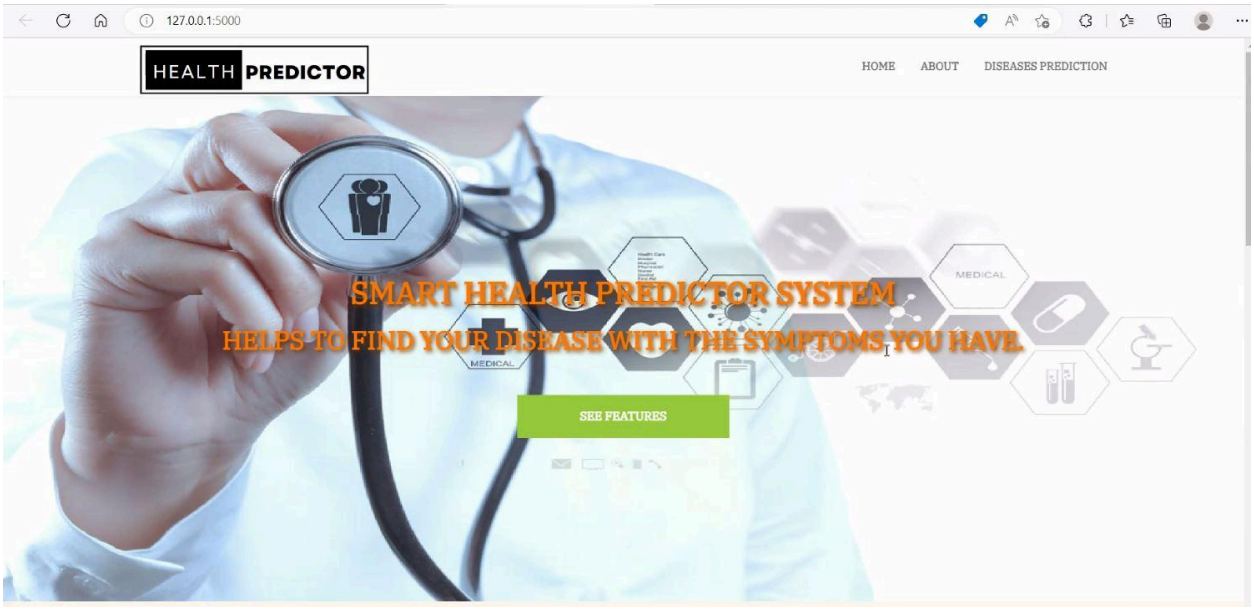
Main Features

Our application allows users or any healthcare patients to navigate through the list of specified symptoms and select some of the symptoms they are facing. Further the user can also input their age for further help in analyzing their health condition. The webpage is interactive and user friendly so that it can be easily used by anyone. After giving every input and selecting the algorithm the application takes less than a few seconds to predict what the health condition might be. The output of the predicted disease is shown on the screen.

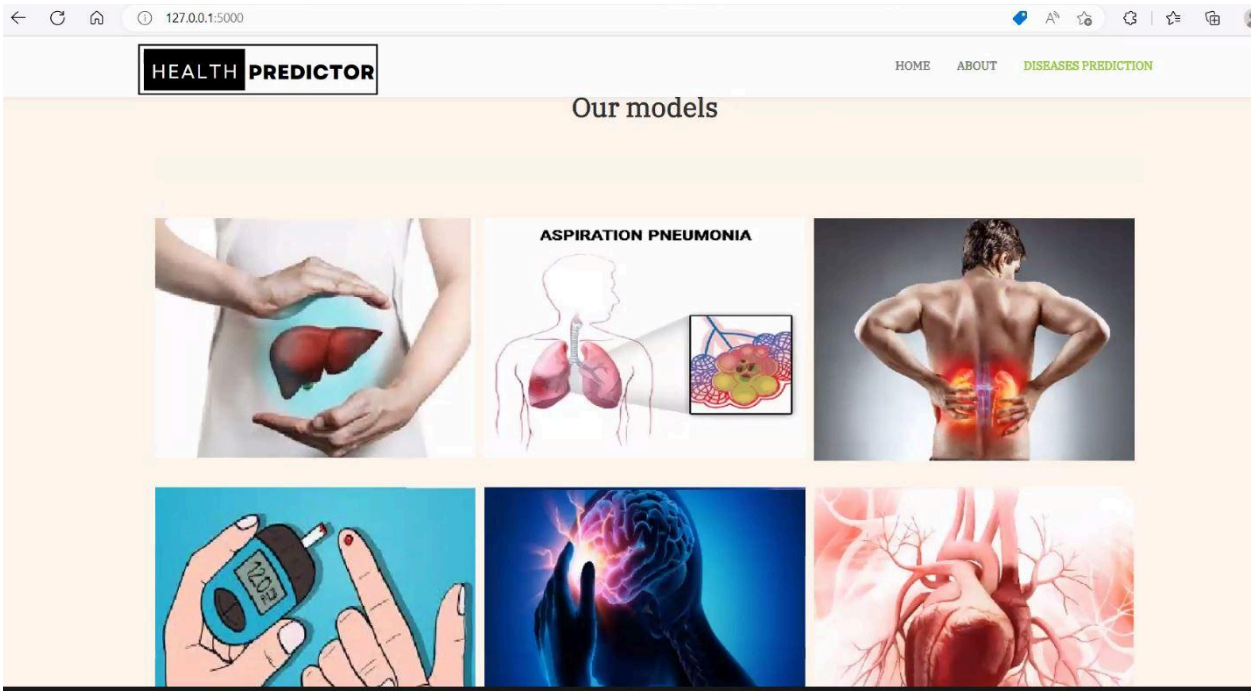
Main Walkthrough

The main feature of our smart healthcare system is to correctly predict the disease or health condition a person might be facing. For that it takes as input the age, and symptoms of the patients and runs a machine learning algorithm in the backend to predict the disease based on the patient's age, and symptoms they input.

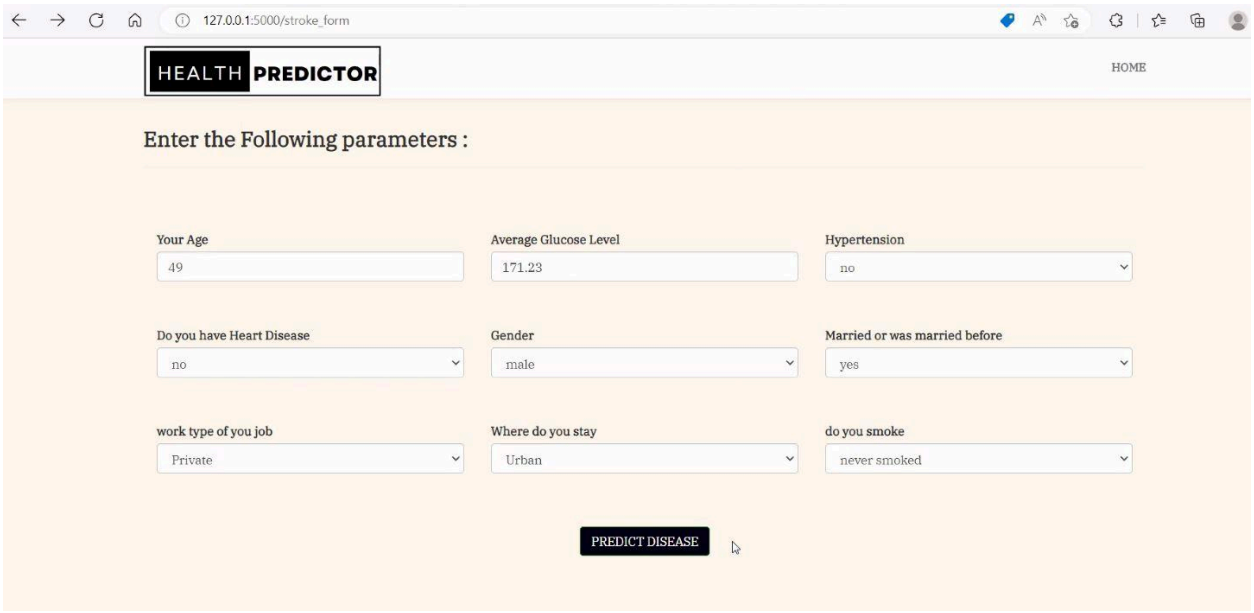
The homepage is as follows:



The following page after the homepage. These are the options the user can choose from depending on the health problem they have:

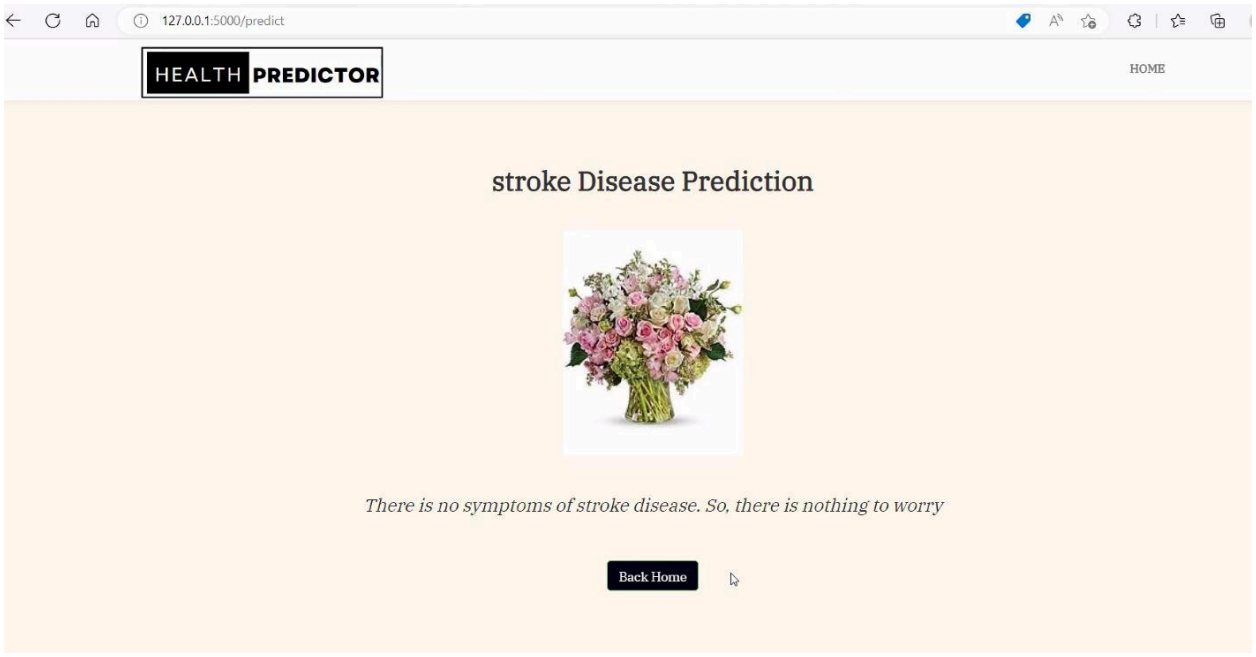


This is the screen asking for user inputs and they can click on the predict disease button to further continue:



The predicted health condition is displayed on the screen after the process is complete.

The predicted results are displayed on the screen:



Alternative Walkthroughs

There are some alternative walkthroughs which the user can take with an upgraded version of this application. Those are manual input fields and report generation.

- Manual input fields

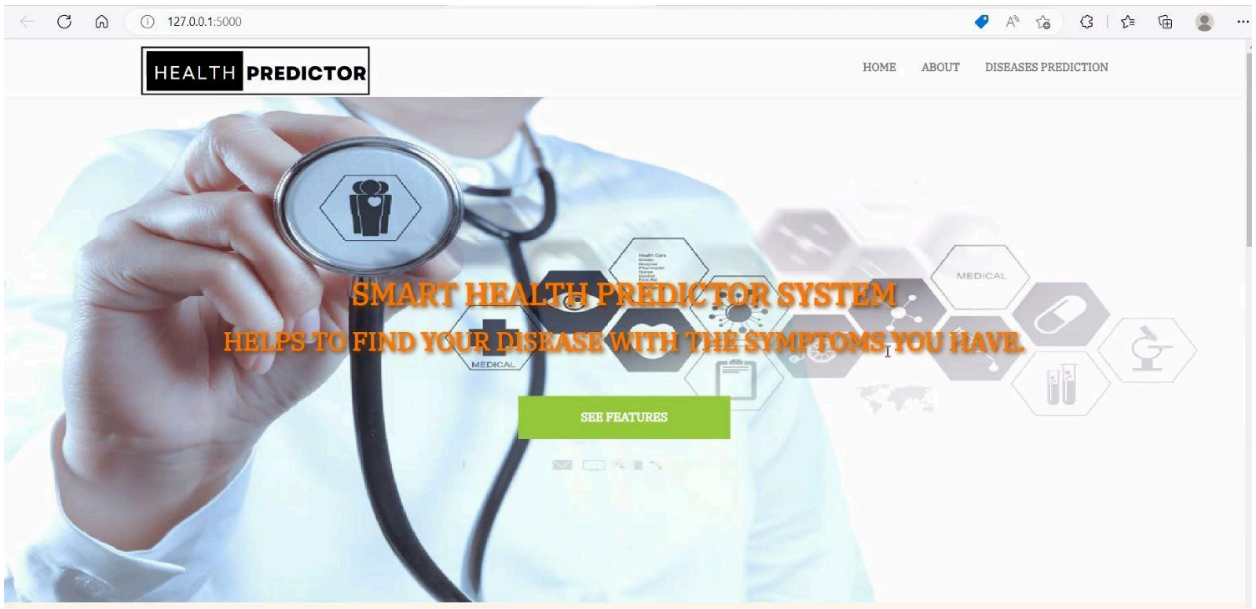
User's can type in the fields instead of using dropbox, when they know the accurate values and condition instead of scrolling down the entire list. Overall it makes user interaction convenient for users using our system and reduces the effort and time required to process.

- Report Generation

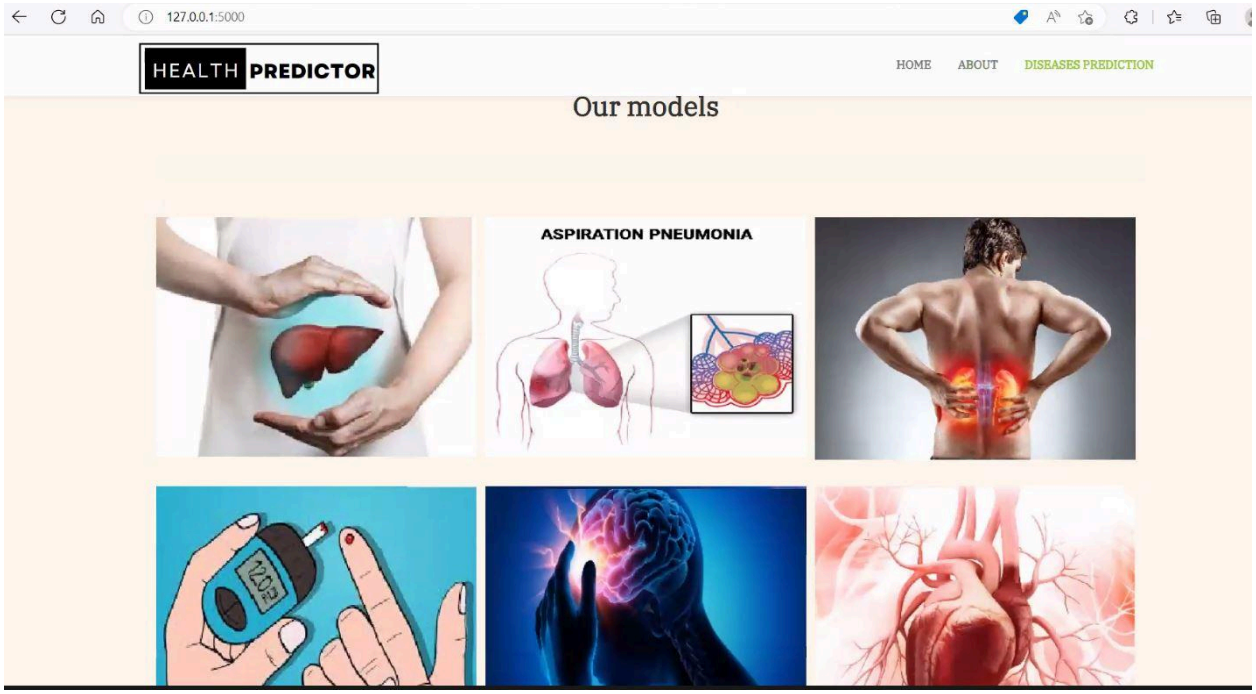
With an updated version of this application, users can not only have their health condition predicted but also a detailed report of the disease can be generated and also shared with the healthcare providers for further clinical diagnosis. This way the time and effort of both the patient and doctors can be reduced.

12. Project Demonstration

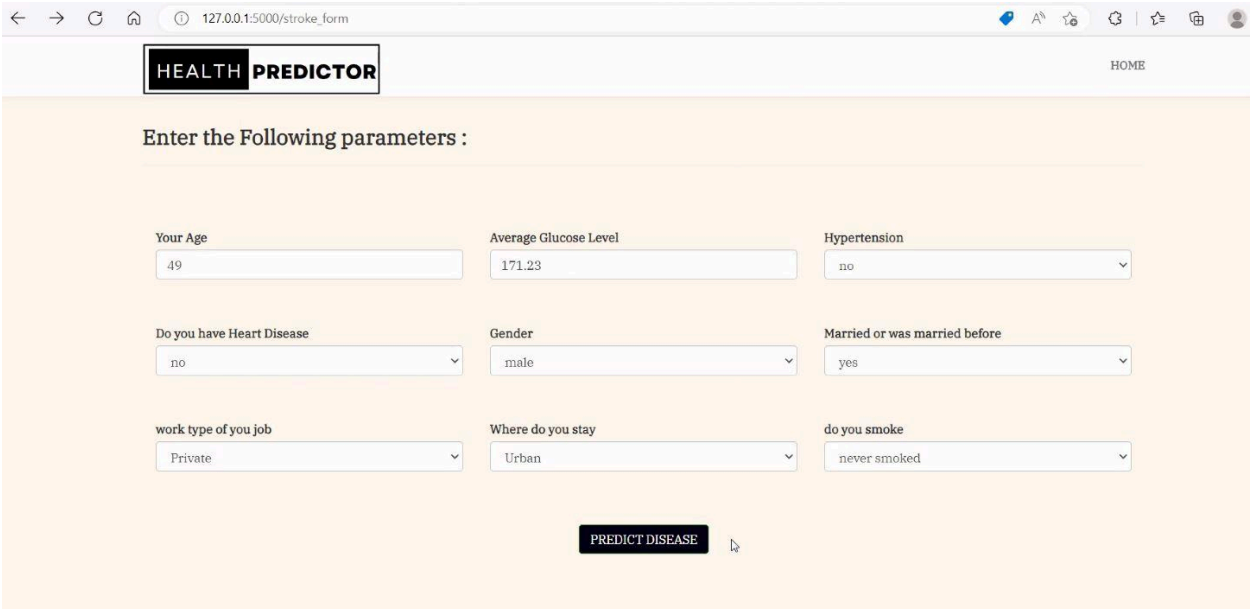
This is the main page of the application :



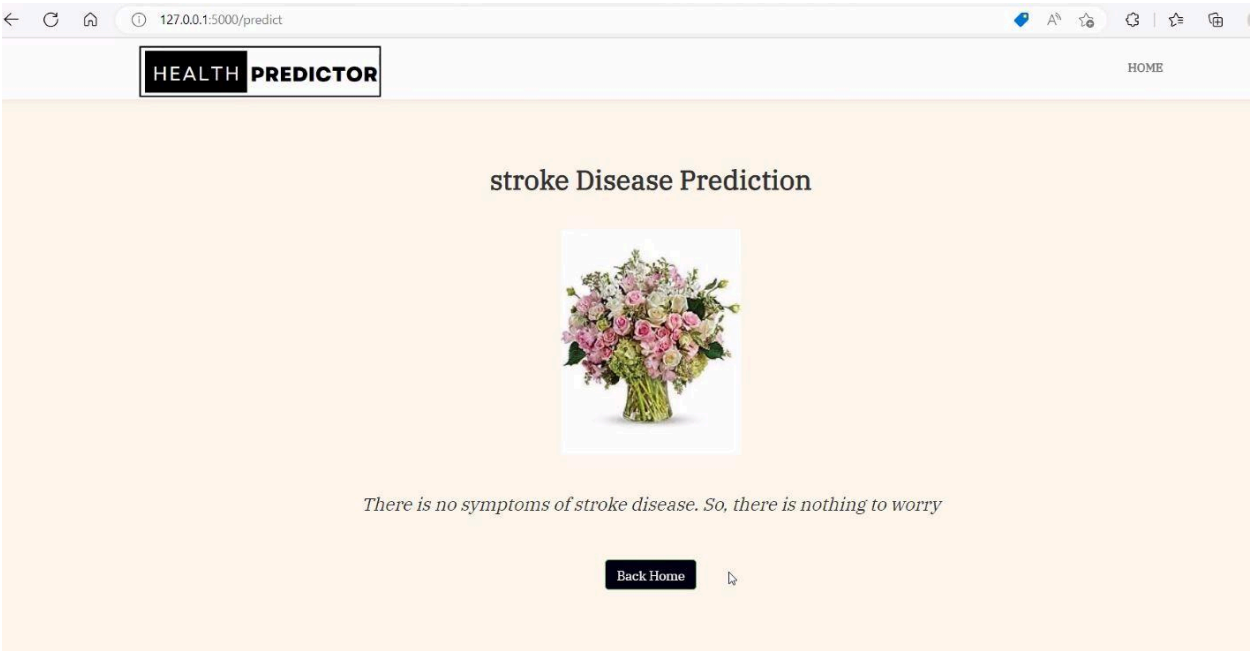
When we click on see features we get something like this:



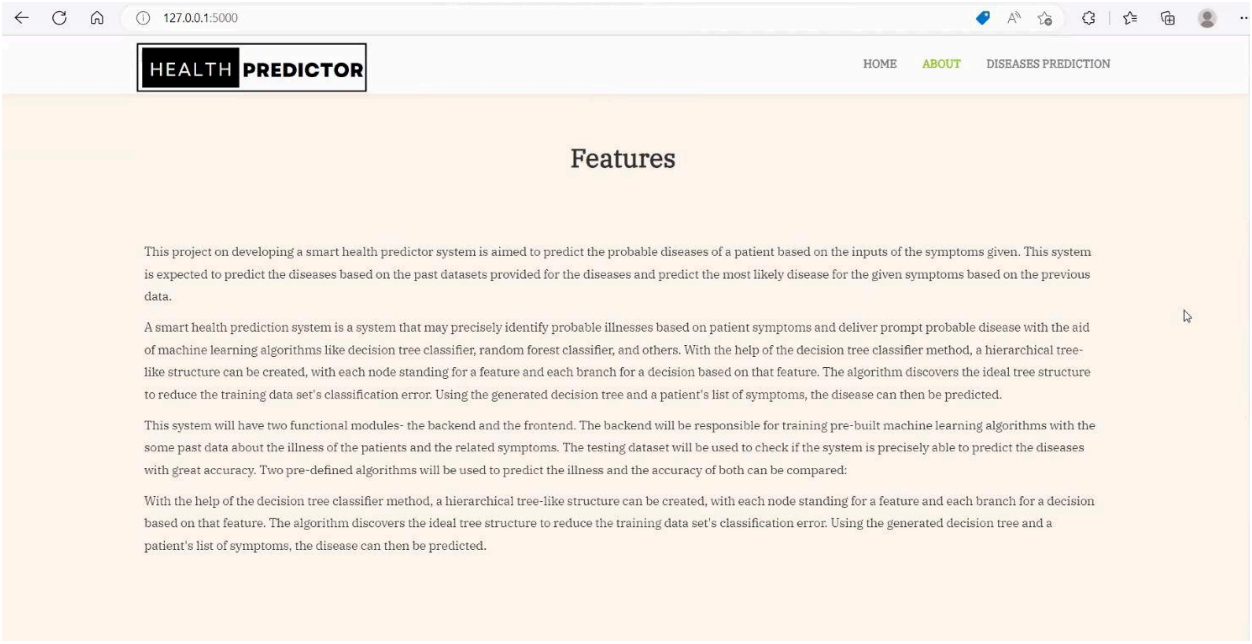
As we can see we can select different types of targeted areas for a particular disease. Now if we click on stroke we get something like this :



Next we input all these values and enter the age of the patient and click on predict disease. We get something like this :



As we can see, this shows the patient is not suffering from any stroke related symptoms and is healthy.
Some additional features of this model as displayed on the main page :



13. Conclusion

The main objective of this project was to develop a system that can assist healthcare providers as well as patients by predicting the health condition/ disease the patient might be suffering from by taking in input their symptoms, age and gender. The UI design has been made interactive and easy to use. Our smart healthcare system achieves good accuracy using the machine learning algorithms in the backend.
For future work, this system can be extended by training it on a larger dataset for better accuracy that can also be deployed for real time use.

References:

1. Machine learning models and datasets taken from :

https://www.kaggle.com/code/therealcyberlord/pneumonia-detection-using-deep-learning
https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset
https://www.kaggle.com/datasets/mathchi/diabetes-data-set
https://www.kaggle.com/code/niteshyadav3103/chronic-kidney-disease-prediction-98-accuracy/notebook
https://www.kaggle.com/datasets/uciml/indian-liver-patient-records
https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

2. Images are taken from google images.