

**VISUALIZING AND PREDICTING HEART
DISEASES WITHAN INTERACTIVE DASHBOARD**

Domain: DATA ANALYSIS

Team ID: PNT2022TMID28187

Submitted by

Sriram E (312419104143)

Suresh G (312419104147)

Venkateshwaran R (312419104156)

Vignesh Balaji A (312419104158)

Vignesh Kumar R (312419104159)

- 1. INTRODUCTION**
 - 1.1 Project Overview
 - 1.2 Purpose
- 2. LITERATURE SURVEY**
 - 2.1 Existing problem
 - 2.2 References
 - 2.3 Problem Statement Definition
- 3. IDEATION & PROPOSED SOLUTION**
 - 3.1 Empathy Map Canvas
 - 3.2 Ideation & Brainstorming
 - 3.3 Proposed Solution
 - 3.4 Problem Solution fit
- 4. REQUIREMENT ANALYSIS**
 - 4.1 Functional requirement
 - 4.2 Non-Functional requirements
- 5. PROJECT DESIGN**
 - 5.1 Data Flow Diagrams
 - 5.2 Solution & Technical Architecture
 - 5.3 User Stories
- 6. PROJECT PLANNING & SCHEDULING**
 - 6.1 Sprint Planning & Estimation
 - 6.2 Sprint Delivery Schedule
 - 6.3 Reports from JIRA
- 7. CODING & SOLUTIONING (Explain the features added in the project along with code)**
 - 7.1 Feature 1
 - 7.2 Feature 2
 - 7.3 Database Schema (if Applicable)
- 8. TESTING**
 - 8.1 Test Cases
 - 8.2 User Acceptance Testing
- 9. RESULTS**
 - 9.1 Performance Metrics
- 10. ADVANTAGES & DISADVANTAGES**
- 11. CONCLUSION**
- 12. FUTURE SCOPE**
- 13. APPENDIX**
 - Source Code
 - GitHub & Project Demo Link

CHAPTER 1

INTRODUCTION:

1.1 : PROJECT OVERVIEW

The terms "heart disease" and "cardiovascular disease" are frequently used interchangeably. Heart disease is a general term that covers a wide range of heart related medical conditions. The irregular health state that directly affects the heart and all of its components is characterized by these medical conditions. In order to forecast cardiac disease, this study discusses various data mining, big data, and machine learning techniques. Building an important model for the medical system to forecast heart disease or cardiovascular illness requires the use of data mining and machine learning. Our application helps the user in finding out if they have heart disease or not. They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users. This project focuses on Random Forest Classifier. The accuracy of our project is 87% for which is better than most other systems in terms of achieving accuracy quickly.

1.2 : PURPOSE

This project's goal is to determine, depending on the patient's medical characteristics—such as gender, age, chest pain, fasting blood sugar level, etc.—whether they are likely to be diagnosed with any cardiovascular heart illnesses. The leading cause of death in the developed world is heart disease. Heart disease cases are rising quickly every day, thus it's crucial and worrisome to predict any potential illnesses in advance. This diagnosis is a challenging task that requires accuracy and efficiency. Therefore, there needs to be work done to help prevent the risks of having a heart attack or stroke. It is the main factor in adult deaths. By using a person's medical history, our initiative can identify those who are most likely to be diagnosed with a cardiac condition. It can assist in identifying disease with less medical tests and effective therapies, so that patients can be treated appropriately. It can identify anyone who is experiencing any heart disease symptoms, such as chest pain or high blood pressure. Around the world, machine learning is applied in many different fields. There is no exception in the healthcare sector. Machine learning may be crucial in determining whether locomotor disorders, heart illnesses, and other conditions are present or absent. If foreseen well in advance, such information can offer valuable insights to doctors, who can then customise their diagnosis and course of care for each patient.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

A quiet significant amount of works related to the diagnosis of Heart disease using Machine Learning algorithms have been made. An efficient heart disease prediction has been made by using various algorithms some of them include Logistic Regression, KNN, Random Forest Classifier etc. It can be seen in results that each algorithm has its strength to register the defined objectives. The model incorporating IHDPS had the ability to calculate the decision boundary using the previous and new model of machine learning and deep learning. It facilitated the important and the most basic factors/knowledge such as family history connected with any heart disease. But the accuracy that was obtained in such IHDPS model was far more less than the new upcoming model such as detecting coronary heart disease using artificial neural network and other algorithms of machine and deep learning.

2.2 REFERENCES

- [1] Ali, Liaqat, et al, "An optimized stacked support vector machines based expert system for the effective prediction of heart failure." IEEE Access 7 (2019): 54007-54014. www.ijcrt.org © 2020 IJCRT | Volume 8, Issue 8 August 2020 | ISSN: 2320-2882 IJCRT2008170 International Journal of Creative Research Thoughts (IJCRT) www.ijcrt.org 1606
- [2] Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective heart disease prediction using hybrid machine learning techniques." IEEE Access 7 (2019): 81542-81554.

- [3] Purushottam, Kanak Saxena and Richa Sharma, "Efficient heart disease prediction system." *Procedia Computer Science* 85 (2016): 962-969.
- [4] Singh, Yeshvendra K., Nikhil Sinha, and Sanjay K. Singh, "Heart Disease Prediction System Using Random Forest", *International Conference on Advances in Computing and Data Sciences*. Springer, Singapore, 2016.
- [5] Santhana Krishnan. J, Geetha S., "Prediction of Heart Disease Using Machine Learning Algorithms", 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)
- [6] Bo Jin ,Chao Che, Zhen Liu, Shulong Zhang, Xiaomeng Yin, And Xiaopeng Wei, "Predicting the Risk of Heart Failure With EHR Sequential Data Modeling", *IEEE Access* 2018.
- [7] Aakash Chauhan , Aditya Jain , Purushottam Sharma , Vikas Deep, "Heart Disease Prediction using Evolutionary Rule Learning", "International Conference on "Computational Intelligence and Communication Technology" (CICT 2018).
- [8] Boshra Bahrami, Mirsaeid Hosseini Shirvani, "Prediction and Diagnosis of Heart Disease by Data Mining Techniques", *Journal of Multidisciplinary Engineering Science and Technology (JMEST)* ISSN: 3159-0040 Vol. 2 Issue 2, February–2015.
- [9] M.Satish, D Sridhar, "Prediction of Heart Disease in Data Mining Technique", *International Journal of Computer Trends & Technology (IJCTT)*, 2015.
- [10] Lokanath Sarangi, Mihir Narayan Mohanty, Srikanta Pattnaik, "An Intelligent Decision Support System for Cardiac Disease Detection", *IJCTA*, International Press 2015.

3.3 PROPOSED SOLUTION

Our application helps the user in finding out if they have heart disease or not. They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users. Our application has one of the smoothest user interfaces on the internet making it easy for the user to find their needs quickly and efficiently. And the tool utilizes best machine learning algorithms for better prediction. There's separate sections for viewing treatment options, warning signs of cardiac arrest, risk factors and causes of various types of heart diseases.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict whether the patient has heart disease or not.
2.	Idea / Solution description	Our application helps the user in finding out if they have heart disease or not. They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users.
3.	Novelty / Uniqueness	Our application has one of the smoothest user interfaces on the internet making it easy for the user to find their needs quickly and efficiently. And the tool utilizes best machine learning algorithms for better prediction. There's separate sections for viewing treatment options, warning signs of cardiac arrest, risk factors and causes of various types of heart diseases.
4.	Social Impact / Customer Satisfaction	Users can find out if they have heart diseases very easily hence minimising death rate around the world.
5.	Business Model (Revenue Model)	Not for profit model.
6.	Scalability of the Solution	A future update shall comprise of section for viewing renowned cardiologists and scan centres in their city.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that we have found a problem with our customer and that the solution we have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioural patterns and recognize what would work and why. The purpose is to solve complex problems in a way that fits the state of your customers and succeed faster and increase your solution adoption by tapping into existing mediums and channels of behaviour.

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Who is your customer? i.e. working parents of 0-5 y/o kids	6. CUSTOMER CONSTRAINTS What constraints prevent your customer from taking action or limit their choices of solutions? i.e. accounting power, budget, no cash, network connection, available devices	5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What does it cost? What solutions have/it vs. pen and paper is an alternative to digital telehealth	Explore AS, differentiate
	1. Patients 2. Curious users 3. Family members of the patients	Some people are not willing to go to hospital due to financial constraints and some staying at remote locations	Visiting the experienced cardiologist to verify	
Focus on J&P, map into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, or place different sides.	9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the basic story behind the need to do this job? i.e. customers have to do it because of the change in regulations.	7. BEHAVIOUR What does your customer do to address the problem and get the job done? (i.e. directly related: find the right solar panel installer; calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. in nepotism))	Focus on J&P, map into BE, understand RC
	To predict whether the user has heart disease or not	Users are in need of these kind of solutions because 1. Because heart disease is hereditary 2. Some people may be leading unhealthy lifestyles which might make them more susceptible to heart related diseases	1. Schedule an appointment 2. Search for the best cardiologist online 3. Speak to family/friends regarding solutions	
Identify strong TR & EM	3. TRIGGERS What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.	10. YOUR SOLUTION If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.	8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from it:	Extract online & offline CH of BE
	Feeling discomfort in their chest. Users making sure that they are healthy	Our application helps the user in finding out if they have heart disease or not. They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users	1. Talk with friends/family 2. Browse health related websites	
	4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterwards? i.e. fast, insecure > confident, in control - use it to your career/career strategy & design.		8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from it and use them for customer development.	
	Before taking the test, the user will be anxious. After taking the test, the user will either be relieved that they are healthy or go to the hospital for double checking/treatment		1. Reach out to the nearest cardiologist	

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

- Users have to register.
- Function to view the homepage by the user.
- Function to display information related to heart diseases on the website.
- Function to provide textboxes to enter medical results.
- Function to predict heart disease using ML model.
- Function to display visualisations of the final results.
- Function to provide dashboard to user.

4.2 NON-FUNCTIONAL REQUIREMENTS

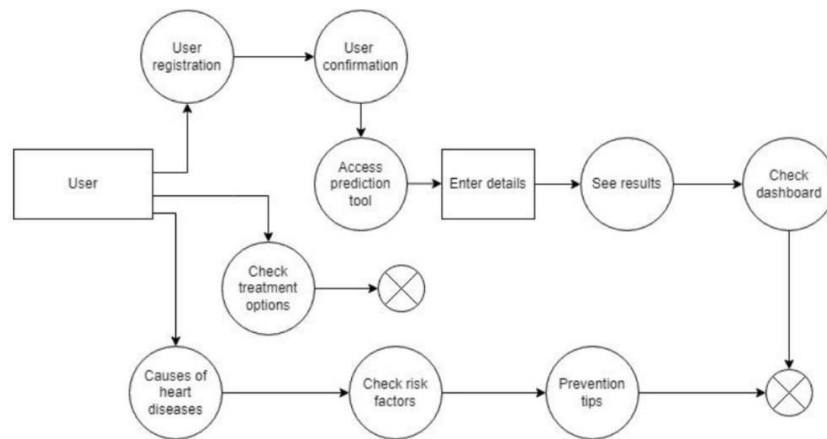
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The website will utilise better user interface for easy navigation. The process of finding out the results will be smooth and easy for the user.
NFR-2	Security	The website will be protected against SQL injection, DDoS attacks and SHA are used making the website very safe for use.
NFR-3	Reliability	The tool will give accurate and reliable results most of the time.
NFR-4	Performance	The website will be well optimized which includes fast rendering of the pages, providing a bug-free, smooth and hassle-free experience for the user.
NFR-5	Availability	The tool will be available for users most of the time.
NFR-6	Scalability	The system will be scalable enough to support a lot of users at the same time while maintaining optimal performance.

CHAPTER 5

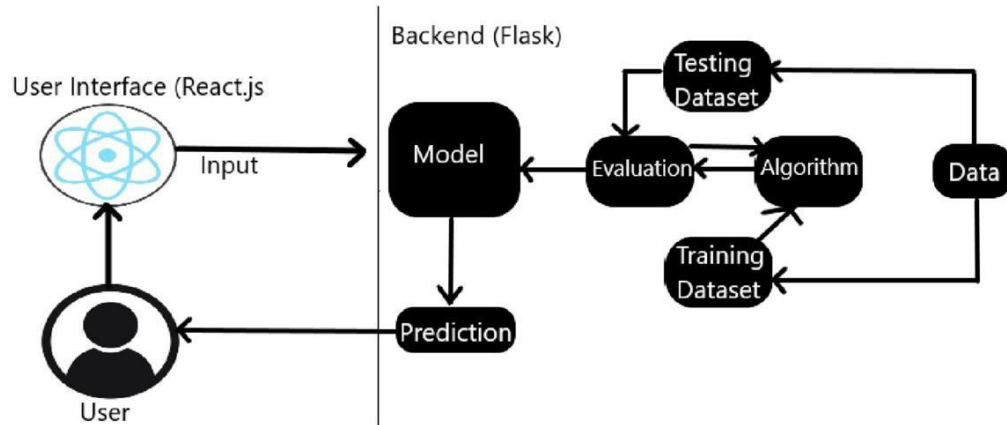
PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture Diagram:



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Outlook	I can register with Outlook Credentials	Low	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I can register with Gmail Credentials	Low	Sprint-1
	Login	USN-5	As a user, I can log into the website by entering email & password	I can access the prediction tool	High	Sprint-1
	Homepage	USN-6	As a user, I can check the treatment options	I can read the treatment details	High	Sprint-2
		USN-7	As a user, I can check the causes of heart diseases	I can read the causes of heart diseases	High	Sprint-2
		USN-8	As a user, I can check the risk factors	I can read the risk factors	High	Sprint-2
		USN-9	As a user, I can check the prevention tips	I can read the prevention tips	Medium	Sprint-2
	Prediction Tool	USN-10	As a user, I can predict if I have heart disease or not	I can get results either positive/negative	High	Sprint-3
		USN-11	As a user, I can save the results	I can save it for my account/ send results to my email	Low	Sprint-3
		USN-12	As a user, I can see the dashboard	I can save visualizations	Medium	Sprint-4
Administrator	Portal	USN-13	As an administrator, I can see the registered users	I can manage the users	High	Sprint-1
	Homepage	USN-14	As an administrator, I can update the details regarding treatments, causes and risk factors	I can check if the page is updated	High	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Manish.P, Mohammed Safwan.S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	Manish.P, Mohammed Safwan.S
Sprint-1		USN-3	As a user, I can register for the application through Outlook	4	Low	Manish.P, Mohammed Safwan.S
Sprint-1		USN-4	As a user, I can register for the application through Gmail	4	Low	Manish.P, Mohammed Safwan.S
Sprint-1	Login	USN-5	As a user, I can log into the website by entering email & password	5	High	Manish.P, Mohammed Safwan.S
Sprint-2	Homepage	USN-6	As a user, I can check the treatment options	5	High	Liju Daniel.M, Mohamed Afsal.S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2		USN-7	As a user, I can check the causes of heart diseases	5	High	Liju Daniel.M, Mohamed Afsal.S
Sprint-2		USN-8	As a user, I can check the risk factors	5	High	Liju Daniel.M, Mohamed Afsal.S
Sprint-2		USN-9	As a user, I can check the prevention tips	5	Medium	Liju Daniel.M, Mohamed Afsal.S
Sprint-3	Prediction Tool	USN-10	As a user, I can predict if I have heart disease or not	10	High	Mohammed Safwan.S, Mohamed Afsal.S
Sprint-3		USN-11	As a user, I can save the results	10	Low	Mohammed Safwan.S, Mohamed Afsal.S
Sprint-4		USN-12	As a user, I can see the dashboard	20	Medium	Manish.P, Liju Daniel.M, Mohammed Safwan.S, Mohamed Afsal.S
Sprint-1	Portal	USN-13	As an administrator, I can see the registered users	5	Low	Mohammed Safwan.S, Mohamed Afsal.S

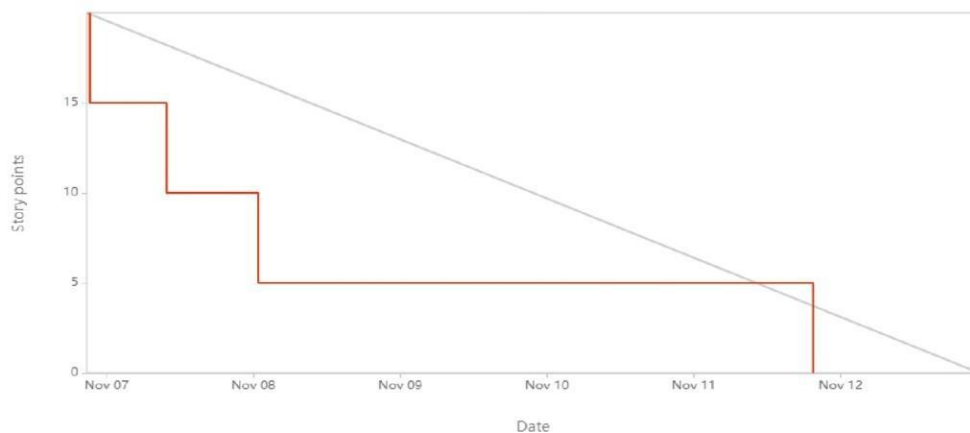
6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



Projects / ihdcardio

Backlog

MP
...
Epic
Insights

▼ HDCRD Sprint 4 11 Nov – 18 Nov (4 issues)

5
5
10
Complete sprint

HDCRD-13 Dashboard - Homepage

5 DONE

HDCRD-14 Dashboard - Logout Option

5 DONE

HDCRD-15 Dashboard - Fetching Data for Visualizations

5 IN PROGRESS

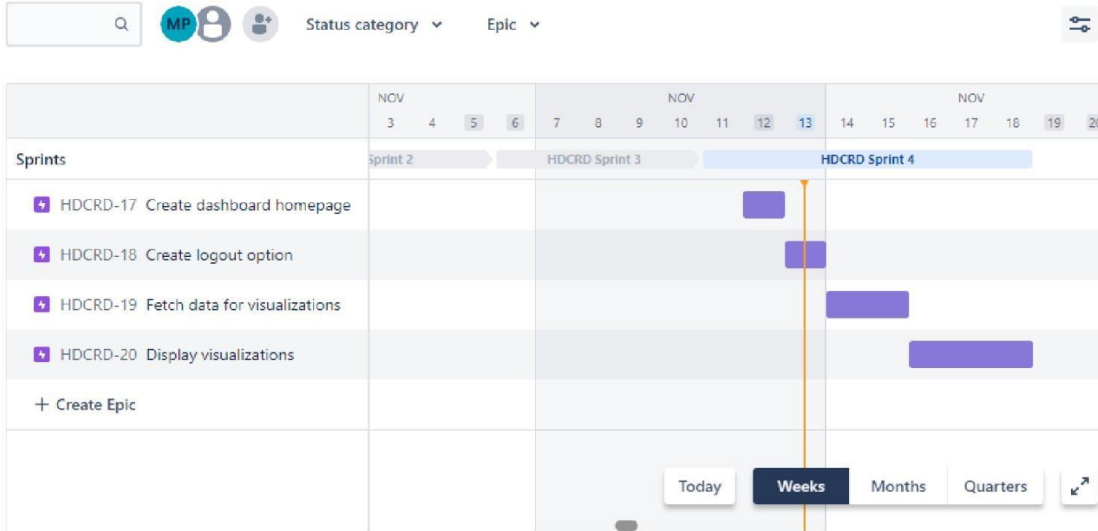
HDCRD-16 Dashboard - Visualization Pages

5 TO DO

+ Create Issue

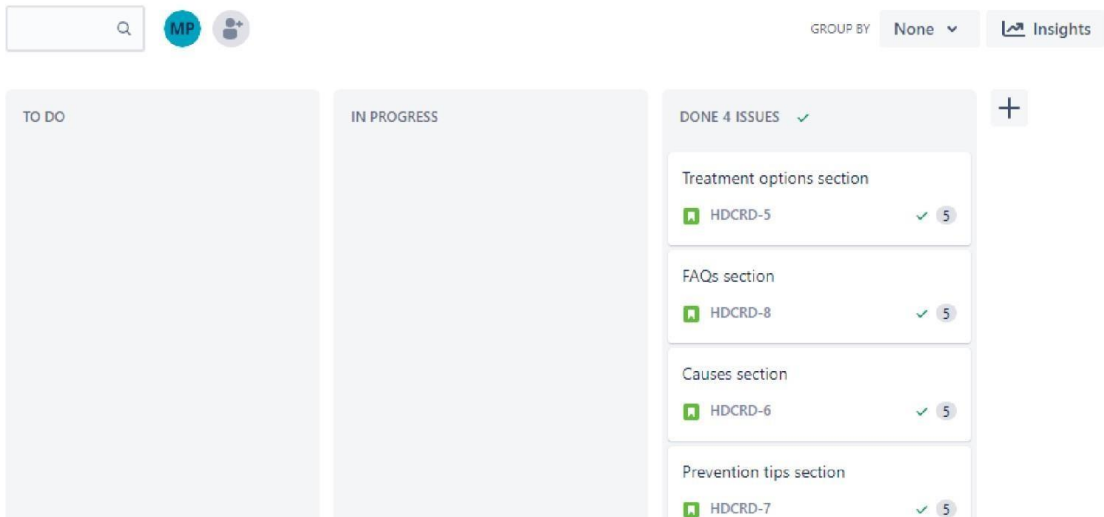
Roadmap

Give feedback Share Export ...



HDCRD Sprint 2

0 days remaining **Complete sprint** ...



CHAPTER 7

CODING & SOLUTIONING

7.1 FEATURE 1

Prediction Model: When applied to a nonlinear data set, the random forest technique performs better than the decision tree. The collection of decision trees known as a random forest was produced by several root nodes. The random forest algorithm can achieve more accuracy quickly and produce expected results.

Algorithm:

Step 1: Input the required details

Step 2: The model processes the input with the help of random forest algorithm

Step 3: The results are displayed

Code:

```
import axios from 'axios';

import React, { useState } from 'react';

import { useNavigate } from 'react-router-dom';

import './FormPageCommons.css';

function PredictionPage() {

  const [age, setAge] = useState("");

  const [sex, setSex] = useState("");

  const [chestPainType, setChestPainType] = useState("");

  const [bp, setBp] = useState("");

  const [cholesterol, setCholesterol] = useState("");

  const [fbs, setFbs] = useState("");
```

```

const [ekg, setEkg] = useState("");

const [maxHr, setMaxHr] = useState("");

const [angina, setAngina] = useState("");

const [stDepression, setStDepression] = useState("");

const [slopeOfSt, setSlopeOfSt] = useState("");

const [fluro, setFluro] = useState("");

const [thallium, setThallium] = useState("");

const navigate = useNavigate();

function predictResult(event) {

  event.preventDefault();

  if (age && sex && chestPainType && bp && cholesterol && fbs && ekg && maxHr &&
  angina && stDepression && slopeOfSt && fluro && thallium) {

    const predictionParams = { age, sex, chest_pain_type:
chestPainType, bp, cholesterol, fbs_over_120: fbs, ekg_results: ekg, max_hr:
maxHr, exercise_angina: angina, st_depression: stDepression, slope_of_st: slopeOfSt,
number_of_vessels_fluro: fluro, thallium, };

    const headers = {

      Authorization: `Bearer ${sessionStorage.getItem('token')}`,

    };

    axios.post('http://127.0.0.1:8000/predict', predictionParams, { headers })

      .then((response) => {

        predictionParams.prediction = response.data.prediction;

        [predictionParams.date, predictionParams.time] = new Date().toLocaleString().split(' ');

        sessionStorage.setItem('prediction', JSON.stringify(predictionParams));

        navigate('/results');

```

```

    }) // eslint-disable-line no-console

    .catch((ex) => console.log(ex)); // eslint-disable-line no-console
  }
}

return (

  <div className="main-container">

    <form className="main-form" onSubmit={predictResult}>

      <div className="title">Enter your details</div>

      <div className="form-input">

        <p>Age</p>

        <input type="number" name="Age" id="text-input" value={age} onChange={(e) =>
setAge(e.target.value)} />

      </div>

      <div className="form-input">

        <p>Sex</p>

        <select

          name="sex"

          id="sel-input"

          value={sex}

          onChange={(e) => setSex(e.target.value)}

        >

          <option value="-1">--Select Value--</option>

          <option value="1">Male</option>

          <option value="0">Female</option>

        </select>

```

```

</div>

<div className="form-input">

  <p>Chest Pain Type</p>

  <select

    name="chest-pain"

    id="sel-input"

    value={ chestPainType }

    onChange={ (e) => setChestPainType(e.target.value) }

    >

      <option value="-1">--Select Value--</option>

      <option value="1">Typical Angina</option>

      <option value="2">Atypical Angina</option>

      <option value="3">Non Anginal Pain</option>

      <option value="4">Asymptomatic Pain</option>

    </select>

  </div>

  <div className="form-input">

    <p>Blood Pressure</p>

    <input type="number" className="form__field" name="bp" id="text-input"
value={ bp } onChange={ (e) => setBp(e.target.value) } />

  </div>

  <div className="form-input">

    <p>Cholesterol</p>

    <input type="number" name="cholesterol" id="text-input" value={ cholesterol }
onChange={ (e) => setCholesterol(e.target.value) } />

```

```

</div>

<div className="form-input">

  <p>Fasting Blood Sugar Over 120</p>

  <select

    name="fbs"

    id="sel-input"

    value={ fbs }

    onChange={ (e) => setFbs(e.target.value) }

  >

    <option value="-1">--Select Value--</option>

    <option value="1">Yes</option>

    <option value="0">No</option>

  </select>

</div>

<div className="form-input">

  <p>EKG Results</p>

  <select

    name="ekg"

    id="sel-input"

    value={ ekg }

    onChange={ (e) => setEkg(e.target.value) }

  >

    <option value="-1">--Select Value--</option>

    <option value="0">Normal</option>

```

`<option value="1">Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of greater than 0.05 mV)</option>`

`<option value="2">Showing probable or definite left ventricular hypertrophy by Estes`'` criteria</option>`

`</select>`

`</div>`

`<div className="form-input">`

`<p>Maximum Heart Rate</p>`

`<input type="number" name="mhr" id="text-input" value={ maxHr } onChange={(e) => setMaxHr(e.target.value)} />`

`</div>`

`<div className="form-input">`

`<p>Exercise Induced Angina</p>`

`<select`

`name="angina"`

`id="sel-input"`

`value={ angina }`

`onChange={ (e) => setAngina(e.target.value) }`

`>`

`<option value="-1">--Select Value--</option>`

`<option value="1">Yes</option>`

`<option value="0">No</option>`

`</select>`

`</div>`

`<div className="form-input">`

```

    <p>ST Depression</p>

    <input type="number" name="stdep" id="text-input" value={ stDepression}
onChange={ (e) => setStDepression(e.target.value)} />

</div>

<div className="form-input">

    <p>Slope of ST</p>

    <select

        name="slope-st"

        id="sel-input"

        value={ slopeOfSt }

        onChange={ (e) => setSlopeOfSt(e.target.value)}

    >

        <option value="-1">--Select Value--</option>

        <option value="1">Upsloping</option>

        <option value="2">Flat</option>

        <option value="3">Downsloping</option>

    </select>

</div>

<div className="form-input">

    <p>Number of major vessels colored by Flouroscope</p>

    <select

        name="fluro"

        id="sel-input"

        value={ fluro }

        onChange={ (e) => setFluro(e.target.value)}

```

```

>

<option value="-1">--Select Value--</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3">3</option>

</select>

</div>

<div className="form-input">

  <p>Thallium</p>

  <select

    name="thallium"

    id="sel-input"

    value={ thallium }

    onChange={ (e) => setThallium(e.target.value) }

  >

    <option value="-1">--Select Value--</option>

    <option value="3">Normal</option>

    <option value="6">Fixed Defect</option>

    <option value="7">Reversible Defect</option>

  </select>

</div>

<button type="text" className="submit">Submit</button>

</form>

```



```

    </div>

  );
}

export default PredictionPage;

```

7.2 FEATURE 2

Dashboard: Our application helps the user in finding out if they have heart disease or not. They can find out by entering details such as their heart rate, cholesterol, blood pressure etc. A dashboard is also attached along with the results for better understanding where they can compare their blood pressure and similar metrics with other users.

Code:

```

import Sidebar from "../../components/sidebar/Sidebar"

import "../home.scss"

import "../Cards.css"

const DashboardHome = () => {

  return (

    <div className="home">

      <Sidebar />

      <div className="homeContainer">

        <h1>Welcome to your Dashboard!</h1>

        <br />

        <br />

        <br />

```

```

<h3>Check out your email. Your results have been sent there.</h3>

<br />

<br />

<h3>Here you can check out different kind of visualizations to get a general idea about
the factors increasing causes of getting a cardiac arrest.</h3>

<br />

<div class="row">

  <div class="col-md-4 col-xl-3">

    <div class="card bg-c-blue order-card">

      <div class="card-block">

        <h2      class="text-right"><i      class="fa      fa-cart-plus      f-
left"></i><span>87%</span></h2>

        <p class="m-b-0">Accuracy</p>

      </div>

    </div>

  </div>

</div>

<div class="col-md-4 col-xl-3">

  <div class="card bg-c-green order-card">

    <div class="card-block">

      <h2 class="text-right"><i class="fa fa-cart-plus f-left"></i><span>5</span></h2>

      <p class="m-b-0">Visualization Types</p>

    </div>

  </div>

</div>

<div class="col-md-4 col-xl-3">

```

```

<div class="card bg-c-yellow order-card">

  <div class="card-block">

    <h2 class="text-right"><i class="fa fa-cart-plus f-left"></i><span>Random Forest
Classifier</span></h2>

    <p class="m-b-0">ML Model</p>

  </div>

</div>

</div>

</div>

</div>

</div>

</div>

)
}

export default DashboardHome

```

7.3 FEATURE 3

Login

Algorithm :

1. Input the credentials (email and password).
2. If already logged in user is taken to home page
3. Else, check for validity of credentials
4. If wrong credentials entered , notification is displayed to user and user stays in login page.
5. On correct credentials, user is taken to home page.

Code:

```
import axios from 'axios';

import React, { useState } from 'react';

import './FormPageCommons.css';

import { useNavigate } from 'react-router-dom';

const Login = () => {

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const navigate = useNavigate();

  function loginUser(event) {

    event.preventDefault();

    const userDetails = {

      email, password,

    };

    // console.log(userDetails); // eslint-disable-line no-console

    if (userDetails && userDetails.email && userDetails.password) {

      axios.post('http://127.0.0.1:8000/login', userDetails)

        .then((response) => {

          sessionStorage.setItem('token', response.data.token);

          navigate('/predict');

        })

        .catch((ex) => {

          // console.log(ex); // eslint-disable-line no-console

          // const error = JSON.parse(ex);
```

```

    if (ex.response && ex.response.status && ex.response.status === 404) {

        alert('User not found'); // eslint-disable-line no-alert

    } else {

        console.log(ex); // eslint-disable-line no-console

    }

    });

} else {

    alert('Please enter valid credentials'); // eslint-disable-line no-alert

}

}

return (

    <div className="main-container">

        <form className="main-form" onSubmit={loginUser}>

            <div className="form-input">

                <p>e-mail</p>

                <input type="email" name="email" id="email" value={email} onChange={(e) =>
setEmail(e.target.value)} />

            </div>

            <div className="form-input">

                <p>Password</p>

                <input type="password" name="password" id="password" value={password}
onChange={(e) => setPassword(e.target.value)} />

            </div>

            <div className="button-container">

                <input type="submit" />

```

```

        </div>

    </form>

</div>

);

};

export default Login;

```

7.4 FEATURE 4

Signup

Algorithm:

1. Input the signup form fields (name , email , password , re-enter password).
2. All credentials are validated.
3. Website checks whether the given email exists in the database.
4. If already registered, notification is displayed. Or else, the user is taken to the login page.

Code:

```

import axios from 'axios';

import React, { useState } from 'react';

import { useNavigate } from 'react-router-dom';

import { Link } from 'react-router-dom';

const Register = () => {

    const [name, setName] = useState("");

```

```

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const [confPassword, setConfPassword] = useState("");

const navigate = useNavigate();

function registerUser(event) {

  event.preventDefault();

  const userDetails = {

    fullName: name, email, password,

  };

  if (userDetails && userDetails.fullName && userDetails.password && userDetails.email)
  {

    // console.log('Hi'); // eslint-disable-line no-console

    axios.post('http://127.0.0.1:8000/register', userDetails)

      .then(() => {

        // console.log(response.json); // eslint-disable-line no-console

        navigate('/login');

      })

      .catch((ex) => console.log(ex)); // eslint-disable-line no-console

  }

}

return (

  <div className="main-container">

    <form className="login-form" onSubmit={registerUser}>

      <div className="form-input">

        <p>Name</p>

```

```

    <input type="text" name="text" id="text" value={name} onChange={(e) =>
setName(e.target.value)} />

</div>

<div className="form-input">

    <p>e-mail</p>

    <input type="email" name="email" id="email" value={email} onChange={(e) =>
setEmail(e.target.value)} />

</div>

<div className="form-input">

    <p>Password</p>

    <input type="password" name="password" id="password" value={password}
onChange={(e) => setPassword(e.target.value)} />

</div>

<div className="form-input">

    <p>Confirm Password</p>

    <input type="password" name="password" id="conf-password" value={confPassword}
onChange={(e) => setConfPassword(e.target.value)} />

</div>

<div className="button-container">

    <input type="submit" />

    <p>Already have an account?</p>

    <Link to="/login">

        <button type="button">Login</button>

    </Link>

</div>

```



```
    </form>

  </div>

);

};

export default Register;
```

7.5 DATABASE SCHEMA

NoSQL databases like MongoDB offer high performance, high availability, and easy scalability. MongoDB is a document-oriented database which stores data in JSON-like documents with dynamic schema. It means you can store your records without worrying about the data structure such as the number of fields or types of fields to store values. MongoDB documents are similar to JSON objects. Details like name, e-mail, password of the registered user are stored so that when the user tries to login, authentication takes place and the user is logged in.

CHAPTER 8

TESTING

8.1 TEST CASES

Testcase 1: Logging in with registered login details.

Testcase 2: Logging in with invalid login details.

Testcase 3: Registering with existing user's details.

Testcase 4: Entering wrong values while filling medical related details.

Testcase 5: Producing visualisations for given input.

8.2 USER ACCEPTANCE TESTING

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/signup popup when user clicked on My account button	Stable internet connection, Compatible browser, Login credentials	1. Enter URL and click go 2. Click on My Account dropdown button 3. Verify login/signup popup displayed or not	https://shopenzer.com/	Login/signup popup should display	Working as expected	Pass	Login page displayed successfully	N	1
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/signup popup	Proper code for UI elements, Elements position, Buttons and Textbox response	1. Enter URL and click go 2. Click on My Account dropdown button 3. Verify login/signup popup with below UI elements: a. email text box b. password text box c. Login button d. New customer? Create account link e. Last password? Recovery password link	https://shopenzer.com/	Application should show below UI elements: a. email text box b. password text box c. Login button with orange colour d. New customer? Create account link e. Last password? Recovery password link	Working as expected	Fail	Elements are displayed successfully but recovery password button is not present	N	2
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with valid credentials	User credentials, Database with credentials of existing users	1. Enter URL(https://shopenzer.com/) and click go 2. Click on My Account dropdown button 3. Enter Valid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	Logged in successfully	N	3

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with invalid credentials	User credentials, Database with credentials of existing users	1. Enter URL(https://shopenzer.com/) and click go 2. Click on My Account dropdown button 3. Enter invalid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button	Username: chalam@gmail.com password: Testing123	Application should show "incorrect email or password" validation message.	Working as expected	Fail	Login failed due to incorrect login details or user not registered	N	4
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with invalid credentials	User credentials, Database with credentials of existing users	1. Enter URL(https://shopenzer.com/) and click go 2. Click on My Account dropdown button 3. Enter Valid username/email in Email text box 4. Enter invalid password in password text box 5. Click on login button	Username: chalam@gmail.com password: Testing123678686786876876	Application should show "incorrect email or password" validation message.	Working as expected	Fail	Login failed due to incorrect login details or user not registered	N	5
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with invalid credentials	User credentials, Database with credentials of existing users	1. Enter URL(https://shopenzer.com/) and click go 2. Click on My Account dropdown button 3. Enter invalid username/email in Email text box 4. Enter invalid password in password text box	Username: chalam@gmail.com password: Testing123678686786876876	Application should show "incorrect email or password" validation message.	Working as expected	Fail	Login failed due to incorrect login details or user not registered	N	6

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

- Hours worked: 50 hours
- Stick to Timelines: 100%
- Stay within budget: 100%
- Consistency of the product: 85%
- Efficiency of the product: 85%
- Quality of the product: 85%

CHAPTER 10

ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Smooth User Interface
- Accuracy is achieved quickly

DISADVANTAGES:

- Random forest can be used for both classification and regression tasks, but it is not more suitable for Regression tasks.

CHAPTER 11

CONCLUSION

This overview of the project conveys the idea that numerous methods have been investigated for diagnosing cardiovascular disease. Big data, machine learning, and data mining can be used to great success to analyse the prediction model with the highest degree of accuracy. The primary goal of this project is to diagnose cardiovascular disease or heart disease utilizing a variety of techniques and procedures to obtain a prognosis.

CHAPTER 12

FUTURE SCOPE

A future update shall comprise of section for viewing renowned cardiologists and scan centres in their city. The obtained output can be further processed and sent to smart devices to provide necessary assistance. Constant monitoring can provide necessary data to recommend to consult a doctor in case of an emergency.

CHAPTER 13

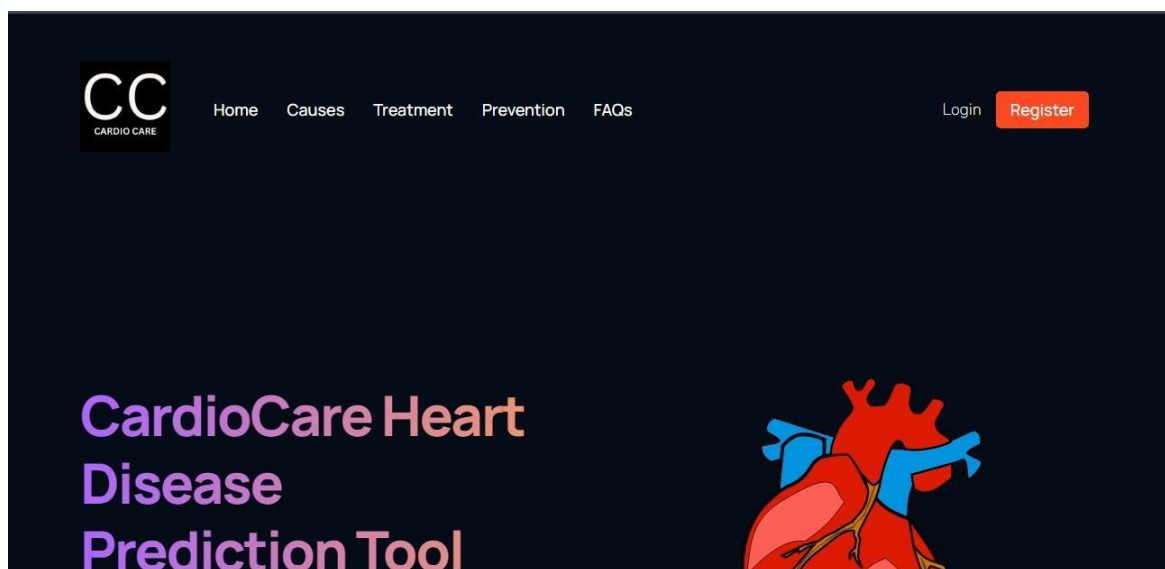
APPENDIX

PROJECT DEMONSTRATION LINK:

https://drive.google.com/file/d/1q7tOP9nDfifH4nYbaZsZA9IhbLpLJY0W/view?usp=share_link

GITHUB LINK: <https://github.com/IBM-EPBL/IBM-Project-29395-1660125130>

APPENDIX A1: SCREENSHOTS



Here are some treatment options for heart diseases.

Heart disease treatment depends on the cause and type of heart damage. Healthy lifestyle habits – such as eating a low-fat, low-salt diet, getting regular exercise and good sleep, and not smoking – are an important part of treatment.

Medications

If lifestyle changes alone do not work, medications may be needed to control heart disease symptoms and to prevent complications. The type of medication used depends on the type of heart disease.

Coronary artery bypass surgery

This allows blood flow to reach a part of the heart when an artery is blocked. Coronary artery bypass grafting is the most common surgery. A surgeon can use a healthy blood vessel from another part of the body to repair a blocked one.

Coronary angiography

This is a procedure that widens narrow or blocked coronary arteries. It is often combined with the insertion of a stent, which is a wire-mesh tube that allows easier blood flow.

Valve replacement or repair

A surgeon can replace or repair a valve that is not functioning correctly.

Repair surgery

A surgeon can repair congenital heart defects, aneurysms, and other problems.



Already have an account?



Enter your details

Age

Sex

Chest Pain Type

Blood Pressure

Cholesterol

Fasting Blood Sugar Over 120



Based on the parameters provided, our tool indicates there may be a chance of Presence of heart disease. Do consult with a cardiologist for more information.