*A Mini Project Report*

**ON**

# MACHINE LEARNING BASED HEALTH CARE SYSTEM

***Submitted in partial fulfillment for the Degree of B.Tech***

**in**

**Artificial Intelligence**

***By***

*RAYABARAPU PRABHATH KUMAR (19911A3543)*

*PASUPUNURI RAKESH ROHAN (19911A3540)*

**Under the guidance of**

Dr.A.Obulesh

HOD, Dept of AI

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE**

**VIDYA JYOTHI INSTITUTE OF TECHNOLOGY**

(An Autonomous Institution)

**Accredited by NAAC & NBA, Approved by AICTE New Delhi & Permanently Affiliated to JNTUH**

2022 – 2023

VIDYA JYOTHI
INSTITUTE OF TECHNOLOGY
AN AUTONOMOUS INSTITUTION

# *CERTIFICATE*

This is to certify that the project report entitled **Machine Learning Based Health Care System** submitted by **Rayabarapu Prabhath Kumar (19911A3543)** and **Pasupunuri Rakesh Rohan (19911A3540)** to Vidya Jyothi Institute of Technology, Hyderabad, in partial fulfillment for the award of the degree of **B.Tech in Artificial Intelligence** a *bonafide* record of project work carried out by us under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

**Signature of HOD & Supervisor**

**Dr.A.Obulesh**

**HOD and Supervisor**

**Department of Artificial Intelligence**

# *DECLARATION*

We declare that this project report titled **Machine Learning Based Health Care System** submitted in partial fulfillment of the degree of **B.Tech in Artificial Intelligence** is a record of original work carried out under the supervision of **Dr.A.Obulesh**, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Rayabarapu Prabhath Kumar (19911A3543)

Pasupunuri Rakesh Rohan (19911A3540)

## *ACKNOWLEDGEMENT*

Any endeavor in the field of development is a person's intensive activity. A successful project is a fruitful culmination of efforts by many people, some directly involved and some others who have quietly encouraged and supported.

We would like to thank **Dr. A.Obulesh**, Department of Artificial Intelligence who has patiently guided and helped me throughout my project.

We wish to express profound gratitude to **Dr A.Obulesh**, Professor and Head of the Department, Artificial Intelligence, for his continuous encouragement to ensure the successful results in all my endeavors.

We wish to express our heart full thanks to the Principal **Dr. A.Padmaja** Madam, whose support was indispensable to me during the course.

We wish to express our heart full thanks to our Beloved Director **Dr E. Sai Baba Reddy**, whose support was indispensable to me during the course.

We wish to express our heart full thanks to our **College Management,** for providing all the Infrastructure and facilities during the course.

By

Rayabarapu Prabhath Kumar (19911A3543)

Pasupunuri Rakesh Rohan (19911A3540)

# *ABSTRACT*

Artificial Intelligence or AI technology is gaining popularity worldwide. It is being used in big organizations as well as our daily lives. Now, taking one step forward, AI is developed to prove useful in the healthcare industry. In simple terms, all the data from diagnosis will be collected and then used for understanding the diseases, so that they can be treated with more success rate.

Nowadays disease can spread wider and easier between living beings. When this happens, we will immediately head to the hospital to get treated or have a checkup done. The problem is, a hospital with more than thousands of patients feels difficult to categorize the patients and to prioritize whom to give first preference to treat. The manual reviews of patient's health records to identify their issues are time consuming.

To solve this we developed a health care system called Health.ai by using machine learning models that can predict various diseases like coronary heart disease, breast cancer, diabetes and also can predict number of calories burnt using a variety of machine learning algorithms, including logistic regression, linear regression, Support Vector Machine, K Nearest Neighbors, and others. By using this health care system, users can check for various disease predictions at a one stop.

Technologies like HTML, CSS, Javascript, Flask, Python, and machine learning were used in the development of this system.

**Keywords:** logistic regression, linear regression, Support Vector Machine, K Nearest Neighbors, heart disease, breast cancer, diabetes, calories

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

Diseases are spreading quickly as a result of modern lifestyle. Our way of life and eating habits have an effect on our health, causing heart disease and other health problems. Artificial intelligence can help patient care providers and intelligent health systems. Healthcare digitization presents a number of opportunities for reducing human error rates, improving clinical outcomes, tracking data over time, etc. Artificial intelligence techniques, from machine learning to deep learning, play a critical role in the improvement of new clinical systems, patient information and records, to diagnose illnesses, find new drugs, identify patient risks, and treat a variety of illnesses. Artificial intelligence has also primarily improved hospital visits and sped up the process of getting patients ready to continue their recovery at home. The use of artificial intelligence (AI) in medical services offers previously unheard-of opportunities to recover patient and clinical group results, cut costs, etc.

To identify diseases like skin, liver, heart, and Alzheimer's that need to be diagnosed early, researchers have used a variety of AI-based techniques including machine and deep learning models. Hence, in our research we used several classification algorithms to identify diseases. Three distinct disease datasets, namely those for diabetes, coronary heart disease, and breast cancer, were used to train the models using a variety of machine learning algorithms. The effectiveness of each model is then calculated and compared. Additionally, we developed a model to predict the number of calories burnt during exercise. It has also been tested on samples of patients data on the three disease classification models to determine whether the patient has that disease or not.

## 1.2 Issues Identified

Considering the personal health issues, the longer wait times frequently make them worse. Patients may have a variety of medical conditions that make it challenging to manually identify and classify the disease. Since patients may not usually be able to go to care meetings because of difficulty in accessing this assistance, it may be best to focus on helping them self-manage their care by providing advice and education as well as the freedom to check their health results whenever they want. Multiple visits to the doctor for different health issues could be expensive and time-consuming.People frequently look up different health-related issues on different websites, which takes time and may not result in accurate predictions. When the disease is not identified early on and no precautions have been taken, unfortunate events can happen.

### 1.3 Objective

To develop a one-stop web application that can accurately predict different health conditions in order to speed up medical procedures, give users the freedom to check their health at any time and from any location, reduce the risk of disease, or detect it early enough to receive the best care.

### 1.4 Motivation for work

As a result of all the advancements and discoveries in the disciplines of AI and ML, doing in-depth research and offering a practical application has become an absolute joy. We believe AI would be extremely beneficial to the healthcare industry. Our goal was to accurately predict whether the user is suffering with the disease or not using a straightforward, user-friendly web application.

# CHAPTER 2
# LITERATURE SURVEY

## 2.1 Introduction

Artificial intelligence (AI) in the healthcare sector is receiving attention from researchers and health professionals. Few previous studies have investigated this topic from a multi-disciplinary perspective, including accounting, business and management, decision sciences and health professions.

The structured literature review with its reliable and replicable research protocol allowed the researchers to extract 288 peer-reviewed papers from Scopus. The authors used qualitative and quantitative variables to analyze authors, journals, keywords, and collaboration networks among researchers.

## 2.2 Existing Model

There are a variety of web applications to improve healthcare.some of them are developed few ML algorithms on various Disease classification datasets like real patients data.There are various web applications for classification problems in health care.They have used the Logistic Regression,SVM,SVC,Random Forest,Decision tree,KNN,Naive Bayes ..etc for supervised Learning Techniques.few of them also uses Fuzzy logic,DL techniques,Ensemble Learning Techniques..etc.

A tool developed by the Houston Method Research Institute developed to detect breast cancer has shown the ability to see mammograms with up to 99% accuracy and give diagnostic information thirty times faster than the average human. Such tools can also reduce the need for biopsies.

There was one more paper and project developed for heart disease prediction Sanath Kapoor, Lekhraj Kasar, Ashutosh Mandole and Dr Jayant Mahajan, “Heart Disease Prediction using Machine Learning and Data Analytics Approach”, 7th International Conference on Computing in Engineering & Technology (ICCET 2022), IEEE Conference, 2022.This paper uses ANN and obtained 93% training accuracy and 84% validation accuracy.There was one more paper published on diabetes detection using Random Forest and achieved accuracy of 82%.

We took a reference of those models built and tried to mimic the behavior using ML models and integrate with flask web applications and those models we built act as endpoint for our web application just like API.

## 2.3 Proposed System

In classification problems we have a lot of ML classifiers which can be used to structure .data in our case like heart disease,diabetes,breast cancer.By collecting the datasets from online and then just following the ML workflow like preprocessing, features extraction,model building,model testing.In case of model building We have trained the machine learning algorithms like SVM,KNN,linear regression, logistic regression decision tree on those collected datasets After building those models we saved them in .pkl files and integrated with flask framework for our web application.we took those files as endpoint creation locally simply takes input and gives output.

## 2.4 Conclusion

We achieved an accuracy of 98% for  Breast Cancer using Logistic regression,87% for Heart Disease using SVM ,73% for Diabetes prediction using SVC and calories prediction of model score 0.96 using Linear regression.

# CHAPTER 3

# METHODOLOGY

## 3.1 Introduction

To implement the models to predict various diseases we have followed the machine learning process as shown in the figure 3.1

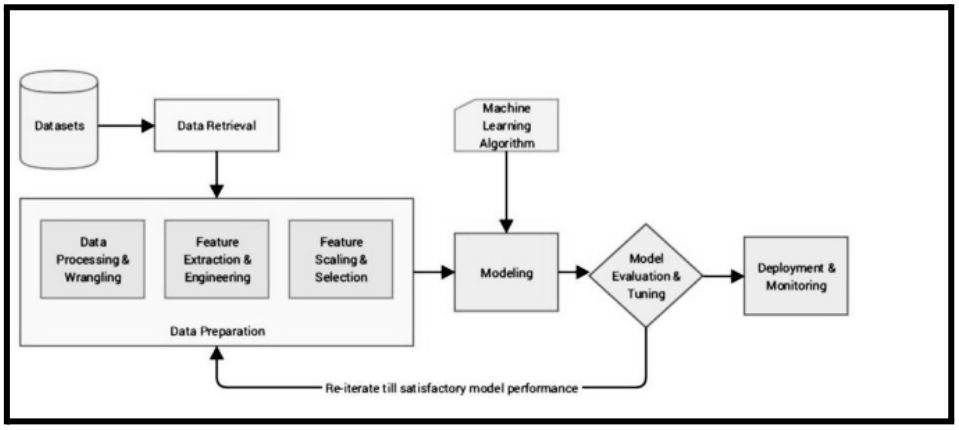


**Fig 3.1:** Machine Learning Process

## 3.2 Machine Learning Process

The first step in the machine learning process is to gather data from a variety of sources, which is then refined. This data will then be fed into ML algorithms based on the problem statement, such as predictive, classification, and other models that are available in the ML world. Let's go over each procedure here one at a time.

### 3.2.1 Data Gathering

The first stage of the machine learning life cycle is data gathering. This step's objective is to locate and collect all data-related issues.

As data can be gathered from a variety of sources, including files, databases, the internet, and mobile devices, we must first identify the various data sources in this step. One of the most crucial phases of the life cycle, it. The effectiveness of the output will depend on the quantity and caliber of the data gathered. The prediction will be more accurate the more data there is. The following tasks are part of this step:

1) Identify different sources of data
2) Collect data
3) Integrate the information from various sources.

By completing the aforementioned task, we obtain a coherent set of data, also known as a dataset which is used in further steps.

### 3.2.2 Data Preparation

In this step we make the collected data more meaningful and accurate. In this step you perform putting all of your data together and randomizing it. By doing this, it is possible to ensure that data are distributed equally and that the learning process is unaffected by the ordering. Cleaning the data to get rid of unnecessary data, missing values, rows and columns, duplicate values, data type conversion, etc. You might even need to change the dataset's rows, columns, or index of rows and columns. Visualize the data to comprehend its structure and the connections between the different variables and classes that are present. Creating two sets out of the cleaned data: a training set and a testing set. Your model learns from a set called the training set. The accuracy of your model is tested using a testing set after training.

### 3.2.3 Modeling

Now the cleaned and prepared data is passed on to the analysis step. This step involves:

1) Selection of analytical techniques
2) Building models
3) Review the result

The aim of this step is to build a machine learning model to analyze the data using various analytical techniques and review the outcome. It starts with the determination of the type of the problems, where we select the machine learning techniques such as Classification, Regression, Cluster analysis, Association, etc. then build the model using prepared data, and evaluate the model.

### 3.2.4 Model Evaluation and tuning

In this phase we are going to test the trained model to evaluate how well it is predicting accurately and to know its strengths and weaknesses. Each model has its own model evaluation mythology, some of the best evaluations are here.

1) Evaluating the Regression Model
    a) Sum of Squared Error (SSE)
    b) Mean Squared Error (MSE)
    c) Root Mean Squared Error (RMSE)
    d) Mean Absolute Error (MAE)
    e) Coefficient of Determination (R2)
    f) Adjusted R2
2) Evaluating Classification Model
    a) Confusion Matrix
    b) Accuracy Score
    c) AUC and ROC

### 3.2.5 Model Deployment

A satisfactory ML-model is deployed when it is integrated into a production environment and output is obtained to support business decisions.

## 3.3 Technologies Used

To develop the Health care system we have used various technologies like python, flask, HTML, CSS and JS. We will understand each technology.

### 3.3.1 Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. Python is used to create web and desktop applications, and some of the most popular web applications like Instagram, YouTube, Spotify all have been developed in Python. You can also develop the next big thing by using Python.

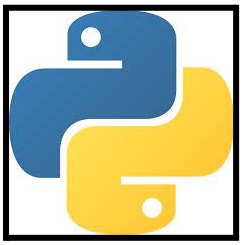

**Fig 3.2** Python

### 3.3.2 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values.

Recommendation engines are a common use case for machine learning. Other popular uses include fraud detection, spam filtering, malware threat detection, business process automation (BPA) and Predictive maintenance.

### 3.3.3 Flask

Flask is a web framework, it's a Python module that lets you develop web applications easily. It has a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features.It does have many cool features like url routing, template engine. It is a WSGI web app framework.

A Web Application Framework or simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on. Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine.


**Fig 3.3:** Flask

Advantages of using Flask are:

- Built-in development server
- Lightweight
- Secure cookies are supported
- Templating using Jinja2
- Request dispatching using REST
- Support for unit testing is built-in

### 3.3.4 HTML

HyperText Markup Language (HTML) is a type of markup language. It is made by the World Wide Web Consortium (W3C). It is used to make webpages. Webpages can include writing, links, pictures, and even sound and video. HTML tells web browsers what webpages should look like. HTML also adds meta information to webpages. Meta information is information about a webpage e.g., the name of the person who created the page. Web browsers usually do not show meta information.


**Fig 3.4:** HTML version 5

### 3.3.5 CSS

Cascading Style Sheets fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. CSS is easy to learn and understood, but it provides powerful control over the presentation of an HTML document.

We use CSS because of the following reasons:

- CSS saves time: You can write CSS once and reuse the same sheet on multiple HTML pages.
- Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.
- Search Engines: CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.
- Superior styles to HTML: CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

### 3.3.6 Javascript

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed. Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behavior to web pages.


**Fig 3.5:** Javascript

# CHAPTER 4
# SYSTEM DESIGN

## 4.1 Introduction

We introduce you to the system architecture ,client server model along with UML diagrams.

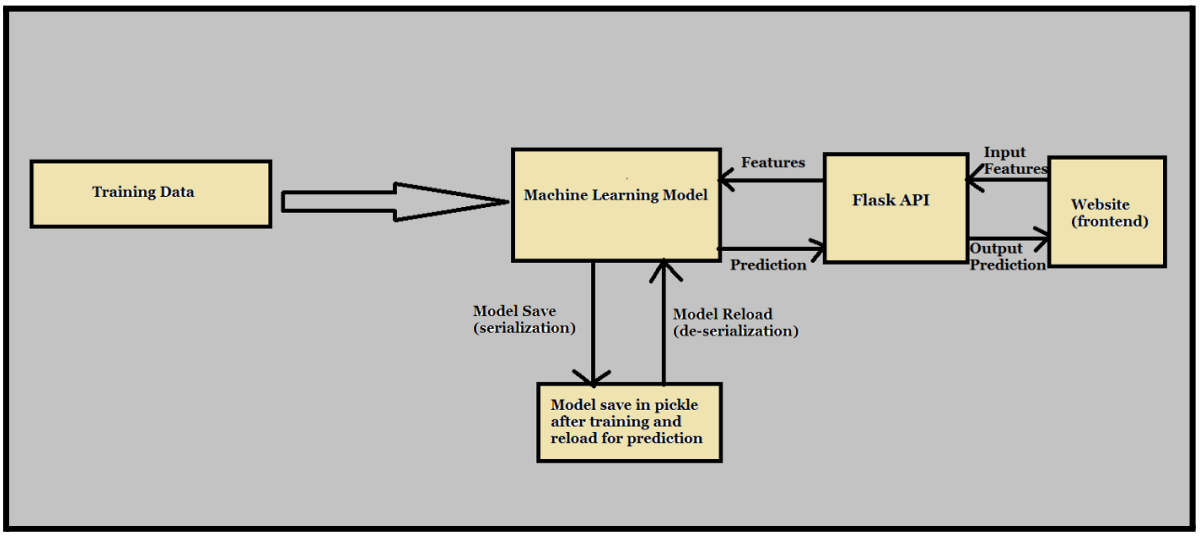


**Fig 4.1:** Project architecture

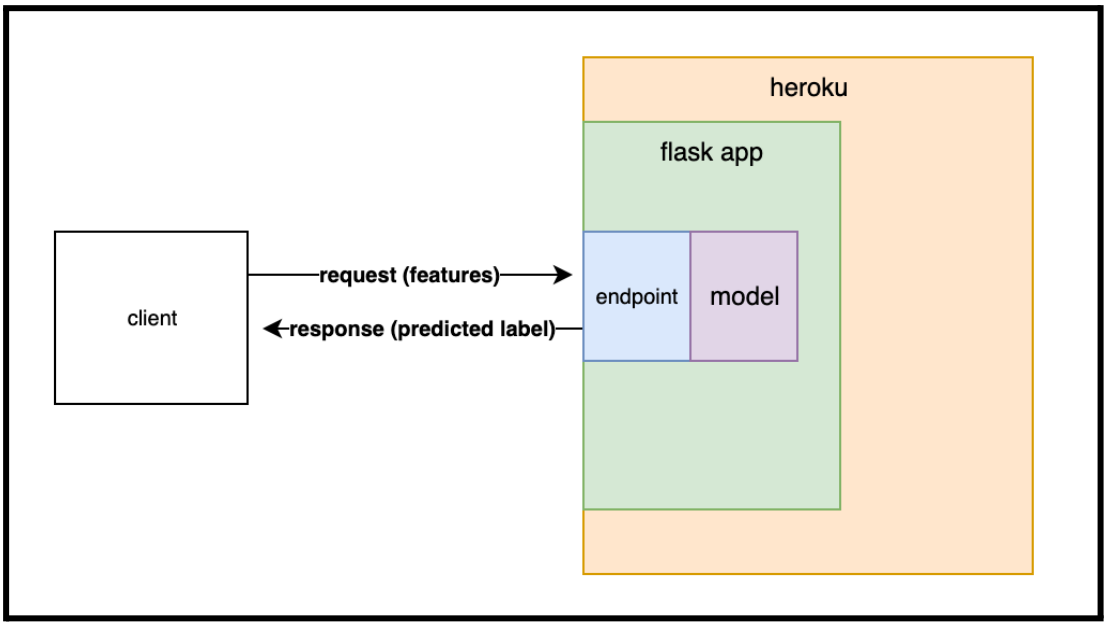


**Fig 4.2:** Client-Server Model

### 4.1.1 Why is the UML diagram used?

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale.

### 4.1.2 Goals of UML

The primary goals in the design of the UML are:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Being independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.

## 4.2 Types of UML diagrams

There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation).

The two most broad categories that encompass all other types are Behavioral UML diagram and Structural UML diagram. As the name suggests, some UML diagrams try to analyze and depict the structure of a system or process, whereas other describe the behavior of the system, its actors, and its building components.

### 4.2.1 Class Diagram

In software engineering, a class diagram in the unified modeling language (uml) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains the switch class contains information.

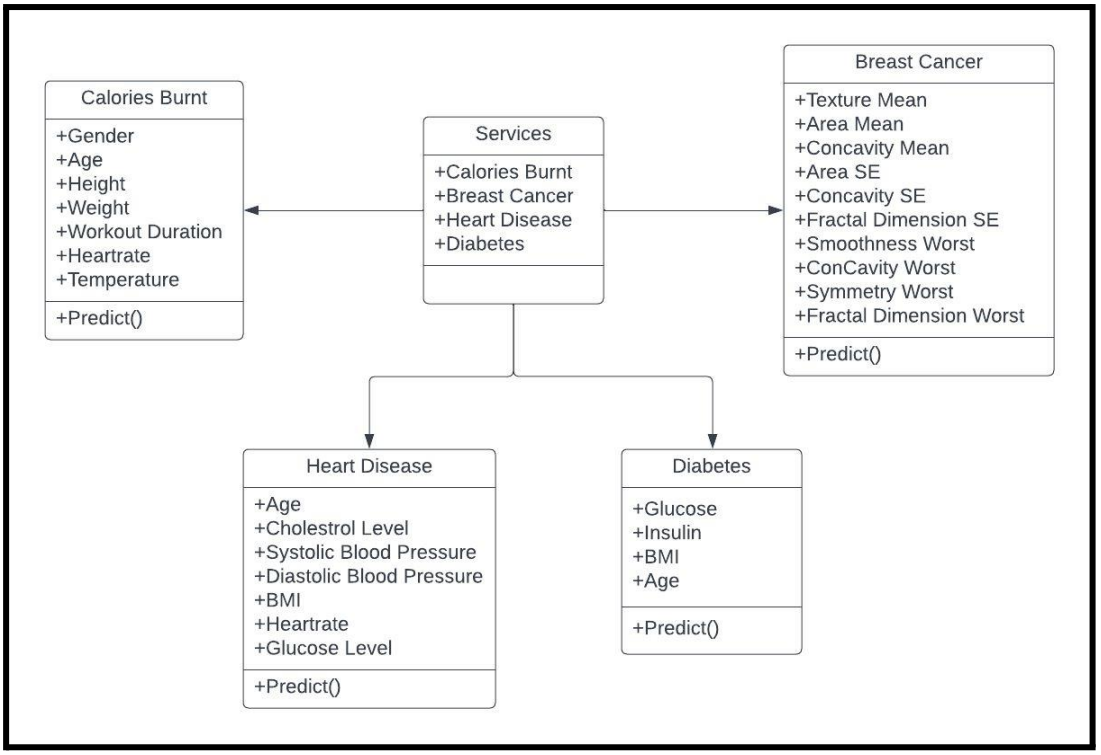


**Fig 4.3:** Class Diagram of Health Care System

### 4.2.2 Use Case Diagram

A use case diagram in the unified modeling language (uml) is a type of behavioral diagram defined by and created from a use case analysis.Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

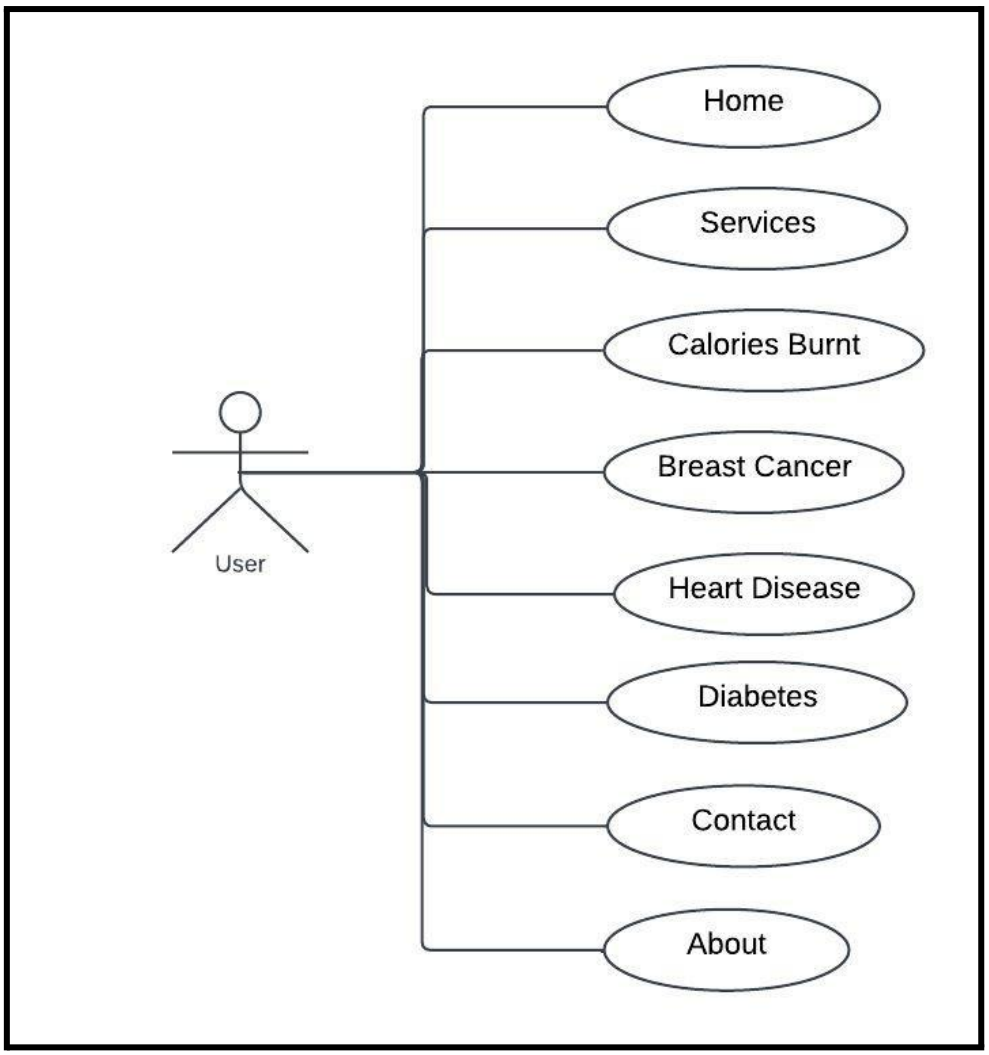


**Fig 4.4:** Use Case Diagram of HCS

### 4.2.3 Activity Diagram

Activity diagrams are for graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the unified modeling language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

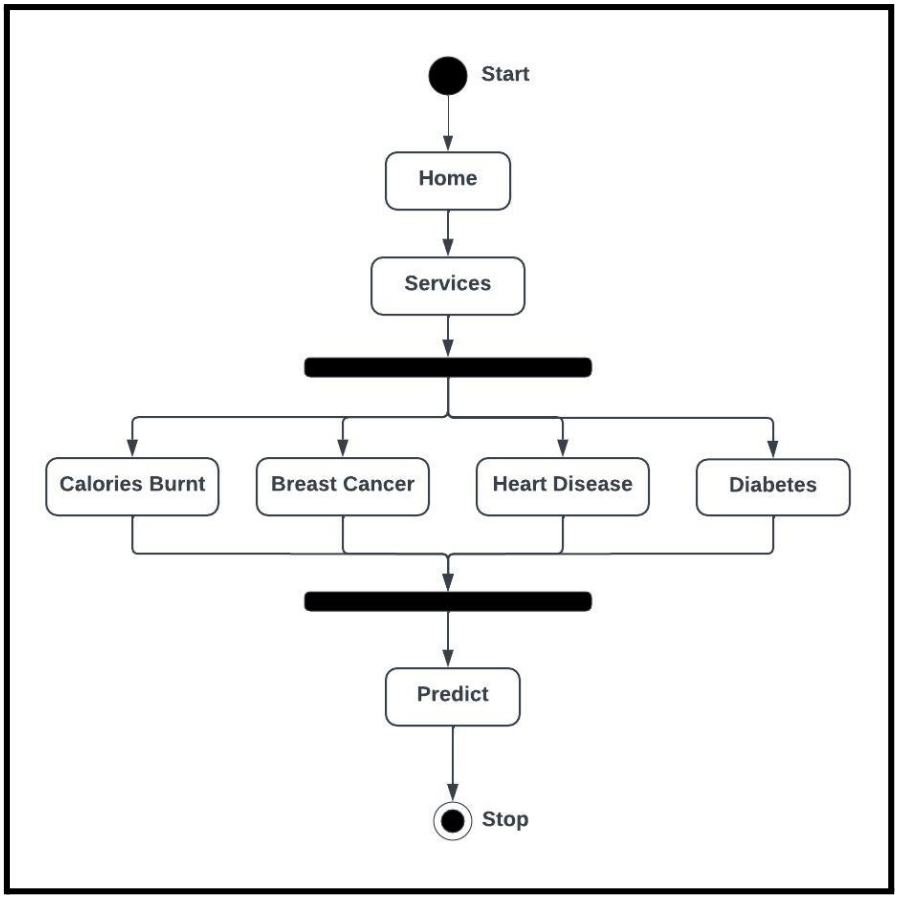


**FIg 4.5:** Activity diagram of HCS

### 4.2.4 Sequence Diagram

A Sequence Diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

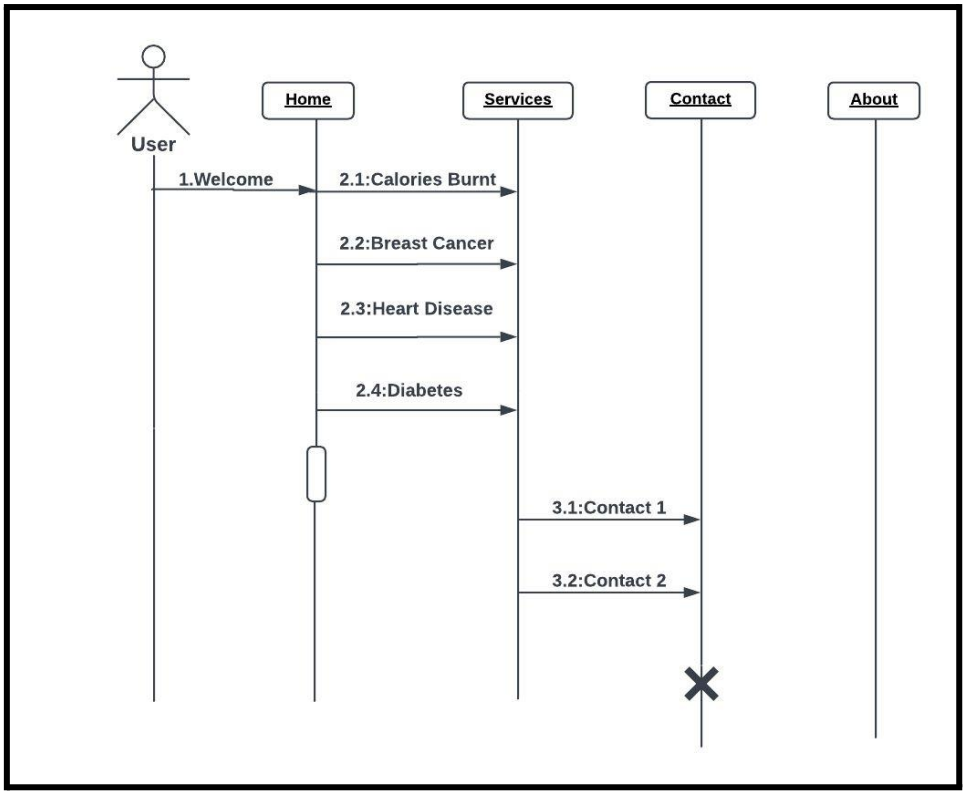


**Fig 4.6:** Sequence diagram of HCS

### 4.3 Conclusion

We have demonstrated various designs and provided you with a thorough explanation of the structure and operation of our healthcare system.

# CHAPTER 5

# IMPLEMENTATION

## 5.1 Introduction

This chapter discusses the implementation of the system. To implement this project you need python and a few packages to be installed in your system. Section 5.2 provides the steps to install python in your system.

## 5.2 Installations

### 5.2.1 Python Installation

1) Visit official python website https://www.python.org/downloads/


**Fig 5.1:** Python website

2) Click the **Download Python 3.9.6 (or the latest version)** button. The following pop-up window titled **Opening python-3.96-amd64.exe** will appear.

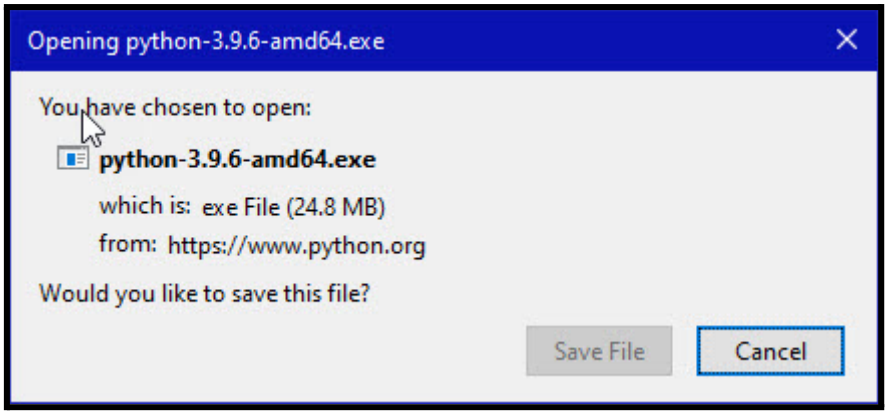


**Fig 5.2:** Download python

Click the **Save File** button. The file named **python-3.9.6-amd64.exe** should start downloading into your standard download folder. This file is about 25 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).

3) Double-click the icon labeling the file **python-3.9.6-amd64.exe**.
A **Python 3.9.6 (64-bit) Setup** pop-up window will appear.

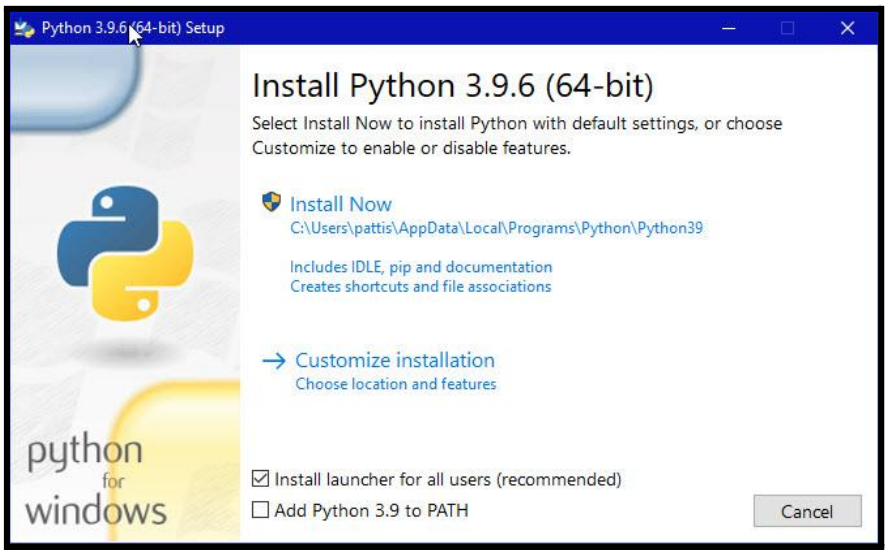


**Fig 5.3** Installing python

Ensure that **both** the **Install launcher for all users (recommended)** and the **Add Python 3.9 to PATH** checkboxes at the bottom are checked: typically only first is checked by default.
If the Python Installer finds an earlier version of Python installed on your computer, the **Install Now** message may instead appear as **Upgrade Now** (and the checkboxes will not appear). Highlight the **Install Now** (or **Upgrade Now**) message, and then click it. When run, a **User Account Control** pop-up window may appear on your screen. I could not capture its image, but it asks, **Do you want to allow this app to make changes to your device?''**

4) Click the **Yes** button.
A new **Python 3.9.6 (64-bit) Setup** pop-up window will appear with a **Setup Progress** message and a progress bar.

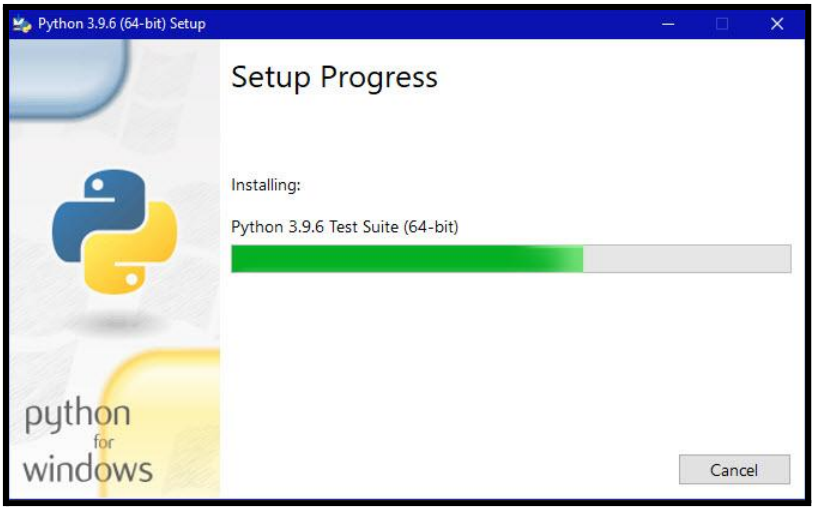


**Fig 5.4:** Python is installing

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new **Python 3.9.6 (64-bit) setup** pop-up window will appear with a **Setup was successfull** message.

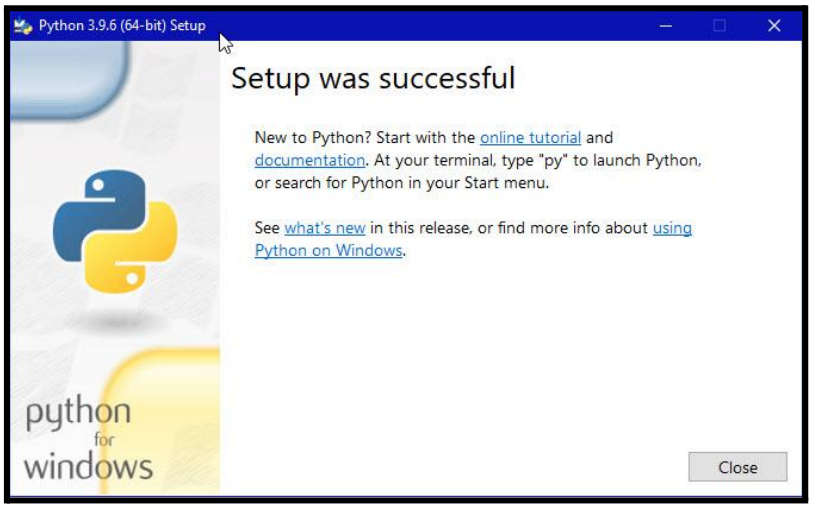


**Fig 5.5:** Successful python installation

5) Click the **Close** button.

6) To try to verify installation, Navigate to the directory **C:\Users\Pattis\AppData\Local\Programs\Python\Python39** (or to whatever directory Python was installed: see the pop-up window for Installing

in Fig 5.3 Double-click the icon/file **python.exe**.
The following pop-up window will appear.

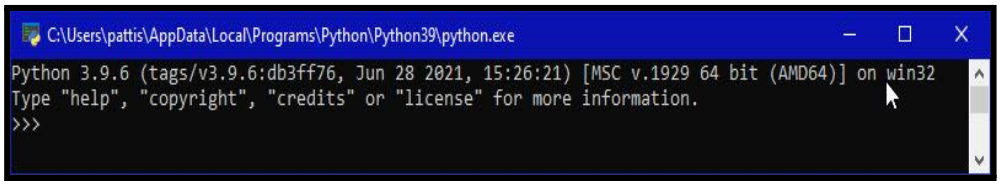


**Fig 5.6:** Running Python

A pop-up window with the title **C:\Users\Pattis\AppData\Local\Programs\Python\Python39\python.exe** appears, and inside the window; on the first line is the text **Python 3.9.6 ...** (notice that it should also say 64 bit). Inside the window, at the bottom left, is the prompt >>>: type **exit()** to this prompt and press **enter** to terminate Python.

### 5.2.2 Scikit Learn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. We have built the project on scikit-learn version 0.24.1. To install scikit-learn run the command

```
pip install -U scikit-learn==0.24.1
```

### 5.2.3 Installing required packages

We have shared all the required packages to be installed in file requirements.txt. To install all the packages mentioned in the requirements.txt file run the below command

```
pip install -r requirements.txt
```

## 5.3 Implementing models

### 5.3.1 Breast Cancer Prediction Model

Breast cancer is cancer that forms in the cells of the breasts. After skin cancer, breast cancer is the most common cancer diagnosed in women in the

United States. Breast cancer can occur in both men and women, but it's far more common in women. Some of the symptoms of breast cancer are

- A breast lump or thickening that feels different from the surrounding tissue
- Change in the size, shape or appearance of a breast
- Changes to the skin over the breast, such as dimpling
- A newly inverted nipple
- Peeling, scaling, crusting or flaking of the pigmented area of skin surrounding the nipple (areola) or breast skin
- Redness or pitting of the skin over your breast, like the skin of an orange

In order to classify whether the person had breast cancer or not we use a Logistic regression model.

Dataset: https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

After data processing we found out the major features in the dataset to predict the disease. Ten real-valued features are computed for each cell nucleus:

1. Radius (mean of distances from center to points on the perimeter)
2. Texture (standard deviation of gray-scale values)
3. Perimeter
4. Area
5. Smoothness (local variation in radius lengths)
6. Compactness (perimeter^2 / area - 1.0)
7. Concavity (severity of concave portions of the contour)
8. Concave Points (number of concave portions of the contour)
9. Symmetry
10. Fractal Dimension

**Logistic Regression:** This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

$$Logit(pi) = 1/(1 + exp(- pi))$$

$$ln(pi/(1 - pi)) = Beta\_0 + Beta\_1 * X\_1 + ... + B\_k * X\_k$$

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds.

After building the model, we have got following classification report as shown in fig 5.7

```
Confusion Matrix :

 [[106   4]
 [  2  59]]
Classification Report :

              precision    recall  f1-score   support

           0       0.98      0.96      0.97       110
           1       0.94      0.97      0.95        61

    accuracy                           0.96       171
   macro avg       0.96      0.97      0.96       171
weighted avg       0.97      0.96      0.97       171
```

**Fig 5.7:** Classification report for Breast cancer prediction model

### 5.3.2 Coronary Heart Disease Prediction Model

Heart disease refers to various types of conditions that can affect heart function. These types include: coronary artery (atherosclerotic) disease that affects the blood supply to the heart, valvular heart disease that affects how the valves function to regulate blood flow in and out of the heart, cardiomyopathies that affect heart muscles, heart rhythm disturbances (arrhythmias) that affect the electrical conduction and congenital heart diseases where the heart has structural problems that develop before birth.

Heart disease is the major cause of morbidity and mortality globally: it accounts for more deaths annually than any other cause. For example an estimated 17.9 million people died from heart diseases in 2016, representing 31% of all global deaths. Over three quarters of these deaths took place in low- and middle-income countries.

Dataset:
https://www.kaggle.com/code/lauriandwu/machine-learning-heart-disease-framingham

After data processing we found out the major features in the dataset to predict the disease. They are

1. Tot Chol: total cholesterol level (Continuous)
2. Sys BP: systolic blood pressure (Continuous)
3. Dia BP: diastolic blood pressure (Continuous)
4. BMI: Body Mass Index (Continuous)
5. Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of a large number of possible values.)

6. Glucose: glucose level (Continuous)

We have implemented various ML algorithms to select the best one and we have found that Support Vector Machine is performing better than others.

**Support Vector Machine:** A support vector machine is a machine learning model that is able to generalize between two different classes if the set of labeled data is provided in the training set to the algorithm. The main function of the SVM is to check for that hyperplane that is able to distinguish between the two classes. There can be many hyperplanes that can do this task but the objective is to find that hyperplane that has the highest margin that means maximum distances between the two classes, so that in future if a new data point comes that is to be classified then it can be classified easily.

| | Accuracy | AUC | F1 score |
|---|---|---|---|
| Logistic regression | 0.654148 | 0.701467 | NaN |
| K-nearest neighbours | 0.838428 | 0.849633 | 0.834378 |
| Decision trees | 0.746725 | 0.757393 | 0.699793 |
| Support vector machine | 0.872489 | 0.935686 | 0.858527 |

**Fig 5.8:** Comparing different models for heart disease prediction.

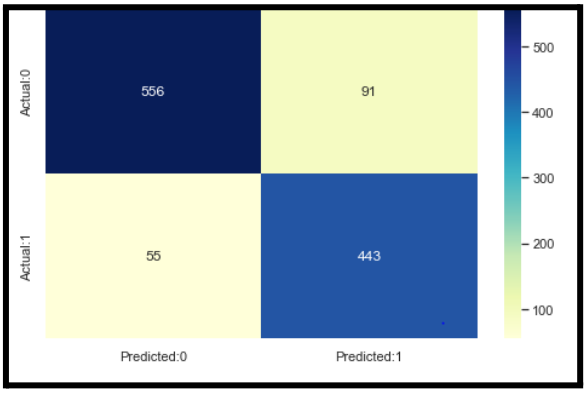


**Fig 5.9:** Confusion matrix for SVM.

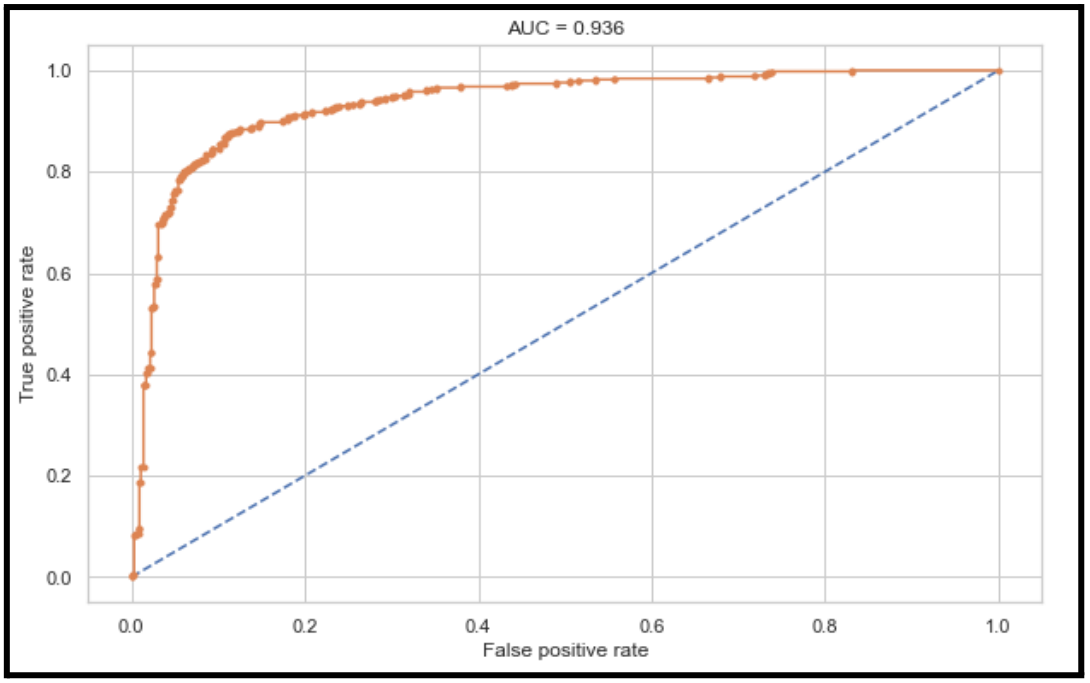


**Fig 5.10:** Area Under Curve diagram for SVM.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.94 | 0.76 | 0.84 | 647 |
| 1 | 0.75 | 0.94 | 0.83 | 498 |
| accuracy | | | 0.84 | 1145 |
| macro avg | 0.85 | 0.85 | 0.84 | 1145 |
| weighted avg | 0.86 | 0.84 | 0.84 | 1145 |

**Fig 5.11:** Confusion report for SVM model

### 5.3.3 Diabetes Prediction Model

This section describes the model implementation to predict diabetes. Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels

Type 1 Diabetes Mellitus is classified by a deficiency of insulin in the blood. The deficiency is caused by the loss of insulin-producing beta cells in the pancreas. Type 1 diabetes is also hereditary from your parents. You are most likely to have type 1 diabetes if any of your parents had it. Type 2 diabetes mellitus is the most common

type of diabetes in people.It is characterized by the inefficiency of body tissues to effectively respond to insulin because of this it may be combined by insulin deficiency.

Dataset: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

After data processing we found out the major features in the dataset to predict the disease. They are

1. Glucose: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
2. Insulin: 2-Hour serum insulin (mu U/ml)
3. Age: Age (years)
4. Outcome: Class variable (0 or 1)

**Support Vector Classifiers:** To predict diabetes, we have chosen Support Vector Machine to build the model. In section 5.3.2 we have discussed SVM. To build a diabetes prediction model we have used SVC, or Support Vector Classifier, is a supervised machine learning algorithm typically used for classification tasks. SVC works by mapping data points to a high-dimensional space and then finding the optimal hyperplane that divides the data into two classes. Here in SVC the kernel is linear.

After implementing the model with SVC, we have achieved accuracy around 73% as shown in fig 5.12

```
from sklearn.svm import SVC
svc = SVC(kernel = 'linear', random_state = 42)
svc.fit(X_train, Y_train)
```

```
SVC(kernel='linear', random_state=42)
```

```
svc.score(X_test, Y_test)
```

```
0.7337662337662337
```

**Fig 5.12:** Diabetes prediction model accuracy

### 5.3.4 Calories Burnt Prediction Model

A calorie is a unit of measurement but it doesn't measure weight or length. A calorie is a unit of energy. When you hear something contains 100 calories, it's a way of describing how much energy your body could get from eating or drinking it.

Dataset:
https://www.kaggle.com/code/ajaysinghpanwar/predicting-calories-burned-using-linear-regression/data

After data processing we found out the major features in the dataset to predict the number of calories burnt. They are

1. Gender:gender of a person 0/1
2. Age : age of the person(in years)
3. Height: Height of a person(in cm)
4. work out duration:how many hours he/she spent on exercise(in hours)
5. Heart rate: speed of a heart beat
6. Body temperature:temperature of a body
7. Calories : how many calories he/she burnt(Target Label)

We can easily understand the dataset by visualizing it. The fig 5.13, 5.14, 5.15. 5.16 shows the distribution of data in the dataset.

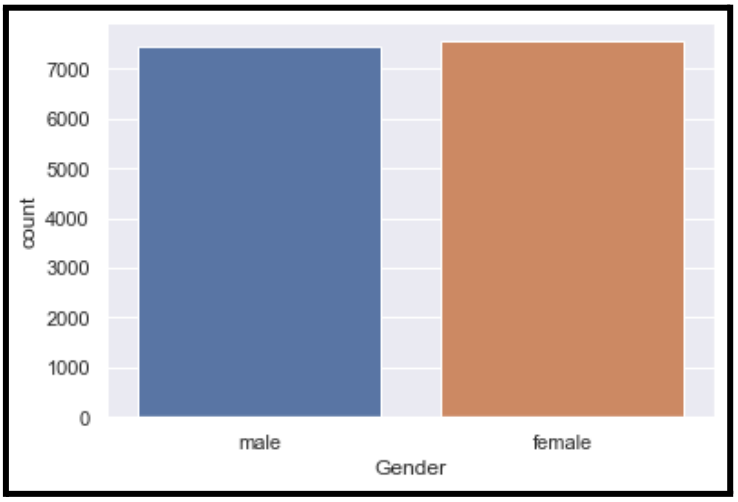


**Fig 5.13:** Gender distribution

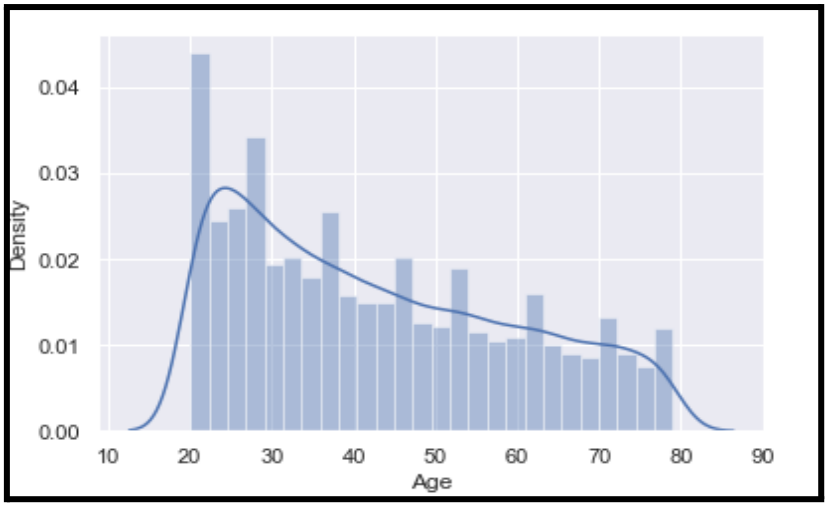


**Fig 5.14:** Age distribution

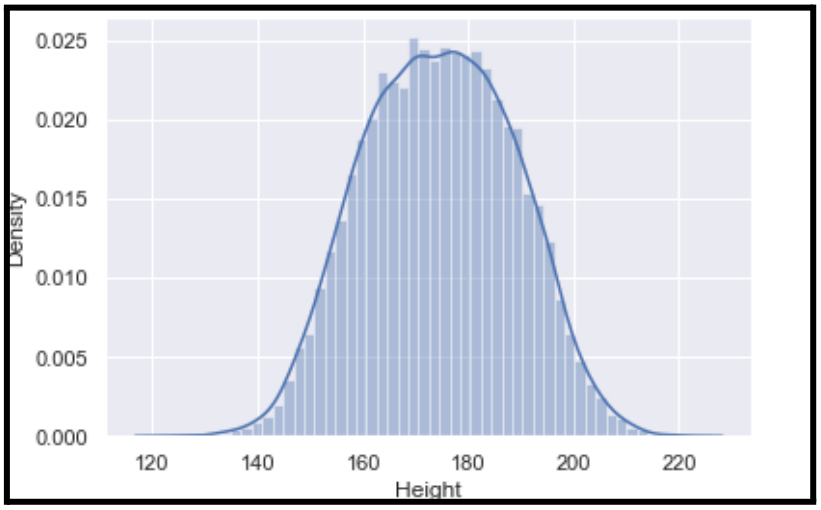


**Fig 5.15:** Height distribution

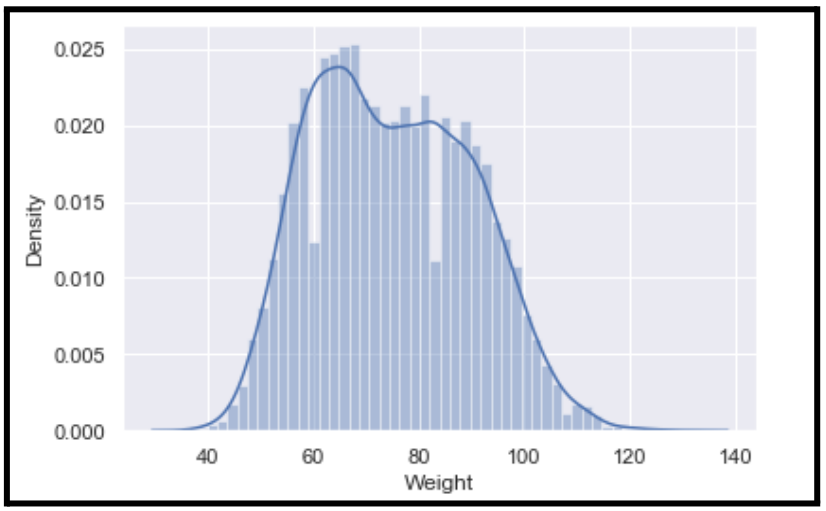


**Fig 5.16:** Weight distribution

**Linear Regression:** A linear regression model describes the relationship between a dependent variable, y, and one or more independent variables, X. The dependent variable is also called the response variable. Independent variables are also called explanatory or predictor variables.

We have achieved 96.5% accuracy in predicting the number of calories burnt.

### 5.4 Building Flask Application

We are going to use flask as the backend of our website. We have written the python script saved in the app.py file where the website paths, models we built previously saved in pickle files can be used. App.py file code is shown below

```
import flask, pickle
import numpy as np
import pandas as pd
import joblib
from flask import Flask, request, render_template ,url_for,
jsonify # Import flask libraries
from sklearn.preprocessing import MinMaxScaler
app = Flask(__name__,template_folder="templates")
#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------

@app.route('/',methods=['GET'])
def index():
    return render_template('index.html') # Render index.html

@app.route('/Home.html')
def home():
    return render_template('Home.html') # Render home.html

#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------

@app.route('/About.html')
def about():
    return render_template('About.html') # Render About.html

#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------
```

```
@app.route('/Contact.html')
def contact():
    return render_template('Contact.html') # Render
Contact.html

#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------

diabdataset = pd.read_csv('Datasets\\diabetes.csv')
diabdataset_X = diabdataset.iloc[:,[1, 2, 5, 7]].values
sc = MinMaxScaler(feature_range = (0,1))
diabdataset_scaled = sc.fit_transform(diabdataset_X)
diabmodel = pickle.load(open('diabetesmodel.pkl', 'rb'))

@app.route('/Diabetes-Prediction.html')
def diabetes():
    return render_template('Diabetes-Prediction.html') #
Render Diabetes-prediction.html
@app.route('/predictdiabetes',methods=['POST'])
def predictdiabetes():
    final_features = [
[
int(request.form["Glucose Level"]),
int(request.form["Insulin"]),
float(request.form["BMI"]),
int(request.form["Age"])
]
]
    print(final_features)
    prediction =
diabmodel.predict(sc.transform(final_features))
    if prediction == 1:
        output = "You have Diabetes, please consult a Doctor"
    elif prediction == 0:
        output = "You don't have Diabetes, you are safe"
    return render_template('Diabetes-Prediction.html',
prediction_text=output)
#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------
```

```
caloriesburntmodel =
pickle.load(open('caloriesburntmodel.pkl', 'rb'))
@app.route('/Calories-Burnt-Prediction.html')
def calories():
    return render_template('Calories-Burnt-Prediction.html')
# Render Calories-Burnt-Prediction.html

@app.route('/predictcaloriesburnt',methods=['POST'])
def predictcaloriesburnt():
    final_features = [
[
int(request.form["Gender"]),
int(request.form["Age"]),
int(request.form["Height"]),
int(request.form["Weight"]),
int(request.form["Duration"]),
int(request.form["Heartrate"]),
int(request.form["Temperature"])
]
    ]
    coloriesburnt =
caloriesburntmodel.predict(final_features)
    return render_template('Calories-Burnt-Prediction.html',
prediction_text="Total calories burnt:
{}".format(coloriesburnt[0]))


#-----------------------------------------------------------
-----------------------------------------------------------
--------------------------

heartdiseasemodel = joblib.load('heartmodel.sav')


@app.route('/Heart-Disease-Prediction.html')
def heart():
    return render_template('/Heart-Disease-Prediction.html')
# Render Heart-Disease-Prediction.html

@app.route('/predictcoronaryheartdisease',methods=['POST'])
def predictcoronaryheartdisease():
```

```
    final_features = [
[
float(request.form["Age"]),
float(request.form["cholesterol level"]),
float(request.form["Systolic blood pressure"]),
float(request.form["Diastolic blood pressure"]),
float(request.form["BMI"]),
float(request.form["Heartrate"]),
float(request.form["Glucose level"])
]
    ]
    prediction = heartdiseasemodel.predict(final_features)
    if prediction == 1:
        return
render_template('Heart-Disease-Prediction.html',
prediction_text="You have coronary heart disease, please
consult a Doctor")
    elif prediction == 0:
        return
render_template('Heart-Disease-Prediction.html',
prediction_text="You don't have coronary heart disease, you
are safe")

#--------------------------------------------------------------
--------------------------------------------------------------
--------------------------

breastcancermodel = pickle.load(open('breastcancermodel.pkl',
'rb'))
breastcancerscaler =
pickle.load(open('breastcancerscaler.pkl', 'rb'))

@app.route('/Breast-Cancer-Prediction.html')
def cancer():
    return render_template('/Breast-Cancer-Prediction.html')
# Render Breast-Cancer-Prediction.html

@app.route('/predictbreastcancer',methods=['POST'])
def predictbreastcancer():
    final_features = [
[
float(request.form["Texture Mean"]),
```

```
float(request.form["Area Mean"]),
float(request.form["Concavity Mean"]),
float(request.form["Area SE"]),
float(request.form["Concavity SE"]),
float(request.form["Fractal Dimension SE"]),
float(request.form["Smoothness Worst"]),
float(request.form["Concavity Worst"]),
float(request.form["Symmetry Worst"]),
float(request.form["Fractal Dimension Worst"])
]
]
    print(final_features)
    final_features =
breastcancerscaler.transform(final_features)
    prediction = breastcancermodel.predict(final_features)
    y_probabilities_test =
breastcancermodel.predict_proba(final_features)
    y_prob_success = y_probabilities_test[:, 1]
    print("final features",final_features)
    print("prediction:",prediction)
    output = round(prediction[0], 2)
    y_prob=round(y_prob_success[0], 3)
    print(output)

    if output == 0:
        return
render_template('Breast-Cancer-Prediction.html',
prediction_text='THE PATIENT IS MORE LIKELY TO HAVE A BENIGN
CANCER WITH PROBABILITY VALUE  {}'.format(y_prob))
    else:
         return
render_template('Breast-Cancer-Prediction.html',
prediction_text='THE PATIENT IS MORE LIKELY TO HAVE A
MALIGNANT CANCER WITH PROBABILITY VALUE  {}'.format(y_prob))

#-------------------------------------------------------------
-------------------------------------------------------------
--------------------------

if(__name__=='__main__'):
    app.run(debug=True)
```

## 5.5 Directory Structure

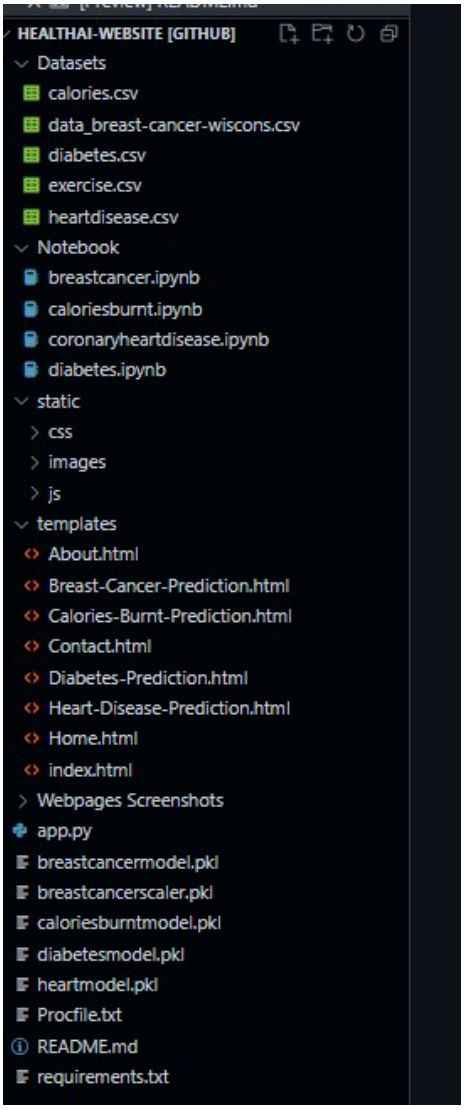


**Fig 5.17:** Project Directory Structure

We have followed a specific directory structure as shown in the fig 6.17 and we have used those files in our project. All the datasets we are using in building the models are placed in a folder called Datasets. Notebook folder contains all the jupyter notebook files in which we have iteratively built the prediction models. All the CSS files, images used in our website and Javascript files are placed in the static folder. Templates folder contains all the html files which are used for website structuring. App.py is the python file used as a flask in our project. all the models we saved in pickle format are kept at the root folder as shown in the figure. When running the project, the package requirements details are saved in requirements.txt file from which we can directly pip install the packages easily.

# CHAPTER 6
# RESULTS

## 6.1 Introduction

A web application is developed to deploy the models and predict various diseases. In this we have created a ‘Home’ page, ‘Breast Cancer Prediction’ page, ‘Coronary Heart Disease Prediction’ page, ‘Diabetes Prediction’ page, ‘Calories Prediction’ page, ‘About’ page, and ‘Contact’ page.

## 6.2 Results

### 6.2.1 Home Page

When you visit the website, the home page is shown first. This page contains detailed visualization of what the project is about.

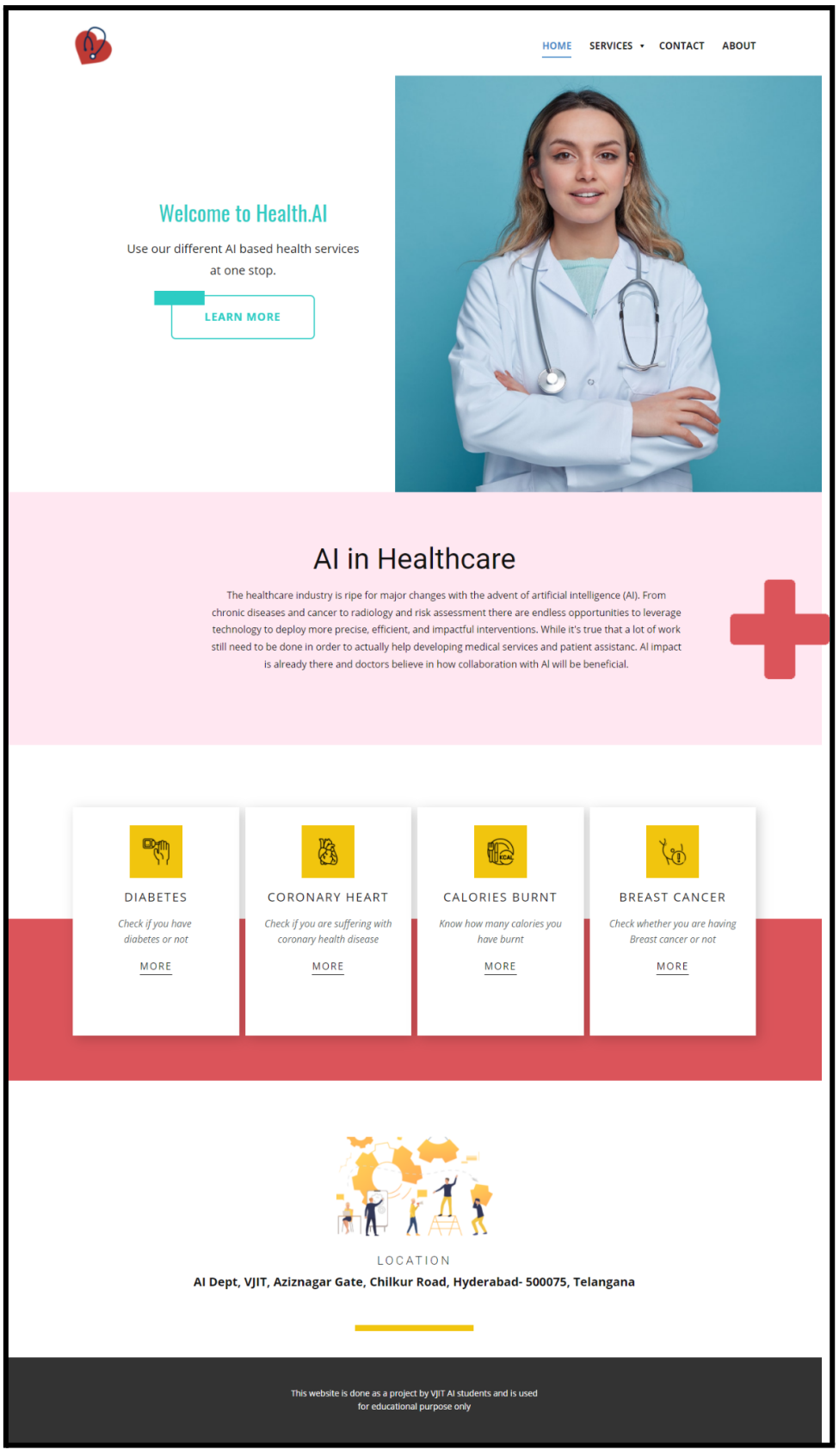


**Fig 6.1:** Home page

### 6.2.2 Coronary Heart Disease page

This page is used to take user input to predict if the user has heart disease or not

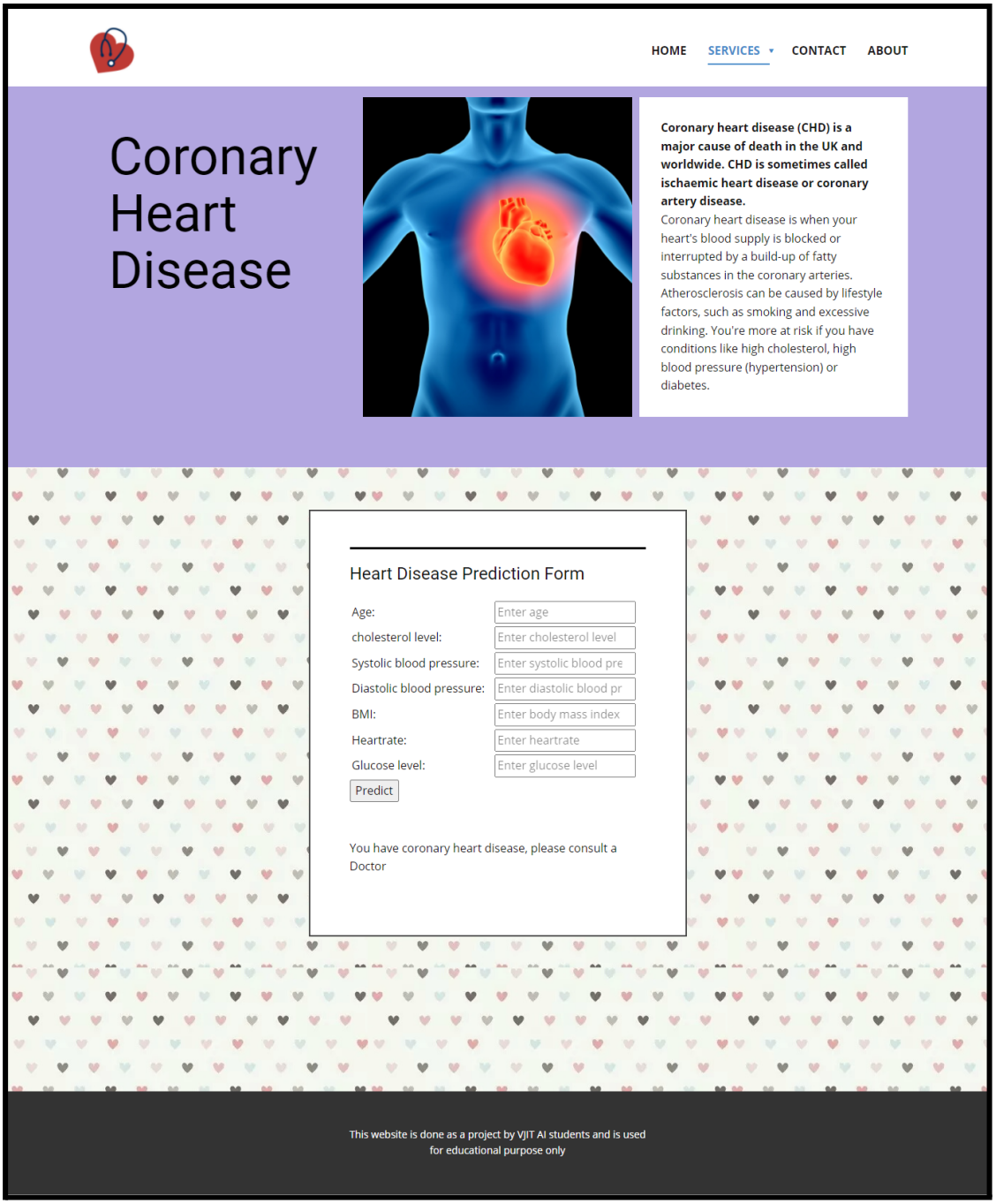


**Fig 6.2:** Coronary heart disease page

### 6.2.3 Breast Cancer Page

This page is used to take user input to predict if the user is suffering from breast cancer or not


**Fig 6.3:** Breast cancer prediction page

### 6.2.4 Diabetes Prediction Page

This page is used to take user input to predict if the user has diabetes or not

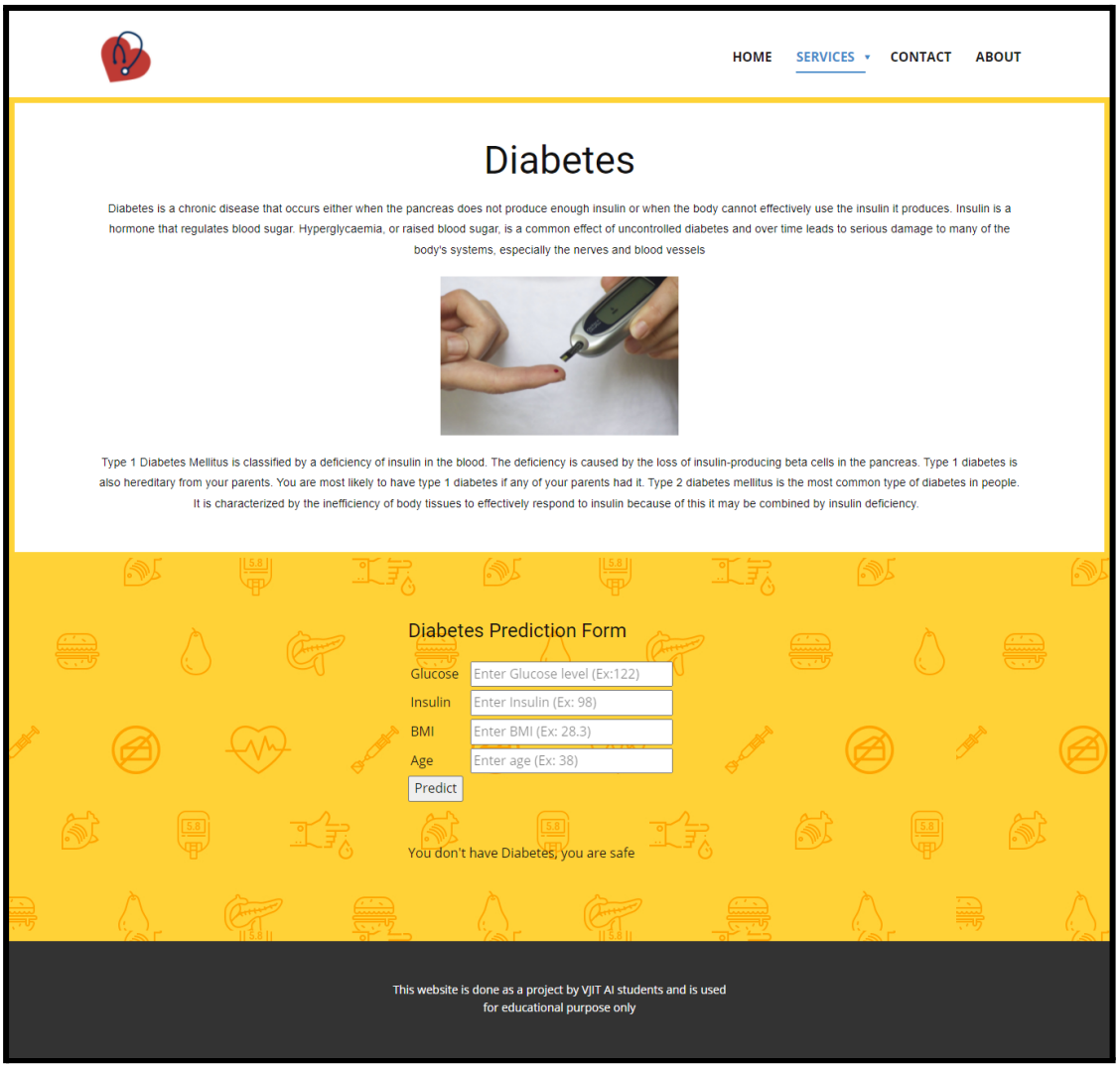


**Fig 6.4:** Diabetes Prediction page

### 6.2.5 Calories Burnt Prediction Page

This page is used to take user input to predict the number of calories he/she has burnt by doing the workout.

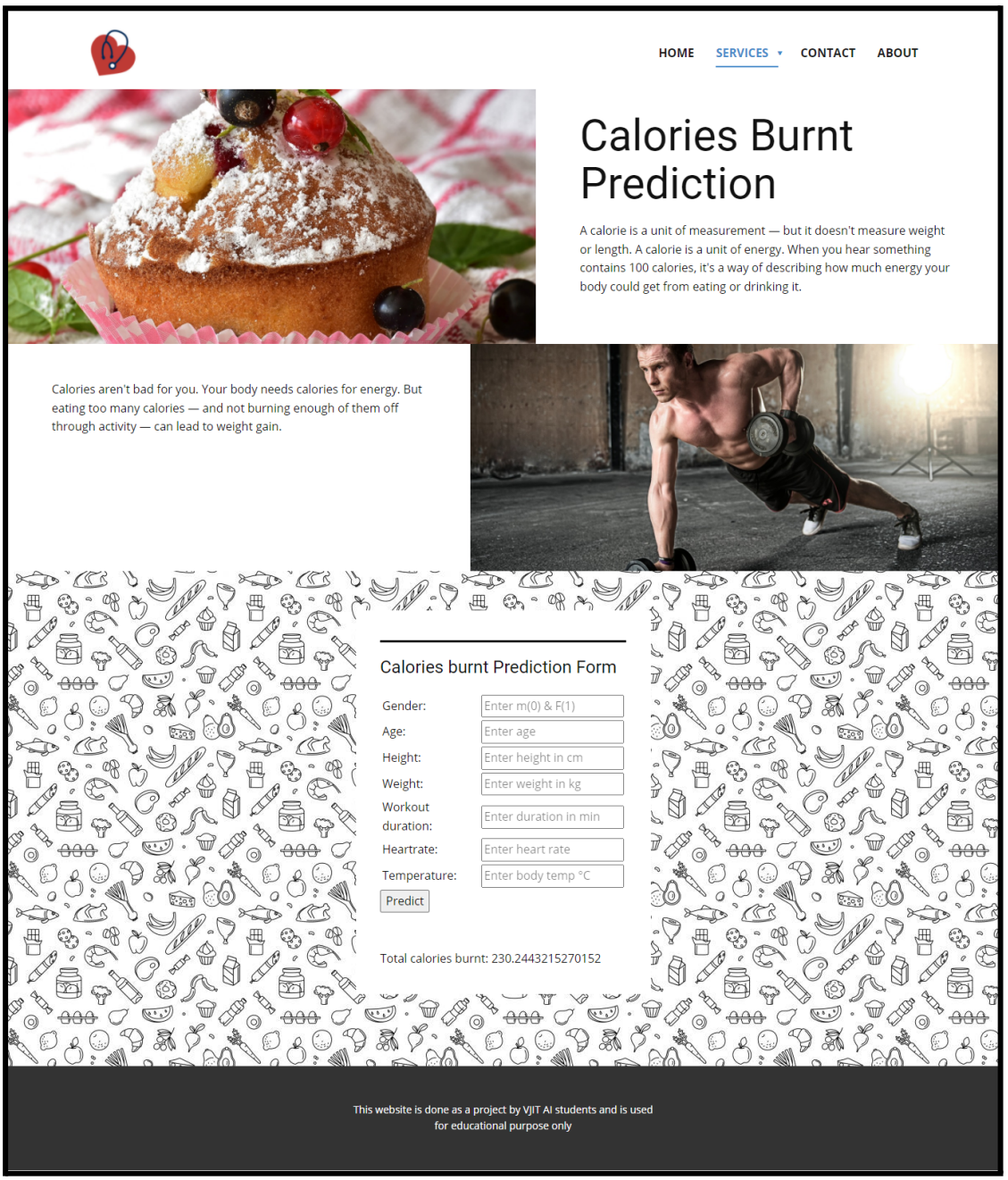


**Fig 6.5:** Calories burnt prediction page

### 6.2.6 Contact Page

This page shows the contact details so that the user can connect to them.

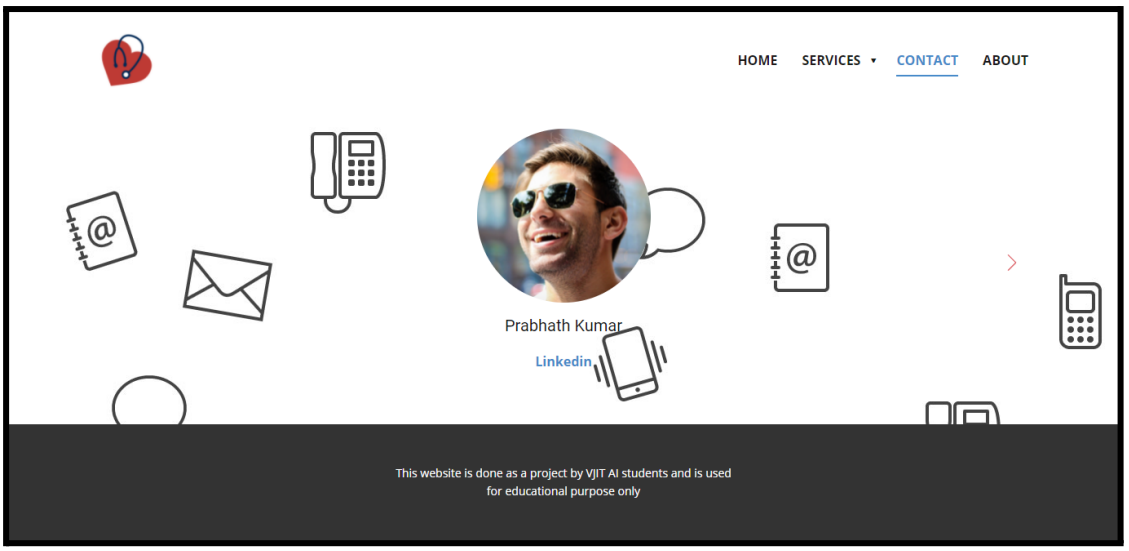


**Fig 6.6:** Contact page

### 6.2.7 About Page

This page tells about the project and website purpose.

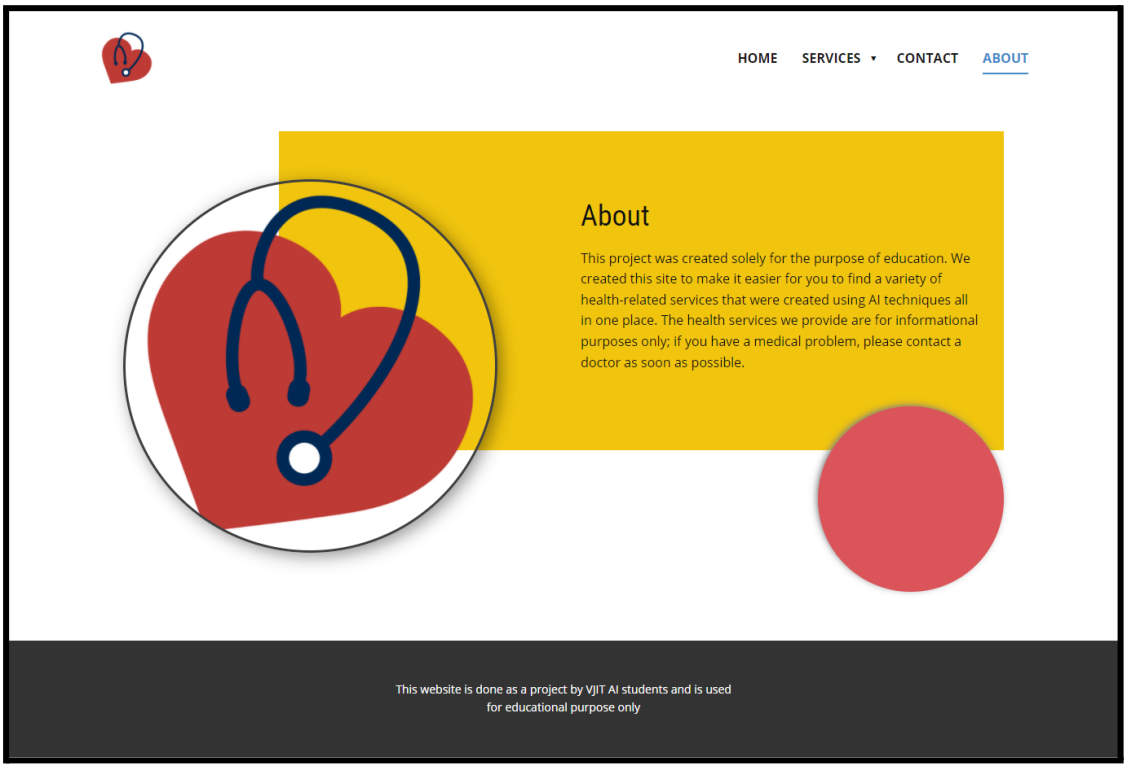


**Fig 6.7:** About page

## 6.3 Conclusion

This chapter includes screenshots of our whole web application as well as instructions for taking user input for predicting the disease..

# CHAPTER 7
# CONCLUSIONS

## 7.1 Conclusion:

Finally we conclude that we provide four health care services through which they can assist their health condition from our website .We build four Machine Learning models for four Services and treat them as API endpoints for our web application.we completed the project by using flask web framework python along with frontend design HTML,CSS,Javascript.

## 7.2 Future Scope:

AI in Healthcare has a lot of future scope like everyone will be getting a smart health assistant without the home/family doctor.

In terms of our project scope or expansion we can add databases and build more complex and accurate classification models with the help of ensemble learning,deep learning techniques and increase the number of services because here in our project we offer only four services.

# CHAPTER 8
# REFERENCES

## 8.1 References

We took the reference of a few research papers published by IEEE.

1. Diabetes Detection using Machine Learning classification Methods
2. Heart Disease Prediction using Machine Learning and Data Analytics Approach
3. A tool developed by the Houston Method Research Institute developed to detect breast cancer has shown the ability to see mammograms with up to 99% accuracy

### 8.2 Websites

https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01488-9

https://ieeexplore.ieee.org/document/9491788

https://www.wired.co.uk/article/cancer-risk-ai-mammograms

https://www.projectpro.io/article/healthcare-machine-learning-projects-with-source-code/508

https://breast-cancer-research.biomedcentral.com/articles/10.1186/s13058-019-1158-4

https://ieeexplore.ieee.org/document/9800258