

# **Heart Disease Prediction Website**

## **A PROJECT REPORT**

**Submitted By**

**ABHINAV CHOUDHARY**

**University Roll No 2100290140002**

**VEDANT GOTRA**

**University Roll No 2100290140143**

**NIKHIL KAUSHIK**

**University Roll No 2100290140096**

**AKSHAY CHAUDHARY**

**University Roll No 2100290140016**

**Submitted in partial fulfillment of the  
Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**Under the Supervision of**

**Under the Supervision of Dr. Vipin Kumar Associate Professor**

**Mr. Ankit Verma Assistant Professor**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**February-2023**

## **CERTIFICATE**

Certified that **Abhinav Choudhary (Roll No 2100290140002)** **Vedant Gotra (Roll No 2100290140143)** **Nikhil Kaushik (Roll No 2100290140096)** **Akshay Chaudhary (Roll No 2100290140016)** have carried out the project work having “**Employee Management System**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**ABHINAV CHOUDHARY**  
**University Roll No 2100290140002**

**VEDANT GOTRA**  
**University Roll No 2100290140143**

**NIKHIL KAUSHIK**  
**University Roll No 2100290140096**

**AKSHAY CHAUDHARY**  
**University Roll No 2100290140016**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Date:**

**Dr. AMIT KUMAR GUPTA**  
**Associate Professor**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner**

**Signature of External Examiner**

**Dr. Arun Tripathi**  
**Head, Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

## **ABSTRACT**

In recent times, Heart Disease prediction is one of the most complicated tasks in medical field. In the modern era, approximately one person dies per minute due to heart disease. Data science plays a crucial role in processing huge amount of data in the field of healthcare. As heart disease prediction is a complex task, there is a need to automate the prediction process to avoid risks associated with it and alert the patient well in advance. This paper makes use of heart disease dataset available in UCI machine learning repository. The proposed work predicts the chances of Heart Disease and classifies patient's risk level by implementing different data mining techniques such as Naive Bayes, Decision Tree, Logistic Regression and Random Forest. Thus, this paper presents a comparative study by analysing the performance of different machine learning algorithms. The trial results verify that Naïve Bayes algorithm has achieved the highest accuracy of 82% compared to other ML algorithms implemented.

**Keywords—** Decision Tree, Naive Bayes, Logistic Regression, Random Forest, Heart Disease Prediction

## **ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Amit Kumar Gupta** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Tripathi Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**ABHINAV CHOUDHARY**  
**University Roll No 2100290140002**

**VEDANT GOTRA**  
**University Roll No 2100290140143**

**NIKHIL KAUSHIK**  
**University Roll No 2100290140096**

**AKSHAY CHAUDHARY**  
**University Roll No 2100290140016**

# **TABLE OF CONTENTS**

Certificate  
Abstract  
Acknowledgement  
Table of Contents

## **CHAPTER 1 : INTRODUCTION**

1.1 Project Description

## **CHAPTER 2 : MODEL**

2.1 Model Used

## **CHAPTER 3 : PROJECT REQUIREMENTS**

3.1 Technologies Used

## **CHAPTER 4 : REQUIREMENT SPECIFICATIONS**

4.1 Hardware Requirements

4.2 Software Requirements

## **CHAPTER 5 : ANALYSIS**

5.1 Feasibility Study

## **CHAPTER 6 : DIAGRAM**

6.1 Gantt Chart

6.2 Data Flow Diagram

6.3 E-R Diagram

6.4 Use Case Diagram

## **CHAPTER 7 : CODE**

## **CHAPTER 8 : Output**

8.1 Homepage of Website

8.2 Prediction and User Input Form page

## **CHAPTER 9 : CONCLUSION**

## **CHAPTER 10 : LITRATURE REVIEW**

## **CHAPTER 11 : REFERENCE**

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT DESCRIPTION

**Heart disease** describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, heart rhythm problems (arrhythmias) and heart defects you're born with (congenital heart defects), among others.

The term "heart disease" is often used interchangeably with the term "cardiovascular disease". Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease.

Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of clinical data analysis. The amount of data in the healthcare industry is huge. Data mining turns the large collection of raw healthcare data into information that can help to make informed decisions and predictions.

Machine learning (ML) proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry.

In this Project, I will be applying Machine Learning approaches (and eventually comparing them) for classifying whether a person is suffering from heart disease or not, using one of the most used dataset from Kaggle.

# CHAPTER 2

## MODEL

### 2.1 Model used: Incremental Model

The model we have used is Incremental Model. In this model, first of all the existing system is observed, then customer requirements are taken in consideration then planning, modelling, construction and finally deployment and again adding the new system if asked by the customer to do so.

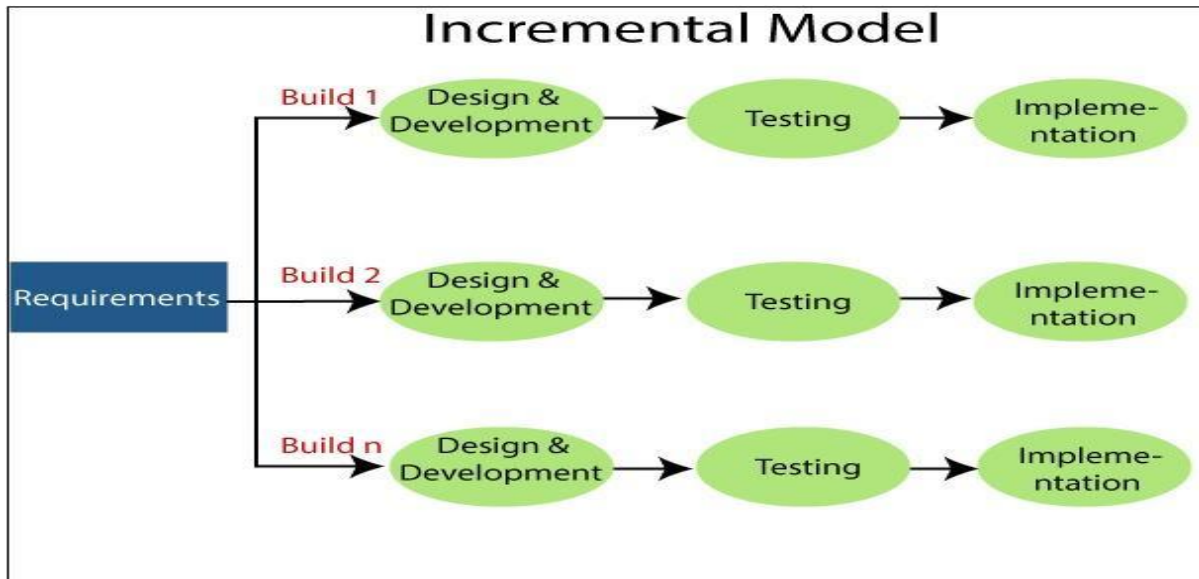


Fig. 1: Incremental Model

Incremental Model is a software development process where requirements are divided into several stand-alone software development modules. In this project the first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments.

## **CHAPTER 3**

### **PROJECT REQUIREMENTS**

#### **2.1 TECHNOLOGIES USED**

##### **XAMPP Server**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

APPLICATION	: Xampp Server
DESIGNING	: React ,Tailswind CSS
BACKEND	: Python Flask,MySql



## **CHAPTER 4**

### **SOFTWARE REQUIREMENT SPECIFICATION**

#### **4.1 GENERAL DESCRIPTION**

This combined aggregation of information and workplace activity constructs a general, specific program or aim which is to be executed or produced within the workplace while working with others as a squad. The history of coaction began many centuries ago, long before the B.C. or A.D. epochs, where at least two persons had to pass on in the attempt of finishing a undertaking, undertaking, or written papers. Therefore, coaction is non a new term, but an enhanced and improved one in the professional workplace.

#### **4.2 PROBLEM STATEMENT**

The problem occurred before having computerized system includes:

- Seeking for the help to play this traditional mode.
- Excessive use of Paper for maintaining register and updating data.
- More chance of Unfairness while giving marks due to biasness.

#### **4.3 SYSTEM OBJECTIVES**

##### **Improvement in control and performance**

The system is developed to cope up with the current issues and problems of forgetting the traditional mechanism. The system identify who is accessing the profile and the data/information will be updated on the portal. To declare the Project and performance of the employee and details.

##### **Save cost**

The existing system is based on the pen paper mode and several in the digital mode but is not secured and efficient to work.

##### **Save Time**

People at any location will able to perform or know there seniors subordinate team and there uniqueness etc. by registering or Logging in the Portal.

#### **4.4 Requirement Specification**

The application requirement specification is produced at the analysis task. The function and performance allocated to application as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints.

#### **4.5 Functional Requirements**

##### **Admin Analyse The Feedback**

The application will show the feedback only to the admin with help of the password.

##### **Predict the Heart Disesease**

User can Predict He Having a heart disease or not.

##### **Feedback**

User can give the feedback.

#### **4.6 Non-functional Requirements**

##### **Performance Requirements**

- User friendly: The system should be user friendly so that it can easily be understand by the user without any difficulty.
- Ease of maintenance :- System should be easy to maintain and use.
- Less time consuming: The system should be less time consuming which could be achieved by good programming.
- Error free: The system should easily handle the user error in any case.
- Static: Application runs on stand alone machine . Support only single user.

#### **4.7 SOFTWARE AND HARDWARE REQUIREMENTS**

This section describes the software and hardware requirements of the system .

## SOFTWARE REQUIREMENTS

**Operating system-** Windows/Linux Operating System This is the web Application which can run on any of the Operating System.

**Database-** MySql is used in storing the data in structured manner.

**XAMPP-** is a Software used for server which is use to serve the client what he/she wants from the server.

**Browser-** Any of the browser can be used to run and test the web application's Appearance and working eg. Internet Explorer, Google Chrome, Mozilla Firefox etc.

**Development tools and Programming language-** React,Tailswind css is used to write the whole designing and operational code. Flask is used for backend maintenance and MySql for database.

## HARDWARE REQUIREMENTS

- Desktop/Laptop any configuration.
- 8 GB RAM

### Software Requirements with specifications

Name of Components	Specifications
Operating system	Windows
Language	JavaScript,Python
Software Development kit	XAMPP, Vs Code,Pycharm
Markup Language Enable	HTML
Libraries	React,Tailwind CSS,Flask

### Hardware Requirements with specifications

Name of Components	Specifications
Desktop/Laptop	Any Configuration
RAM	8 GB

#### 4.8 Software System Attributes

- **Portability:-** The system should be machine independent.
- **Security:-** The system is designed in such a way that it will store the recorded data in the system of the owner. The system will be secure from unauthorized access of the application.
- **Maintainability:** The system will be designed in a maintainable order. The system can be easily modified and renewed according to the need of the organization.

#### 4.9 Feature of Heart Disease prediction website

- Will provide instant prediction about the heart disease
- It can be easily accessible as there is no need of registration
- People can analyse the heart disease of different peoples
- Graphics with a classic look and the feel of a royal Web Application.
- People can give feedback to make some changes or to give there recommendation
- Very easy to use
- Spreads awareness about heart disease
- Minimize manpower.
- Minimize time consumption.
- Greater efficiency.
- Fast.
- Better services.
- User friendliness and Interactive.
- Minimum time required.
- User friendly.
- Free for the user.

#### **4.10 Preliminary Description**

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of project request and make an informed judgement about the feasibility of the proposed project.

**Analyst working on the preliminary investigation should accomplish the following objectives:**

- Clarify and understand the project request.
  - Determine the size of the project.
  - Access costs and benefits of alternative approaches.
  - Determine the technical and operational feasibility of alternative approaches.
  - Report the findings to management with recommendations outlining the acceptance and rejection of the proposal
- 
-

## **CHAPTER 5**

### **ANALYSIS**

#### **5.1 Feasibility study**

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

- **Economical Feasibility**

For the economic feasibility, Economic analysis or cost/benefits analysis is most frequently used technique the effectiveness of a proposed system. it is a procedure to determine the benefits and saving those are expected from the proposes system and compare them with cost .if the benefits outweigh the costs, a decision is taken to design and implement the system. otherwise, further justification or alternative in proposed system will have to be made if it is to have a chance of being approved this is ongoing effort that improves in accuracy at each phase of a system life cycle

- **Technical Feasibility**

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of front end and back end platform.

- **Operational Feasibility**

No doubt the technically growing world needs more enhancement in technology, this apps is very user friendly and all inputs to be taken all self-explanatory even to a layman. As far our study is concerned, the clients will be comfortable and happy as the system has cut down their loads and bring the young generation to the same virtual world they are growing drastically.

Operational feasibility cover two aspects.one technical performance aspects and the other is acceptance within the organization.

Operation feasibility determine how the proposed the system will fit in with the current operation and what needs to implement the system.

CHAPTER 6

6.1 Diagram

Gantt chart

A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the task with milestones and columns indicate duration (Days).

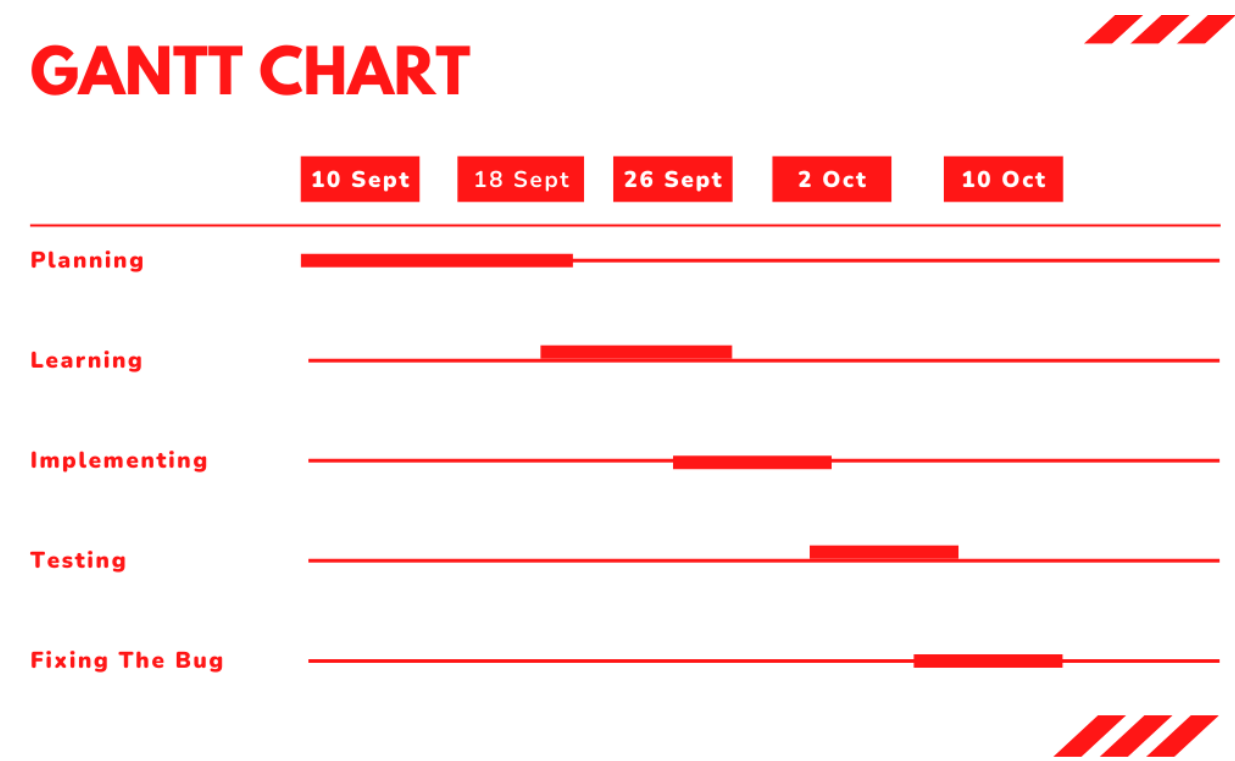


Fig. 2: Gantt Chart



## 6.2 DATA FLOW DIAGRAM

Are used to graphically represent the flow of data in a Employee Performance System. DFD describes the processes that are involved in a system to transfer data from the admin to the employee, employee to the employee, employee to admin etc.

Employee Performance System this system shows the flow of data in admin Modules on many Action. It shows the flow of data among the sub module in it Admin data flow on the sub screen.

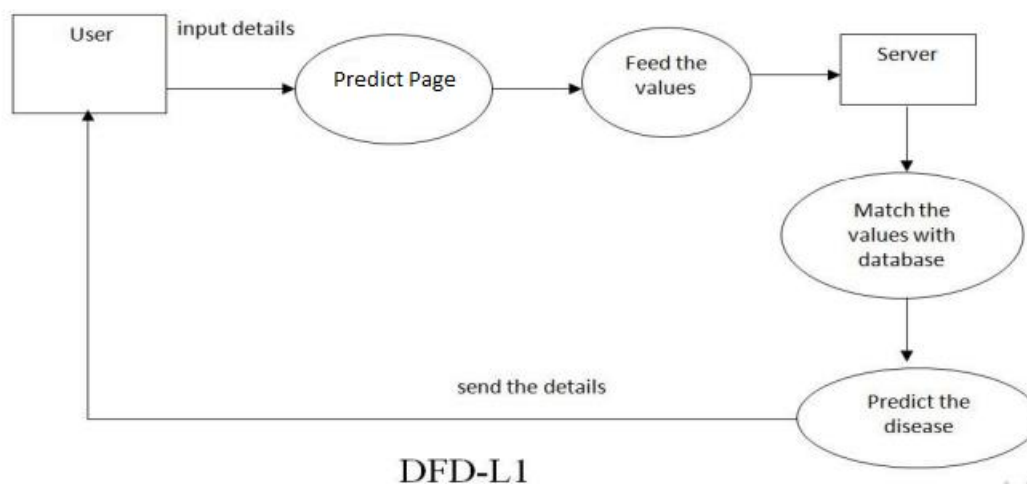


Fig. 3: Admin Mode DFD

This system shows the flow of data in Predict Module, Modules on many Action. It shows the flow of data How Hearrt Disease Prediction is Happeninig..

## 6.3 ENTITY RELATIONSHIP DIAGRAM

This ER Diagram represents the model of Heart Disease Prediction Website. The Entity Relationship Diagram show all visual instrument of Database table and relation between Prediction Page, Admin Page. All of it have Structured data and every entity may have some attributes.

Heart Disease Prediction Website Entity and their Attributes:

1. Admin Interface: (Attribute) : Password
2. User Interface : (Attribute) : Age, gender ,Chest pain, FBS, restecg, exang ,slope ,ca, thal, tbp ,chol, thals ,oldpeak.
3. Built Model : KNN, Decision tree , Random Forest,Navebyes.
4. Database : First Name , Last Name ,Email Id , City,Feedback,phone no .

### Heart Disease Prediction Website

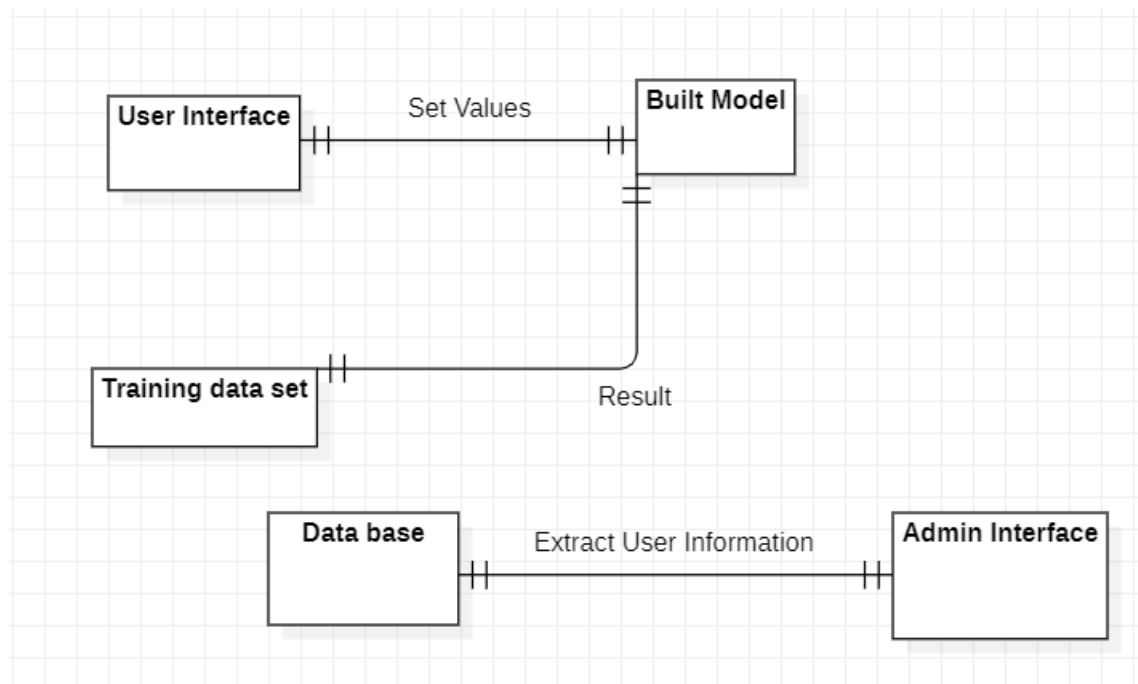


Fig. 5: Entity relationship Diagram

## 6.4 USE CASE DIAGRAM

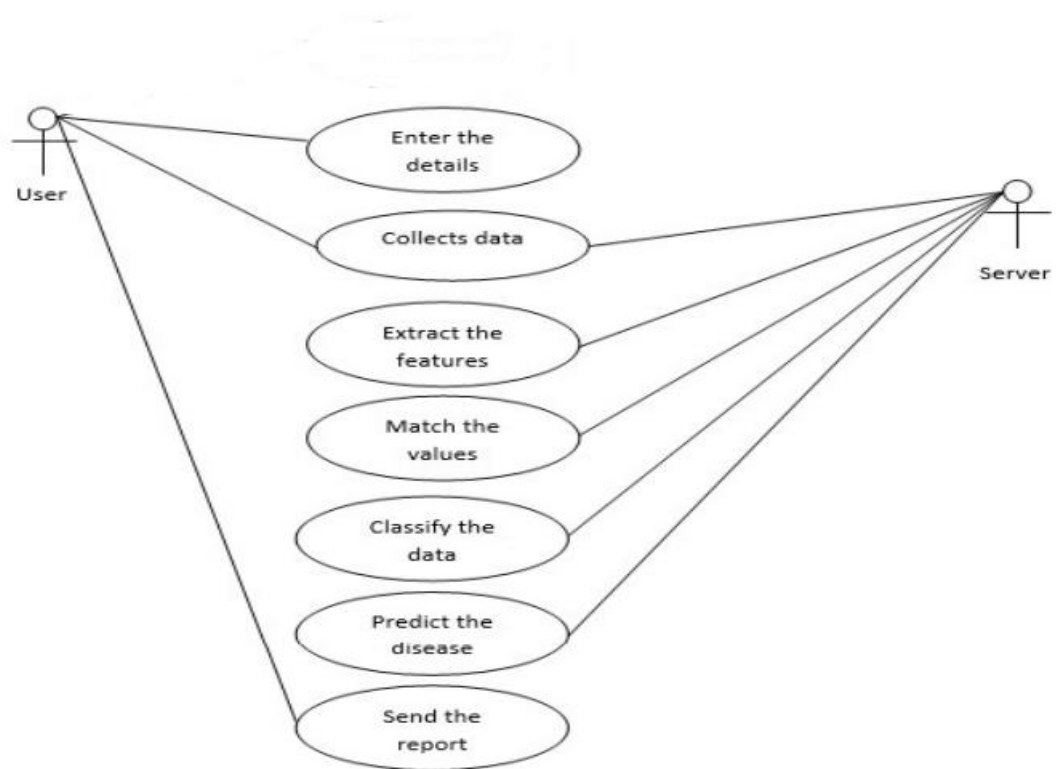
Use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

## USECASE DIAGRAM



## CHAPTER 7

### CODE

#### Front end – React..

#### Components

Admin.jsx

```
import axios from 'axios';
import React, { Component } from 'react';

class Admin extends Component {
  state={
    passwd:"",
    data:[]
  };

  handleOnChange = (event) => {
    const { name, value } = event.target;
    this.setState({
      [name]: value,
    });
  };

  prevent=(e)=>{

    e.preventDefault();
    // const login=prompt("enter password");
    // if(login==='abhinav'){

    const data=this.state;

    axios.post("http://127.0.0.1:5000/admin",data).then((response)=>{
      if(response.data.d===0)
      {

        alert("enter correct password")
        this.setState({data:[]})
      }else
      {
```

```

    this.setState({ data: response.data })
    console.log(this.state.sno)
  }
}).catch((error) => {
  alert(error)
})
}

// else{
//   alert("enter correct password")

render() {

  return (

    <>
    <input className='focus:border-rose-400 text-center mb-2 rounded shadow-xl h-10 w-44
placeholder-red-500' name="passwr" required placeholder="password" value={this.state.pass}
onChange={this.handleChange}/>
    <br/>
    <button onClick={this.prevent} className='text-center bg-black rounded-md hover:bg-gray-
700 text-white p-4 '>
      Get Feedback
    </button>
    <div className="flex flex-col">
    <div className="overflow-x-auto sm:-mx-6 mb-64 lg:-mx-8 ">
    <div className="py-2 inline-block min-w-full sm:px-6 mb-48 lg:px-8">
    <div className="overflow-hidden">
    <table className="min-w-full">
    <thead className="border-b-white border-b-4">
    <tr>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      ID
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      First Name
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      Last Name
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      Phone no
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      Email
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      Date

```

```

    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      City
    </th>
    <th scope="col" className="text-sm font-medium text-red-500 px-6 py-4 text-center">
      Feedback
    </th>
  </tr>
</thead>
<tbody>

    {this.state.data.map((student) => (
      <tr className='border-b-4 border-b-white text-red-500' key={student.sno}>
        <th scope="row ">{student.sno}</th>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.first_name}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.last_name}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.phone_no}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.email}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.date}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.city}</td>
        <td className='text-md text-red-500 font-mono px-6 py-4 whitespace-
nowrap'>{student.feedback}</td>
      </tr>
    ))}
  </tbody>
</table>
</div>
</div>
</div>
</div>
</>

  );
}
}

```

```
export default Admin;
```

## Errorpage.jsx

```

import React, { Component } from 'react';
class ErrorPage extends Component {
  state = { }

```

```

render() {
  return (<
    <div className='h-96'>

      <p className='text-red-500 mt-48 font-mono text-6xl'>Wrong URL</p>

    </div>
  </>);
}
}

```

export default ErrorPage;

### **Feedback.jsx**

```

import React, { Component } from 'react';
import axios from 'axios';
class Feedback extends Component {
  state = { first:"",
    last:"",
    city:"",
    phone:"",
    email:"",
    feedback:"",
  }

  handleOnChange = (event) => {
    const { name, value } = event.target;
    this.setState({
      [name]: value,
    });
  };

  prevent=(response)=>{
    response.preventDefault();
    const data=this.state;
    console.log(data)
    axios.post("http://127.0.0.1:5000/feedback",data).then(()=>{
      alert("success")

    }).catch((error)=>{
      alert(error)
    })
    alert("form submitted");
  }

  render() {
    const {fi,la,ci,ph,em,fe}=this.state;
    return (<

      <div className='App'>

```



```

    <div className='flex flex-col justify-center items-center border-gray-300 pt-24 pb-28'>
    <div className="block p-6 max-w-lg bg-white rounded-lg border border-gray-200 shadow-md
dark:bg-gray-800 dark:border-gray-700 dark:hover:bg-gray-700 ">
    <form method="POST" action="/feedback" className="w-full max-w-lg " name="feedback"
onSubmit={this.prevent} >
    <div className="flex flex-wrap -mx-3 mb-6 ">
    <div className="w-full md:w-1/2 px-3 mb-6 md:mb-0">
    <label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2"
htmlFor="grid-first-name">
    First Name
    </label>
    <input className="appearance-none block w-full text-red-400 placeholder-red-300 border
hover:shadow-md rounded py-3 px-4 mb-3 leading-tight focus:outline-none focus:bg-white
focus:border-rose-400" id="grid-first-name" type="text" placeholder='First Name' name="first"
onChange={this.handleChange} value={fi} required/>
    </div>
    <div className="w-full md:w-1/2 px-3">
    <label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2"
htmlFor="grid-last-name">
    Last Name
    </label>
    <input className="appearance-none block w-full hover:shadow-md text-red-400 border border-
gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white focus:border-rose-400
placeholder-red-300" id="grid-last-name" name="last" type="text" placeholder="Last Name"
onChange={this.handleChange} value={la} required/>
    </div>
    </div>

    <div className="flex flex-wrap -mx-3 mb-2">
    <div className="w-full md:w-1/3 px-3 mb-6 md:mb-0">
    <label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2"
htmlFor="grid-city">
    City
    </label>
    <input className="appearance-none block w-full hover:shadow-md text-red-400 border border-
gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white focus:border-rose-400
placeholder-red-300" id="grid-city" name="city" type="text" onChange={this.handleChange}
value={ci} placeholder="Ghaziabad" required/>
    </div>

    <div className="w-full md:w-1/3 px-3 mb-6 md:mb-0">
    <label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2"
htmlFor="grid-phone">
    Phone no
    </label>
    <input className="appearance-none block w-full hover:shadow-md text-red-400 placeholder-red-

```

```

300 border border-gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white
focus:border-rose-400" id="grid-phone" name="phone" type="tel" onChange={this.handleChange}
value={ph} placeholder="+91#####" required/>
</div>

<div className="w-full md:w-1/3 px-3 mb-6 md:mb-0">
  <label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2"
htmlFor="grid-email">
    Email
  </label>
  <input className="appearance-none block w-full text-red-400 hover:shadow-md border border-
gray-200 rounded py-3 px-4 leading-tight focus:outline-none focus:bg-white focus:border-rose-400
placeholder-rose-300" id="grid-email" type="email" onChange={this.handleChange} value={em}
name="email" placeholder="Email" required/>
</div>
</div>
<br/>
<label className="block uppercase tracking-wide text-red-500 text-xs font-bold mb-2 h-4"
htmlFor="grid-feedback">
  Feedback
</label>
<textarea className="text-red-400 selection:resize border rounded focus:outline-none hover:shadow-
md focus:shadow-outline focus:border-rose-400 h-24 w-4/5" onChange={this.handleChange}
value={fe} name="feedback" required id="grid-feedback"></textarea>
<br/>
<br/>
<button className="bg-red-500 hover:bg-rose-400 text-gray-800 font-semibold py-2 px-4 border
border-gray-400 rounded shadow" onClick={this.handleChange}>
  Submit
</button>

</form>
</div>

</div>
</div>
</>);
}
}

export default Feedback;

```

## Footer.jsx

```
import React, { Component } from 'react';
```

```

class Footer extends Component {
  state = { }
  render() {
    return (
      <>
        <div className="">
          <footer className="p-4 bg-red-600 rounded-lg shadow md:flex md:items-center md:justify-between md:p-6 dark:bg-gray-800">
            <span className="text-sm text-white sm:text-center text-font-sherif dark:text-gray-400">© 2022 <a href="https://www.linkedin.com/in/abhinav-choudhary-a1071a1b0/" className="hover:underline">Abhinav chTM</a>. All Rights Reserved.
          </span>

          </footer>
        </div>
      </>
    );
  }
}

export default Footer;

```

## Home.jsx

```

import { useNavigate } from 'react-router-dom';
//import pridict from '/pridict.jsx';
import img from './Model.JPG';

const Home=()=> {

  const prevent=(e)=>{
    e.preventDefault();

  }
  const navigate=useNavigate();

  return (<>
    <div className='App'>
      <br/>
      <div className='flex justify-center'>
        <div className="p-6 max-w-5/6 w-4/5 bg-white relative rounded-lg border border-gray-200 shadow-lg dark:bg-gray-800 dark:border-gray-700 text-left text-red-500"><h1 className='text-begin font-bold text-3xl font-serif text-red-500' >Introduction</h1><br/><p className='font-serif'>Among

```

various life-threatening diseases, heart disease has garnered a great deal of attention in medical research. The diagnosis of heart disease is a challenging task, which can offer automated prediction about the heart condition of patient so that further treatment can be made effective. The diagnosis of heart disease is usually based on signs, symptoms and physical examination of the patient. There are several factors that increase the risk of heart disease, such as smoking habit, body cholesterol level, family history of heart disease, obesity, high blood pressure, and lack of physical exercise.

Cardiovascular diseases are very common these days, they describe a range of conditions that could affect your heart. World health organization estimates that 17.9 million global deaths from (Cardiovascular diseases) CVDs.

It is the primary reason of deaths in adults. Our project can help predict the people who are likely to diagnose with a heart disease by help of their medical history. It recognizes who all are having any symptoms of heart disease such as chest pain or high blood pressure and can help in diagnosing disease with less medical tests and effective treatments, so that they can be cured accordingly.

</p></div>

</div>

<br/>

<br/>

<div className='flex justify-center '>

<div className="p-6 max-w-5/6 w-3/6 bg-white relative rounded-lg border border-gray-200 shadow-lg dark:bg-gray-800 dark:border-gray-700">

<button onClick={()=>navigate("/pridict")} onChange={prevent} className='bg-black max-w-4/5 rounded hover:bg-gray-500 font-bold h-12 w-44 text-zinc-50 '>Pridict</button>

<div className='flex justify-end absolute -right-10 -top-10'>

<svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" strokeWidth="1.0" stroke="red-600" className="h-4/5 w-2/5 bottom-0 left-0 fill-red-600 max-w-sm">

<path strokeLinecap="round" strokeLinejoin="round" d="M21 8.25c0-2.485-2.099-4.5-4.688-4.5-1.935 0-3.597 1.126-4.312 2.733-.715-1.607-2.377-2.733-4.313-2.733C5.1 3.75 3 5.765 3 8.25c0 7.22 9 12 9 12s9-4.78 9-12z" />

</svg>

</div>

</div>

</div>

<br/>

<div className='flex justify-center'>

<div className="p-6 max-w-5/6 w-4/5 bg-white relative mb-16 rounded-lg border border-gray-200 shadow-lg dark:bg-gray-800 dark:border-gray-700 text-left text-red-400">

<h1 className='text-center text-3xl text-red-500 font-serif font-bold'>Work flow of Model</h1>

<img src={img} alt="model"/>

</div>

</div>

```
    </div>

    </>);
  }
}
```

```
export default Home;
```

### **MainApp.jsx**

```
import React, { Component } from 'react';
import Footer from './Footer.jsx';
```

```
import NavBar from './NavBar.jsx';
```

```
class MainApp extends Component {
  state = { }
  render() {
    return (<>

      <NavBar/>
      <Footer/>

      </>);
  }
}
```

```
export default MainApp;
```

### **NavBar.jsx**

```
import React, { Component } from 'react';
import Footer from './Footer.jsx';
```

```
import NavBar from './NavBar.jsx';
```

```
class MainApp extends Component {
```

```

    state = { }
    render() {
      return (<
        <NavBar/>
        <Footer/>

        </>);
    }
  }

export default MainApp;
Predict.jsx
import React, { Component } from 'react';
import Footer from './Footer.jsx';

```

```

import NavBar from './NavBar.jsx';

```

```

class MainApp extends Component {
  state = { }
  render() {
    return (<
      <NavBar/>
      <Footer/>

      </>);
  }
}

export default MainApp;

```

**Backend – Python ,Flask**

**Dataset :**

age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal,target

63,1,3,145,233,1,0,150,0,2.3,0,0,1,1  
37,1,2,130,250,0,1,187,0,3.5,0,0,2,1  
41,0,1,130,204,0,0,172,0,1.4,2,0,2,1  
56,1,1,120,236,0,1,178,0,0.8,2,0,2,1  
57,0,0,120,354,0,1,163,1,0.6,2,0,2,1  
57,1,0,140,192,0,1,148,0,0.4,1,0,1,1  
56,0,1,140,294,0,0,153,0,1.3,1,0,2,1  
44,1,1,120,263,0,1,173,0,0,2,0,3,1  
52,1,2,172,199,1,1,162,0,0.5,2,0,3,1  
57,1,2,150,168,0,1,174,0,1.6,2,0,2,1  
54,1,0,140,239,0,1,160,0,1.2,2,0,2,1  
48,0,2,130,275,0,1,139,0,0.2,2,0,2,1  
49,1,1,130,266,0,1,171,0,0.6,2,0,2,1  
64,1,3,110,211,0,0,144,1,1.8,1,0,2,1  
58,0,3,150,283,1,0,162,0,1,2,0,2,1  
50,0,2,120,219,0,1,158,0,1.6,1,0,2,1  
58,0,2,120,340,0,1,172,0,0,2,0,2,1  
66,0,3,150,226,0,1,114,0,2.6,0,0,2,1  
43,1,0,150,247,0,1,171,0,1.5,2,0,2,1  
69,0,3,140,239,0,1,151,0,1.8,2,2,2,1  
59,1,0,135,234,0,1,161,0,0.5,1,0,3,1  
44,1,2,130,233,0,1,179,1,0.4,2,0,2,1  
42,1,0,140,226,0,1,178,0,0,2,0,2,1  
61,1,2,150,243,1,1,137,1,1,1,0,2,1  
40,1,3,140,199,0,1,178,1,1.4,2,0,3,1  
71,0,1,160,302,0,1,162,0,0.4,2,2,2,1  
59,1,2,150,212,1,1,157,0,1.6,2,0,2,1  
51,1,2,110,175,0,1,123,0,0.6,2,0,2,1  
65,0,2,140,417,1,0,157,0,0.8,2,1,2,1  
53,1,2,130,197,1,0,152,0,1.2,0,0,2,1  
41,0,1,105,198,0,1,168,0,0,2,1,2,1  
65,1,0,120,177,0,1,140,0,0.4,2,0,3,1  
44,1,1,130,219,0,0,188,0,0,2,0,2,1  
54,1,2,125,273,0,0,152,0,0.5,0,1,2,1  
51,1,3,125,213,0,0,125,1,1.4,2,1,2,1  
46,0,2,142,177,0,0,160,1,1.4,0,0,2,1  
54,0,2,135,304,1,1,170,0,0,2,0,2,1  
54,1,2,150,232,0,0,165,0,1.6,2,0,3,1  
65,0,2,155,269,0,1,148,0,0.8,2,0,2,1  
65,0,2,160,360,0,0,151,0,0.8,2,0,2,1  
51,0,2,140,308,0,0,142,0,1.5,2,1,2,1  
48,1,1,130,245,0,0,180,0,0.2,1,0,2,1  
45,1,0,104,208,0,0,148,1,3,1,0,2,1  
53,0,0,130,264,0,0,143,0,0.4,1,0,2,1  
39,1,2,140,321,0,0,182,0,0,2,0,2,1  
52,1,1,120,325,0,1,172,0,0.2,2,0,2,1

44,1,2,140,235,0,0,180,0,0,2,0,2,1  
47,1,2,138,257,0,0,156,0,0,2,0,2,1  
53,0,2,128,216,0,0,115,0,0,2,0,0,1  
53,0,0,138,234,0,0,160,0,0,2,0,2,1  
51,0,2,130,256,0,0,149,0,0,5,2,0,2,1  
66,1,0,120,302,0,0,151,0,0,4,1,0,2,1  
62,1,2,130,231,0,1,146,0,1,8,1,3,3,1  
44,0,2,108,141,0,1,175,0,0,6,1,0,2,1  
63,0,2,135,252,0,0,172,0,0,2,0,2,1  
52,1,1,134,201,0,1,158,0,0,8,2,1,2,1  
48,1,0,122,222,0,0,186,0,0,2,0,2,1  
45,1,0,115,260,0,0,185,0,0,2,0,2,1  
34,1,3,118,182,0,0,174,0,0,2,0,2,1  
57,0,0,128,303,0,0,159,0,0,2,1,2,1  
71,0,2,110,265,1,0,130,0,0,2,1,2,1  
54,1,1,108,309,0,1,156,0,0,2,0,3,1  
52,1,3,118,186,0,0,190,0,0,1,0,1,1  
41,1,1,135,203,0,1,132,0,0,1,0,1,1  
58,1,2,140,211,1,0,165,0,0,2,0,2,1  
35,0,0,138,183,0,1,182,0,1,4,2,0,2,1  
51,1,2,100,222,0,1,143,1,1,2,1,0,2,1  
45,0,1,130,234,0,0,175,0,0,6,1,0,2,1  
44,1,1,120,220,0,1,170,0,0,2,0,2,1  
62,0,0,124,209,0,1,163,0,0,2,0,2,1  
54,1,2,120,258,0,0,147,0,0,4,1,0,3,1  
51,1,2,94,227,0,1,154,1,0,2,1,3,1  
29,1,1,130,204,0,0,202,0,0,2,0,2,1  
51,1,0,140,261,0,0,186,1,0,2,0,2,1  
43,0,2,122,213,0,1,165,0,0,2,1,0,2,1  
55,0,1,135,250,0,0,161,0,1,4,1,0,2,1  
51,1,2,125,245,1,0,166,0,2,4,1,0,2,1  
59,1,1,140,221,0,1,164,1,0,2,0,2,1  
52,1,1,128,205,1,1,184,0,0,2,0,2,1  
58,1,2,105,240,0,0,154,1,0,6,1,0,3,1  
41,1,2,112,250,0,1,179,0,0,2,0,2,1  
45,1,1,128,308,0,0,170,0,0,2,0,2,1  
60,0,2,102,318,0,1,160,0,0,2,1,2,1  
52,1,3,152,298,1,1,178,0,1,2,1,0,3,1  
42,0,0,102,265,0,0,122,0,0,6,1,0,2,1  
67,0,2,115,564,0,0,160,0,1,6,1,0,3,1  
68,1,2,118,277,0,1,151,0,1,2,1,3,1  
46,1,1,101,197,1,1,156,0,0,2,0,3,1  
54,0,2,110,214,0,1,158,0,1,6,1,0,2,1  
58,0,0,100,248,0,0,122,0,1,1,0,2,1  
48,1,2,124,255,1,1,175,0,0,2,2,2,1  
57,1,0,132,207,0,1,168,1,0,2,0,3,1  
52,1,2,138,223,0,1,169,0,0,2,4,2,1  
54,0,1,132,288,1,0,159,1,0,2,1,2,1  
45,0,1,112,160,0,1,138,0,0,1,0,2,1



53,1,0,142,226,0,0,111,1,0,2,0,3,1  
62,0,0,140,394,0,0,157,0,1,2,1,0,2,1  
52,1,0,108,233,1,1,147,0,0,1,2,3,3,1  
43,1,2,130,315,0,1,162,0,1,9,2,1,2,1  
53,1,2,130,246,1,0,173,0,0,2,3,2,1  
42,1,3,148,244,0,0,178,0,0,8,2,2,2,1  
59,1,3,178,270,0,0,145,0,4,2,0,0,3,1  
63,0,1,140,195,0,1,179,0,0,2,2,2,1  
42,1,2,120,240,1,1,194,0,0,8,0,0,3,1  
50,1,2,129,196,0,1,163,0,0,2,0,2,1  
68,0,2,120,211,0,0,115,0,1,5,1,0,2,1  
69,1,3,160,234,1,0,131,0,0,1,1,1,2,1  
45,0,0,138,236,0,0,152,1,0,2,1,0,2,1  
50,0,1,120,244,0,1,162,0,1,1,2,0,2,1  
50,0,0,110,254,0,0,159,0,0,2,0,2,1  
64,0,0,180,325,0,1,154,1,0,2,0,2,1  
57,1,2,150,126,1,1,173,0,0,2,2,1,3,1  
64,0,2,140,313,0,1,133,0,0,2,2,0,3,1  
43,1,0,110,211,0,1,161,0,0,2,0,3,1  
55,1,1,130,262,0,1,155,0,0,2,0,2,1  
37,0,2,120,215,0,1,170,0,0,2,0,2,1  
41,1,2,130,214,0,0,168,0,2,1,0,2,1  
56,1,3,120,193,0,0,162,0,1,9,1,0,3,1  
46,0,1,105,204,0,1,172,0,0,2,0,2,1  
46,0,0,138,243,0,0,152,1,0,1,0,2,1  
64,0,0,130,303,0,1,122,0,2,1,2,2,1  
59,1,0,138,271,0,0,182,0,0,2,0,2,1  
41,0,2,112,268,0,0,172,1,0,2,0,2,1  
54,0,2,108,267,0,0,167,0,0,2,0,2,1  
39,0,2,94,199,0,1,179,0,0,2,0,2,1  
34,0,1,118,210,0,1,192,0,0,7,2,0,2,1  
47,1,0,112,204,0,1,143,0,0,1,2,0,2,1  
67,0,2,152,277,0,1,172,0,0,2,1,2,1  
52,0,2,136,196,0,0,169,0,0,1,1,0,2,1  
74,0,1,120,269,0,0,121,1,0,2,2,1,2,1  
54,0,2,160,201,0,1,163,0,0,2,1,2,1  
49,0,1,134,271,0,1,162,0,0,1,0,2,1  
42,1,1,120,295,0,1,162,0,0,2,0,2,1  
41,1,1,110,235,0,1,153,0,0,2,0,2,1  
41,0,1,126,306,0,1,163,0,0,2,0,2,1  
49,0,0,130,269,0,1,163,0,0,2,0,2,1  
60,0,2,120,178,1,1,96,0,0,2,0,2,1  
62,1,1,128,208,1,0,140,0,0,2,0,2,1  
57,1,0,110,201,0,1,126,1,1,5,1,0,1,1  
64,1,0,128,263,0,1,105,1,0,2,1,1,3,1  
51,0,2,120,295,0,0,157,0,0,6,2,0,2,1  
43,1,0,115,303,0,1,181,0,1,2,1,0,2,1  
42,0,2,120,209,0,1,173,0,0,1,0,2,1  
67,0,0,106,223,0,1,142,0,0,3,2,2,2,1

76,0,2,140,197,0,2,116,0,1.1,1,0,2,1  
70,1,1,156,245,0,0,143,0,0,2,0,2,1  
44,0,2,118,242,0,1,149,0,0,3,1,1,2,1  
60,0,3,150,240,0,1,171,0,0,9,2,0,2,1  
44,1,2,120,226,0,1,169,0,0,2,0,2,1  
42,1,2,130,180,0,1,150,0,0,2,0,2,1  
66,1,0,160,228,0,0,138,0,2,3,2,0,1,1  
71,0,0,112,149,0,1,125,0,1,6,1,0,2,1  
64,1,3,170,227,0,0,155,0,0,6,1,0,3,1  
66,0,2,146,278,0,0,152,0,0,1,1,2,1  
39,0,2,138,220,0,1,152,0,0,1,0,2,1  
58,0,0,130,197,0,1,131,0,0,6,1,0,2,1  
47,1,2,130,253,0,1,179,0,0,2,0,2,1  
35,1,1,122,192,0,1,174,0,0,2,0,2,1  
58,1,1,125,220,0,1,144,0,0,4,1,4,3,1  
56,1,1,130,221,0,0,163,0,0,2,0,3,1  
56,1,1,120,240,0,1,169,0,0,0,0,2,1  
55,0,1,132,342,0,1,166,0,1,2,2,0,2,1  
41,1,1,120,157,0,1,182,0,0,2,0,2,1  
38,1,2,138,175,0,1,173,0,0,2,4,2,1  
38,1,2,138,175,0,1,173,0,0,2,4,2,1  
67,1,0,160,286,0,0,108,1,1,5,1,3,2,0  
67,1,0,120,229,0,0,129,1,2,6,1,2,3,0  
62,0,0,140,268,0,0,160,0,3,6,0,2,2,0  
63,1,0,130,254,0,0,147,0,1,4,1,1,3,0  
53,1,0,140,203,1,0,155,1,3,1,0,0,3,0  
56,1,2,130,256,1,0,142,1,0,6,1,1,1,0  
48,1,1,110,229,0,1,168,0,1,0,0,3,0  
58,1,1,120,284,0,0,160,0,1,8,1,0,2,0  
58,1,2,132,224,0,0,173,0,3,2,2,2,3,0  
60,1,0,130,206,0,0,132,1,2,4,1,2,3,0  
40,1,0,110,167,0,0,114,1,2,1,0,3,0  
60,1,0,117,230,1,1,160,1,1,4,2,2,3,0  
64,1,2,140,335,0,1,158,0,0,2,0,2,0  
43,1,0,120,177,0,0,120,1,2,5,1,0,3,0  
57,1,0,150,276,0,0,112,1,0,6,1,1,1,0  
55,1,0,132,353,0,1,132,1,1,2,1,1,3,0  
65,0,0,150,225,0,0,114,0,1,1,3,3,0  
61,0,0,130,330,0,0,169,0,0,2,0,2,0  
58,1,2,112,230,0,0,165,0,2,5,1,1,3,0  
50,1,0,150,243,0,0,128,0,2,6,1,0,3,0  
44,1,0,112,290,0,0,153,0,0,2,1,2,0  
60,1,0,130,253,0,1,144,1,1,4,2,1,3,0  
54,1,0,124,266,0,0,109,1,2,2,1,1,3,0  
50,1,2,140,233,0,1,163,0,0,6,1,1,3,0  
41,1,0,110,172,0,0,158,0,0,2,0,3,0  
51,0,0,130,305,0,1,142,1,1,2,1,0,3,0  
58,1,0,128,216,0,0,131,1,2,2,1,3,3,0  
54,1,0,120,188,0,1,113,0,1,4,1,1,3,0

60,1,0,145,282,0,0,142,1,2.8,1,2,3,0  
60,1,2,140,185,0,0,155,0,3,1,0,2,0  
59,1,0,170,326,0,0,140,1,3.4,0,0,3,0  
46,1,2,150,231,0,1,147,0,3.6,1,0,2,0  
67,1,0,125,254,1,1,163,0,0.2,1,2,3,0  
62,1,0,120,267,0,1,99,1,1.8,1,2,3,0  
65,1,0,110,248,0,0,158,0,0.6,2,2,1,0  
44,1,0,110,197,0,0,177,0,0,2,1,2,0  
60,1,0,125,258,0,0,141,1,2.8,1,1,3,0  
58,1,0,150,270,0,0,111,1,0.8,2,0,3,0  
68,1,2,180,274,1,0,150,1,1.6,1,0,3,0  
62,0,0,160,164,0,0,145,0,6.2,0,3,3,0  
52,1,0,128,255,0,1,161,1,0,2,1,3,0  
59,1,0,110,239,0,0,142,1,1.2,1,1,3,0  
60,0,0,150,258,0,0,157,0,2.6,1,2,3,0  
49,1,2,120,188,0,1,139,0,2,1,3,3,0  
59,1,0,140,177,0,1,162,1,0,2,1,3,0  
57,1,2,128,229,0,0,150,0,0.4,1,1,3,0  
61,1,0,120,260,0,1,140,1,3.6,1,1,3,0  
39,1,0,118,219,0,1,140,0,1.2,1,0,3,0  
61,0,0,145,307,0,0,146,1,1,1,0,3,0  
56,1,0,125,249,1,0,144,1,1.2,1,1,2,0  
43,0,0,132,341,1,0,136,1,3,1,0,3,0  
62,0,2,130,263,0,1,97,0,1.2,1,1,3,0  
63,1,0,130,330,1,0,132,1,1.8,2,3,3,0  
65,1,0,135,254,0,0,127,0,2.8,1,1,3,0  
48,1,0,130,256,1,0,150,1,0,2,2,3,0  
63,0,0,150,407,0,0,154,0,4,1,3,3,0  
55,1,0,140,217,0,1,111,1,5.6,0,0,3,0  
65,1,3,138,282,1,0,174,0,1.4,1,1,2,0  
56,0,0,200,288,1,0,133,1,4,0,2,3,0  
54,1,0,110,239,0,1,126,1,2.8,1,1,3,0  
70,1,0,145,174,0,1,125,1,2.6,0,0,3,0  
62,1,1,120,281,0,0,103,0,1.4,1,1,3,0  
35,1,0,120,198,0,1,130,1,1.6,1,0,3,0  
59,1,3,170,288,0,0,159,0,0.2,1,0,3,0  
64,1,2,125,309,0,1,131,1,1.8,1,0,3,0  
47,1,2,108,243,0,1,152,0,0,2,0,2,0  
57,1,0,165,289,1,0,124,0,1,1,3,3,0  
55,1,0,160,289,0,0,145,1,0.8,1,1,3,0  
64,1,0,120,246,0,0,96,1,2.2,0,1,2,0  
70,1,0,130,322,0,0,109,0,2.4,1,3,2,0  
51,1,0,140,299,0,1,173,1,1.6,2,0,3,0  
58,1,0,125,300,0,0,171,0,0,2,2,3,0  
60,1,0,140,293,0,0,170,0,1.2,1,2,3,0  
77,1,0,125,304,0,0,162,1,0,2,3,2,0  
35,1,0,126,282,0,0,156,1,0,2,0,3,0  
70,1,2,160,269,0,1,112,1,2.9,1,1,3,0  
59,0,0,174,249,0,1,143,1,0,1,0,2,0

64,1,0,145,212,0,0,132,0,2,1,2,1,0  
57,1,0,152,274,0,1,88,1,1.2,1,1,3,0  
56,1,0,132,184,0,0,105,1,2.1,1,1,1,0  
48,1,0,124,274,0,0,166,0,0.5,1,0,3,0  
56,0,0,134,409,0,0,150,1,1.9,1,2,3,0  
66,1,1,160,246,0,1,120,1,0,1,3,1,0  
54,1,1,192,283,0,0,195,0,0,2,1,3,0  
69,1,2,140,254,0,0,146,0,2,1,3,3,0  
51,1,0,140,298,0,1,122,1,4.2,1,3,3,0  
43,1,0,132,247,1,0,143,1,0.1,1,4,3,0  
62,0,0,138,294,1,1,106,0,1.9,1,3,2,0  
67,1,0,100,299,0,0,125,1,0.9,1,2,2,0  
59,1,3,160,273,0,0,125,0,0,2,0,2,0  
45,1,0,142,309,0,0,147,1,0,1,3,3,0  
58,1,0,128,259,0,0,130,1,3,1,2,3,0  
50,1,0,144,200,0,0,126,1,0.9,1,0,3,0  
62,0,0,150,244,0,1,154,1,1.4,1,0,2,0  
38,1,3,120,231,0,1,182,1,3.8,1,0,3,0  
66,0,0,178,228,1,1,165,1,1,1,2,3,0  
52,1,0,112,230,0,1,160,0,0,2,1,2,0  
53,1,0,123,282,0,1,95,1,2,1,2,3,0  
63,0,0,108,269,0,1,169,1,1.8,1,2,2,0  
54,1,0,110,206,0,0,108,1,0,1,1,2,0  
66,1,0,112,212,0,0,132,1,0.1,2,1,2,0  
55,0,0,180,327,0,2,117,1,3.4,1,0,2,0  
49,1,2,118,149,0,0,126,0,0.8,2,3,2,0  
54,1,0,122,286,0,0,116,1,3.2,1,2,2,0  
56,1,0,130,283,1,0,103,1,1.6,0,0,3,0  
46,1,0,120,249,0,0,144,0,0.8,2,0,3,0  
61,1,3,134,234,0,1,145,0,2.6,1,2,2,0  
67,1,0,120,237,0,1,71,0,1,1,0,2,0  
58,1,0,100,234,0,1,156,0,0.1,2,1,3,0  
47,1,0,110,275,0,0,118,1,1,1,1,2,0  
52,1,0,125,212,0,1,168,0,1,2,2,3,0  
58,1,0,146,218,0,1,105,0,2,1,1,3,0  
57,1,1,124,261,0,1,141,0,0.3,2,0,3,0  
58,0,1,136,319,1,0,152,0,0,2,2,2,0  
61,1,0,138,166,0,0,125,1,3.6,1,1,2,0  
42,1,0,136,315,0,1,125,1,1.8,1,0,1,0  
52,1,0,128,204,1,1,156,1,1,1,0,0,0  
59,1,2,126,218,1,1,134,0,2.2,1,1,1,0  
40,1,0,152,223,0,1,181,0,0,2,0,3,0  
61,1,0,140,207,0,0,138,1,1.9,2,1,3,0  
46,1,0,140,311,0,1,120,1,1.8,1,2,3,0  
59,1,3,134,204,0,1,162,0,0.8,2,2,2,0  
57,1,1,154,232,0,0,164,0,0,2,1,2,0  
57,1,0,110,335,0,1,143,1,3,1,1,3,0  
55,0,0,128,205,0,2,130,1,2,1,1,3,0  
61,1,0,148,203,0,1,161,0,0,2,1,3,0

58,1,0,114,318,0,2,140,0,4.4,0,3,1,0  
58,0,0,170,225,1,0,146,1,2.8,1,2,1,0  
67,1,2,152,212,0,0,150,0,0.8,1,0,3,0  
44,1,0,120,169,0,1,144,1,2.8,0,0,1,0  
63,1,0,140,187,0,0,144,1,4,2,2,3,0  
63,0,0,124,197,0,1,136,1,0,1,0,2,0  
59,1,0,164,176,1,0,90,0,1,1,2,1,0  
57,0,0,140,241,0,1,123,1,0.2,1,0,3,0  
45,1,3,110,264,0,1,132,0,1.2,1,0,3,0  
68,1,0,144,193,1,1,141,0,3.4,1,2,3,0  
57,1,0,130,131,0,1,115,1,1.2,1,1,3,0  
57,0,1,130,236,0,0,174,0,0,1,1,2,0

### Machine learning Model :

```
""" # **data preprocessings**
```

```
"""
```

```
import pandas as pd  
import pickle  
data=pd.read_csv('heart_disease_data.csv')  
data.head(5)
```

```
data.tail(10)
```

```
data.isnull().sum()
```

```
data.describe()
```

```
data.dtypes
```

```
data['slope'].unique
```

```
X=data.drop(columns='target',axis=1)  
X.head(5)  
Y=data['target']  
Y.head(5)
```

```
"""#**importing classifiers**"""
```

```
from sklearn.tree import DecisionTreeClassifier  
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier  
from sklearn.naive_bayes import GaussianNB  
from sklearn.neighbors import KNeighborsClassifier
```

```

from sklearn.svm import SVC
from sklearn.model_selection import train_test_split

X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.25,random_state=22)
X_train.head(5)

"""#**Models**"""

#Decisisonontree
from sklearn.metrics import accuracy_score
dt_clf=DecisionTreeClassifier(max_depth=4)
#on max_depth 4 it is giving best accuracy
dt_clf.fit(X_train,Y_train)
dt_pred=dt_clf.predict(X_test)
dt_acc=accuracy_score(Y_test,dt_pred)
print(dt_acc*100)

rfc_clf=RandomForestClassifier(n_estimators=100,max_depth=4)
rfc_clf.fit(X_train,Y_train)
rfc_pred=rfc_clf.predict(X_test)
rfc_acc=accuracy_score(Y_test,rfc_pred)
print(rfc_acc*100)

#adaBoostClassifier
ada_clf=AdaBoostClassifier()
ada_clf.fit(X_train,Y_train)
ada_pred=ada_clf.predict(X_test)
ada_acc=accuracy_score(Y_test,ada_pred)
print(ada_acc*100)

#NaiveBaiyse
nb_clf=GaussianNB()
nb_clf.fit(X_train,Y_train)
nb_pred=nb_clf.predict(X_test)
nb_acc=accuracy_score(Y_test,nb_pred)
print(nb_acc*100)

#Kneighbours

knn_clf=KNeighborsClassifier(n_neighbors=20)
knn_clf.fit(X_train,Y_train)
knn_pred=knn_clf.predict(X_test)
knn_acc=accuracy_score(Y_test,knn_pred)
print(knn_acc*100)

#SVC

svc_clf=SVC()
svc_clf.fit(X_train,Y_train)

```

```

svc_pred=svc_clf.predict(X_test)
svc_acc=accuracy_score(Y_test,svc_pred)
print(svc_acc*100)

print("decision tree:",dt_acc*100)
print("random forest classifier:",rfc_acc*100)
print("adaboosclassifier:",ada_acc*100)
print("naiveBayse:",nb_acc*100)
print("knn classifier:",knn_acc*100)
print("svc:",svc_acc*100)
filename = 'p-model'
outfile = open(filename,'wb')
pickle.dump(nb_clf,outfile)
outfile.close()

```

### **Main app :**

```

import json
from flask import Flask,request,jsonify
import pickle
import numpy as np
import pandas as pd
from flask_cors import CORS
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
app=Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = 'mysql+pymysql://root:@localhost/heartpredict'

```

```

db = SQLAlchemy(app)

```

```

CORS(app)

```

```

class Feedback(db.Model):

```

```

    sno = db.Column(db.Integer,primary_key=True)
    f_name = db.Column(db.String(15),nullable=False)
    l_name = db.Column(db.String(15),nullable=False)
    city = db.Column(db.String(15),nullable=False)
    phone_no = db.Column(db.Integer,nullable=False)
    email = db.Column(db.String(15),nullable=False)
    feed = db.Column(db.String(100),nullable=False)
    date = db.Column(db.String(20),nullable=True)

```

```

@app.route("/")
def hello():
    infile = open('p-model', 'rb')
    model = pickle.load(infile)
    infile.close()
    a = [37, 1, 2, 130, 250, 0, 1, 187, 0, 3.5, 0, 0, 2]
    input = np.array(a)
    input = input.reshape(1, 13)
    feature_name = ["age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak",
"slope",
                    "ca", "thal"]
    c = pd.DataFrame(input, columns=feature_name)
    m=model.predict(c)
    print(m)
    if m==1:
        return "<h1> have heart attack</h1>"
    else:
        return "<h1>no heart desease</h1>"

@app.route("/feedback",methods=['POST','GET'])
def feedback():

    if request.method=='POST':
        first=request.json['first']
        last=request.json['last']
        city=request.json['city']
        phone=request.json['phone']
        email=request.json['email']
        feed=request.json['feedback']

        entry =
Feedback(f_name=first,l_name=last,city=city,phone_no=phone,email=email,feed=feed,date=datetime.n
ow())
        db.session.add(entry)
        db.session.commit()

        print(first)
        print(last)
        print(city)
        print(phone)
        print(email)
        print(feed)

        return ""

@app.route("/predict",methods=['POST','GET'])

```



```
def predict():
```

```
    if request.method=='POST':
        feature_name = ["age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang",
"oldpeak", "slope", "ca", "thal"]
        age=int(request.json['age'])
        gender=int(request.json['gender'])
        cp=int(request.json['cp'])
        trestbps=int(request.json['trestbps'])
        chol=int(request.json['chol'])
        fbs=int(request.json['fbs'])
        resecg = int(request.json['restecg'])
        thalach = int(request.json['thalach'])
        exang = int(request.json['exang'])
        oldpeak = float(request.json['oldpeak'])
        slope = int(request.json['slope'])
        ca = int(request.json['ca'])
        thal = int(request.json['thal'])

        # entry =
Feedback(f_name=first,l_name=last,city=city,phone_no=phone,email=email,feed=feed,date=datetime.now())
        # db.session.add(entry)
        # db.session.commit()

        print(type(age))
        print(type(thalach))
        print(type(slope))
        print(type(ca))
        print(type(oldpeak))
        print(type(thal))

        infile = open('p-model', 'rb')
        model = pickle.load(infile)
        infile.close()

        data=[age,gender,cp,trestbps,chol,fbs,resecg,thalach,exang,oldpeak,slope,ca,thal]
        input = np.array(data)
        input = input.reshape(1, 13)
        feature_name = ["age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang",
"oldpeak",
            "slope",
            "ca", "thal"]
        c = pd.DataFrame(input, columns=feature_name)
        m = model.predict(c)
        ans=int(m)
        o={"pred":ans}
        print(m)
        print(type(ans))
```

```
return o
```

```
return "i am get"
```

```
@app.route('/admin',methods=['GET','POST'])
def admin():
    if request.method=='POST':
        print("hello")
        print(request.json['passwrld'])
        if request.json['passwrld']=='abhinav':
            b=Feedback.query.all()
            l=[]

            for i in b:
                n=['sno','first_name','last_name','city','phone_no','email','feedback','date']
                m=[]

                s=i.sno
                f_n=i.f_name
                l_n=i.l_name
                c=i.city
                em=i.email
                phn=i.phone_no
                fe=i.feed
                da=i.date
                da=str(da)
                m.extend((s,f_n,l_n,c,phn,em,fe,da))
                jsn=dict(zip(n,m))
                l.append(jsn)

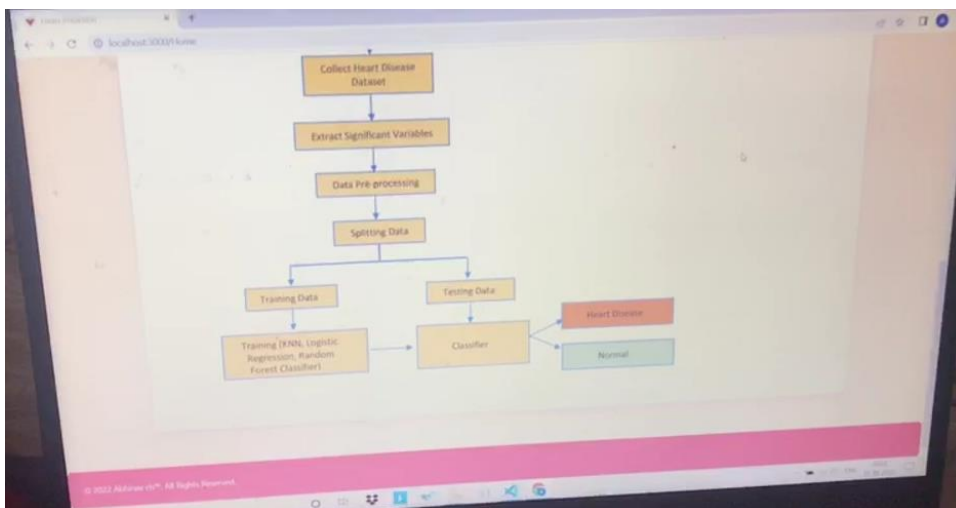
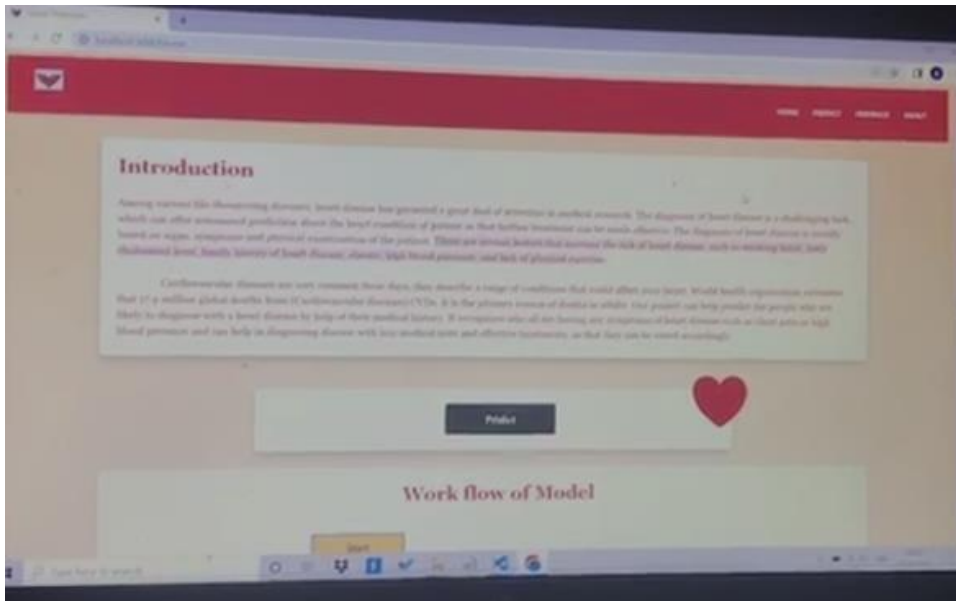
            api={"data":l}
            print(type(l))
            print(type(api))
            return l
        else:
            a=int(0)
            data={"d":a}
            return data

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

# CHAPTER 8

## OUTPUT

### 8.1 Home Page Of Website :



### 8.2 Prediction And User Input Form Page

Enter Details

Age

Gender

☐ Male ☐ Female

CHART PAIR TYPE

Chart Pair Type

TECHNIQUE

ADJUNCT

END

100 mg

100

☐ greater than 100 ☒ less than 100

Save Data

© 2022 All rights reserved.

HOME PREVIEW FEEDBACK

FIRST NAME LAST NAME

First Name Last Name

CITY PHONE NO EMAIL

City Phone No Email

FEEDBACK

Save Data

## **CHAPTER 9**

### **Conclusion**

The proposed system is GUI-based, user-friendly, scalable, reliable and an expandable system. The proposed working model can also help in reducing treatment costs by providing Initial diagnostics in time. The model can also serve the purpose of training tool for medical students and will be a soft diagnostic tool available for physician and cardiologist. General physicians can utilize this tool for initial diagnosis of cardio-patients. There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system. As we have developed a generalized system, in future we can use this system for the analysis of different data sets. The performance of the health's diagnosis can be improved significantly by handling numerous class labels in the prediction process, and it can be another positive direction of research. In DM warehouse, generally, the dimensionality of the heart database is high, so identification and selection of significant attributes for better diagnosis of heart disease are very challenging tasks for future research.

## CHAPTER 10: LITERARY REVIEW

The following research journals from AKTU Nalanda E-Consortium were used as references for completing this project:

1. Advances in Engineering Software: <https://www.sciencedirect.com/journal/advances-in-engineering-software>
2. International Journal on Software Tools for Technology Transfer:  
<https://www.springer.com/journal/10009>
3. Optimization Methods and Software: <https://www.tandfonline.com/journals/goms20>
4. Software Quality Journal: <https://www.springer.com/journal/11219/>
5. Journal of Web Semantics: <https://www.sciencedirect.com/journal/journal-of-web-semantics>
6. Start Programming Using HTML, CSS, and JavaScript:  
<https://www.taylorfrancis.com/books/mono/10.1201/b19402/start-programming-using-html-css-javascript-iztok-fajfar>
7. HTML5 Mobile Websites: <https://www.taylorfrancis.com/books/mono/10.4324/9780240818146/html5-mobile-websites-matthew-david>
8. Managing Web Projects:  
<https://www.taylorfrancis.com/books/mono/10.1201/9781439804964/managing-web-projects-edward-farkas>
9. Web Programming for Business:  
<https://www.taylorfrancis.com/books/mono/10.4324/9780203582084/web-programming-business-david-paper>
10. Web Technology: <https://www.taylorfrancis.com/books/mono/10.1201/9781351029902/web-technology-yakshi-kumar>
11. Web Security: <https://www.taylorfrancis.com/books/mono/10.1201/b18327/web-security-hanqing-wu-liz-zhao>
12. E-Commerce Usability:  
<https://www.taylorfrancis.com/books/mono/10.1201/9780203245910/commerce-usability-david-travis>
13. Innovations in E-Systems for Business and Commerce:  
<https://www.taylorfrancis.com/books/edit/10.1201/9781315207353/innovations-systems-business-commerce-abdelkhalak-el-hami-seifedine-kadry>

14. Cloud Database Development and Management:

<https://www.taylorfrancis.com/books/mono/10.1201/b15264/cloud-database-developmentmanagement-lee-chao>

15. NoSQL Database for Storage and Retrieval of Data in Cloud:

<https://www.taylorfrancis.com/books/edit/10.1201/9781315155579/nosql-ganesh-chandradeka>

# CHAPTER 11

## REFERENCE

- A, A. S., & Naik, C. (2016). Different Data Mining Approaches for Predicting Heart Disease, 277–281. <https://doi.org/10.15680/IJIRSET.2016.0505545>
- Beyene, C., & Kamat, P. (2018). Survey on prediction and analysis the occurrence of heart disease using data mining techniques. *International Journal of Pure and Applied Mathematics*, 118(Special Issue 8), 165–173. Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85041895038&partnerID=40&md5=2f0b0c5191a82bc0c3f0daf67d73bc81>
- Brownlee, J. (2016). Naive Bayes for Machine Learning. Retrieved March 4, 2019, from <https://machinelearningmastery.com/naive-bayes-for-machine-learning/>
- Kirmani, M. (2017). Cardiovascular Disease Prediction using Data Mining Techniques. *Oriental Journal of Computer Science and Technology*, 10(2), 520–528. <https://doi.org/10.13005/ojcs/10.02.38>
- Polaraju, K., Durga Prasad, D., & Tech Scholar, M. (2017). Prediction of Heart Disease using Multiple Linear Regression Model. *International Journal of Engineering Development and Research*, 5(4), 2321–9939. Retrieved from [www.ijedr.org](http://www.ijedr.org)
- Purushottam, Saxena, K., & Sharma, R. (2016). Efficient Heart Disease Prediction System. In *Procedia Computer Science* (Vol. 85, pp. 962–969). <https://doi.org/10.1016/j.procs.2016.05.288>
- Sai, P. P., & Reddy, C. (2017). International Journal of Computer Science and Mobile Computing HEART DISEASE PREDICTION USING ANN ALGORITHM IN DATA MINING. *International Journal of Computer Science & Mobile Computing*, 6(4), 168–172. Retrieved from [www.ijcsmc.com](http://www.ijcsmc.com)
- Science, C., & Faculty, G. M. (2009). Heart Disease Prediction Using Machine learning and Data Mining Technique. *Ijcsc* 0973-7391, 7, 1–9.
- Soni, J., Ansari, U., & Sharma, D. (2011). Intelligent and Effective Heart Disease Prediction System using Weighted Associative Classifiers. *Heart Disease*, 3(6), 2385–2392