IBM NALAIYA THIRAN 2022-23 PROJECT REPORT

VISUALIZING AND PREDICTING HEART DISEASE WITH AN INTERACTIVE DASHBOARD

TEAM ID - PNT2022TMID02476

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Predicts the chances of Heart Disease and classifies patient's risk level. Find Number of people suffering from heart disease and classifying with genders,using data analytics to improve diagnosis.Chest pain is the key to recognize the heart disease. In this work, the heart diseases are predicted by considering major factors with four types of chest pain.The role of exploratory data using tableau provided a visual appealing and accurate clustering experience.

2. LITERATURE SURVEY

Bo Jin, Chao Che et al. (2018) proposed a “Predicting the Risk of Heart Failure With EHR Sequential Data Modeling” model designed by applying neural network. This paper used the electronic health record (EHR) data from real-world datasets related to congestive heart disease to perform the experiment and predict the heart disease before itself. We tend to used one-hot encryption and word vectors to model the diagnosing events and foretold coronary failure events victimization the essential principles of an extended memory network model. By analyzing the results, we tend to reveal the importance of respecting the sequential nature of clinical records

[1]

Aakash Chauhan et al. (2018) presented “Heart Disease Prediction using Evolutionary Rule Learning”. This study eliminates the manual task that additionally helps in extracting the information (data) directly from the electronic records. To generate strong association rules, we have applied frequent pattern growth association mining on patient’s dataset. This will facilitate (help) in decreasing the amount of services and shown that overwhelming majority of the rules helps within the best prediction of coronary sickness [2].

Ashir Javeed, Shijie Zhou et al. (2017) designed “An Intelligent Learning System based on Random Search Algorithm and Optimized Random Forest Model for Improved Heart Disease Detection”. This paper uses random search algorithm (RSA) for factor selection and random forest model for diagnosing the cardiovascular disease. This model is principally optimized for using grid search algorithmic program.Two forms of experiments are used for cardiovascular proposed Random Search Algorithm based random forest model is developed. This methodology is efficient and less complex than conventional random forest model. Comparing to conventional random forest it produces 3.3% higher accuracy. The proposed learning system can help the physicians to improve the quality of heart failure detection.[3]

Aakash Chauhan et al. (2018) presented “Heart Disease Prediction using Evolutionary Rule Learning”. This study eliminates the manual task that additionally helps in extracting the information (data) directly from the electronic records. To generate strong association rules, we have applied frequent pattern growth association mining on patient’s dataset. This will facilitate (help) in decreasing the amount of services and shown that overwhelming majority of the rules helps within the best prediction of coronary sickness [4].

Two forms of experiments are used for cardiovascular disease prediction. In the first form, only random forest model is developed and within the second experiment the proposed Random Search Algorithm based random forest model is developed. This methodology is efficient and less complex than conventional random forest model. Comparing to conventional random forest it produces 3.3% higher accuracy. The proposed learning system can help the physicians to improve the quality of heart failure detection.

K.Prasanna Lakshmi, Dr. C.R.K.Reddy (2015) designed “Fast Rule-Based Heart Disease Prediction using Associative Classification Mining”. In the proposed Stream Associative Classification Heart Disease Prediction (SACHDP), we used associative classification mining over landmark window of data streams. This paper contains two phases: one is generating rules from associative classification mining and next one is pruning the rules using chi-square testing and arranging the rules in an order to form a classifier. Using these phase to predict the heart disease easily [5].

“Prediction and Diagnosis of Heart Disease by Data Mining Techniques” designed by Boshra Bahrami, Mirsaeid Hosseini Shirvani. This paper uses various classification methodology for diagnosing cardiovascular disease. Classifiers like KNN, SVO classifier and Decision Tree are used to divide the datasets. Once the classification and performance evaluation the Decision tree is examined as the best one for cardiovascular disease prediction from the dataset[6].

2.1 PROBLEM STATEMENT DEFINITION

Using various statistical test for feature selection and to find out the most efficient classification algorithm that can help us to detect heart diseases at early stage.This algorithm can be used on heart records of the patient or by using it on classification reports.

3.IDEATION AND PROPOSED SOLUTION

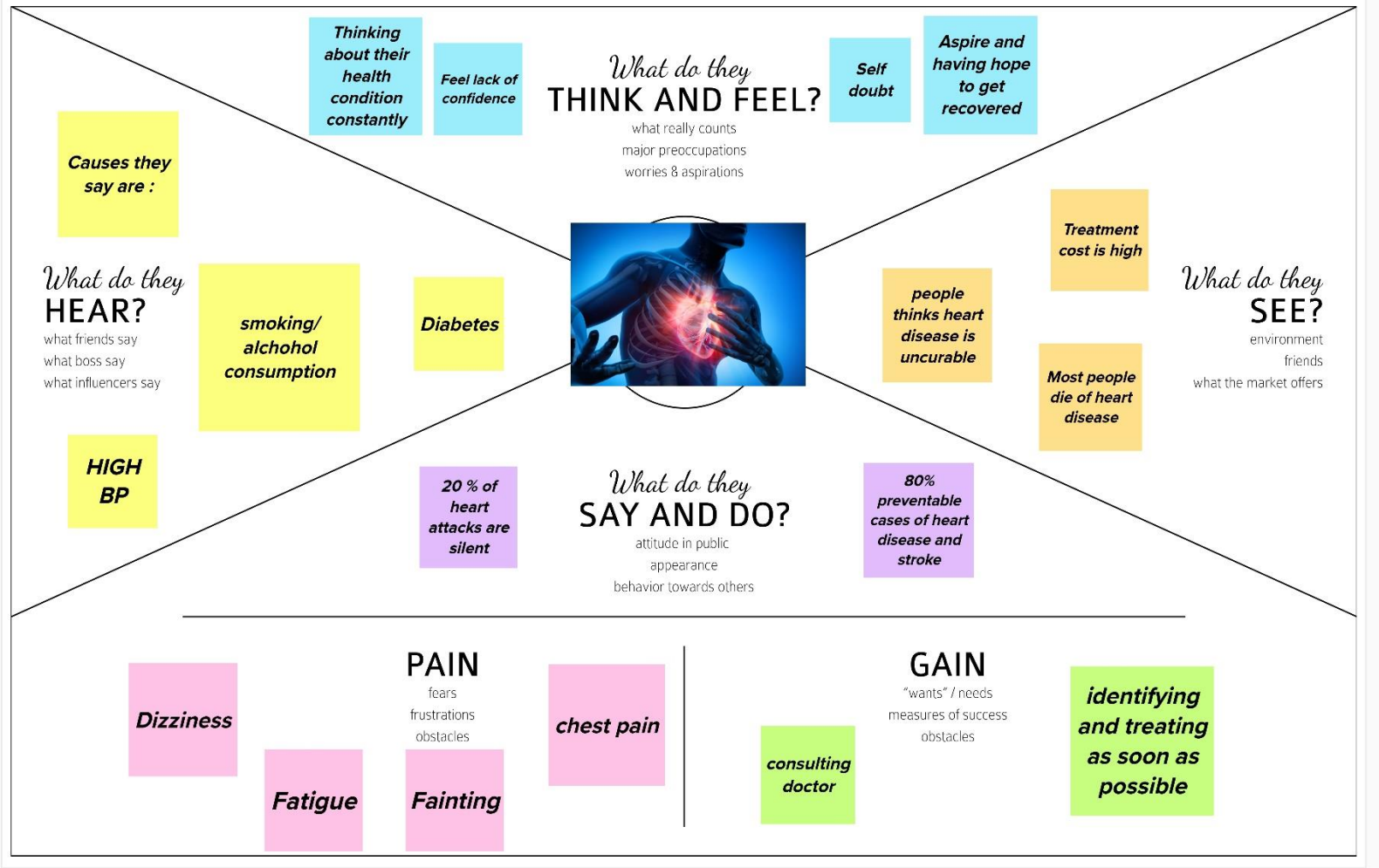
3.1 EMPATHY MAP CANVAS

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/blob/main/project%20design%20and%20planning/ideation%20phase/heart%20disease%20empathy%20map.pdf

3.2 Ideas and Techniques

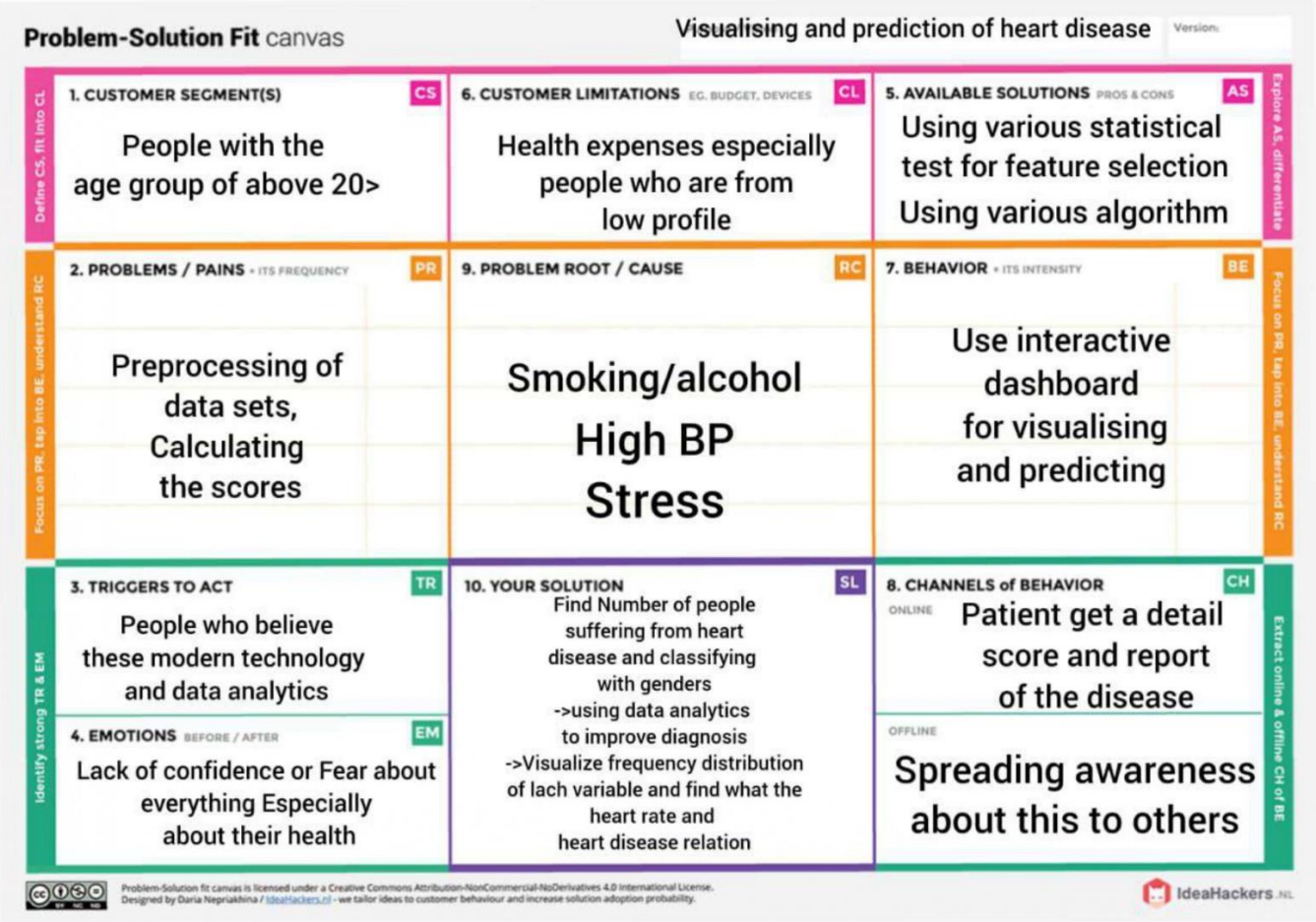
https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/blob/main/project%20design%20and%20planning/ideation%20phase/ideas%20and%20techniques.pdf

EMPATHY MAP :



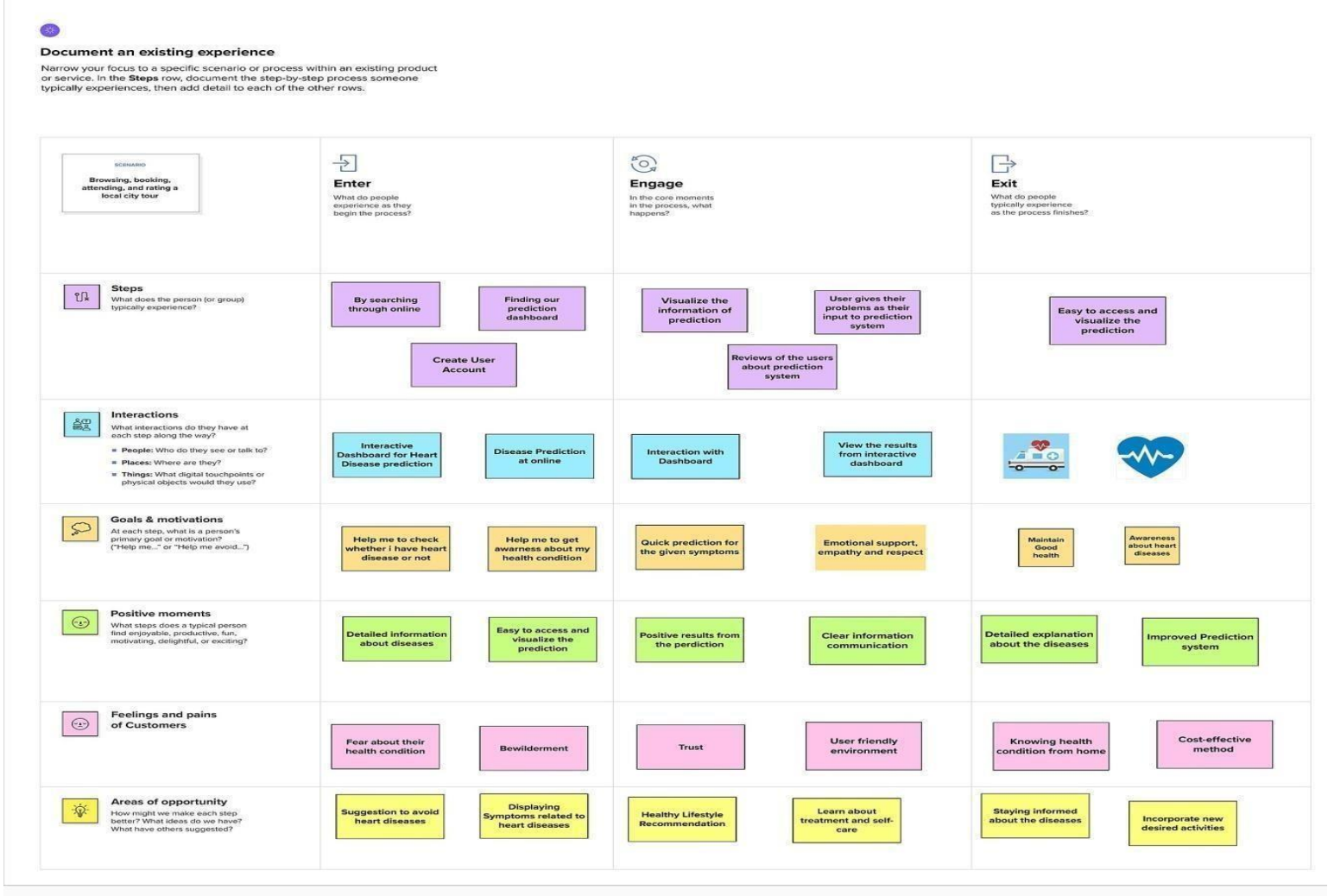
3.3 Proposed solution Fit :

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/blob/main/project%20design%20and%20planning/project%20design%20phase%201/proposed%20solution%20fit.pdf

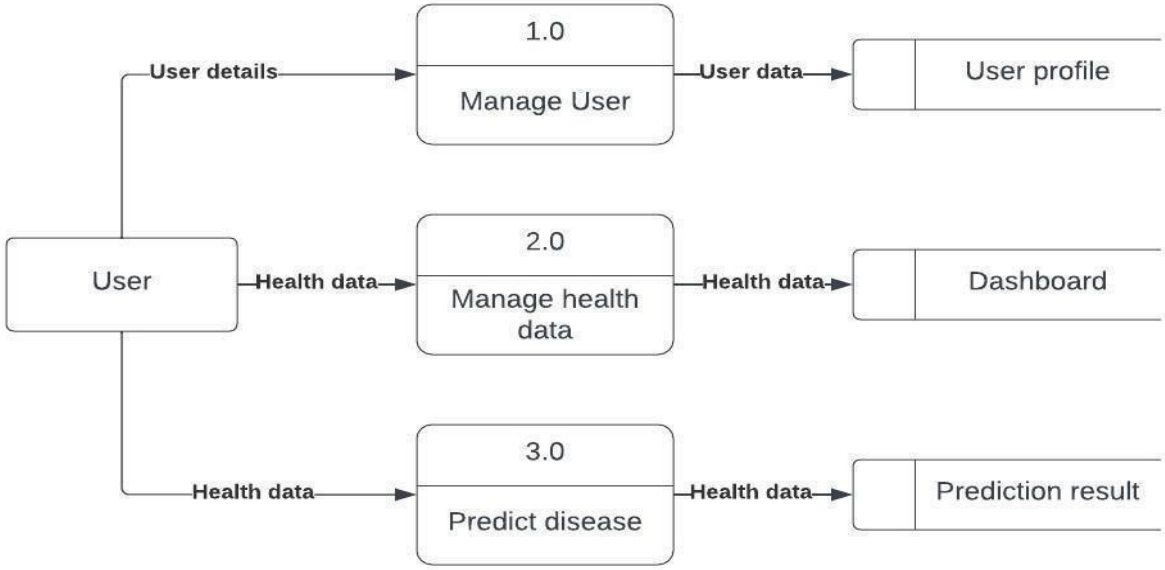


4.CUSTOMER JOURNEY

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/blob/main/project%20design%20and%20planning/project%20design%20phase%20II/Customer%20Journey%20Map%20-%20Visualizing%20and%20Predicting%20Heart%20Diseases%20with%20an%20Interactive%20Dash%20Board.pdf



4.1 DATA FLOWS AND USER STORIES



4.2 Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Flow:

1) User creates an account in the application.
2) User enters the medical records in the dashboard.
3) User can view the visualizations of trends in the form of graphs and charts for his/her medical records with the trained dataset.
4) User can view the accuracy of probability of occurrence of heart disease in the dashboard.

User Stories:

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	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 / Dashboard.	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
	Login	USN-3	As a user, I can log into the application by entering email & password.	I can access my account / Dashboard when logged in.	High	Sprint-1
Customer (Web user)	Dashboard	USN-4	User can view his/her complete medical analysis and accuracy of disease prediction.	I can view my medical analysis in the dashboard.	High	Sprint-2

		USN-5	User can view the accuracy of occurrence of heart disease.	I can view the accuracy of heart disease in the dashboard.	High	Sprint-2
Customer Care Executive	Helpdesk	USN-6	As a customer care executive, he/she can view the customer queries.	I can post my queries in the dashboard.	Medium	Sprint-3
		USN-7	As a customer care executive, he/she can answer the customer queries.	I can get support from helpdesk.	High	Sprint-3
Administrator	User Profile	USN-8	As an admin, he/she can update the health details of users.	I can view my updated health details.	High	Sprint-4

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-9	As an admin, he/she can add or deleteusers.	I can access my account / Dashboard when loggedin.	High	Sprint-4
		USN-10	As an admin, he/she can manage the userdetails.	I can view the organizeddata of myself.	High	Sprint-4

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/blob/main/project%20design%20and%20planning/project%20design%20phase%20II/Data%20Flow%20Diagrams%20and%20User%20Stories%20-%20Visualizing%20and%20Predicting%20Heart%20Diseases%20with%20an%20Interactive%20Dash%20Board.pdf

5.FUNCTIONAL REQUIREMENT

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Enables user to make registration for the applicationthrough Gmail.

FR-2	User Confirmation	Once after registration, the user will get confirmationvia E-mail.
FR-3	Visualizing Data	User can visualize the trends on the heart diseasethrough Dashboard created using IBM Cognos Analytics.
FR-4	Generating Report	User can view his/her health report and can makedecisions accordingly.

Non-functional Requirements:

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

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application will have a simple and user-friendly graphical interface. Users will be ableto understand and use all the features of the application easily. Any action has to be performed with just a few clicks.
NFR-2	**Security**	For security of the application the techniqueknown as database replication should be used, so that all the important data should be kept safe. In case of crash, the system should be able to backup and recover the data.

NFR-3	**Reliability**	The application has to be consistent at everyscenario and has to work without failure in any environment.
NFR-4	**Performance**	Performance of the application depends on the response time and the speed of the data submission. The response time of the applicationis direct and faster which depends on the efficiency of implemented algorithm.
NFR-5	**Availability**	The application has to be available 24 x 7 for userswithout any interruption.
NFR-6	**Scalability**	The application can withstand the increase in theNumber of users and has to be able to develop higher versions.

6.PROJECT DEVELOPMENT PHASE

6.1 SPRINT 1

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/tree/main/project%20development%20phase/sprint%201

6.2 SPRINT 2

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/tree/main/project%20development%20phase/sprint%202

6.3 SPRINT 3

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/tree/main/project%20development%20phase/sprint%203

6.4 SPRINT 4

https://github.com/IBM-EPBL/IBM-Project-24635-1659946282/tree/main/project%20development%20phase/sprint%204

7.CODING AND SOLUTION

HOMEPAGE

16 lines (14 sloc) | 276 Bytes

Raw Blame

```jsx
import React from 'react';
import { Causes, Treatment, Prevention, FAQS, Header } from '../containers';

function HomePage() {
  return (
    <>
      <Header />
      <Causes />
      <Treatment />
      <Prevention />
      <FAQS />
    </>
  );
}

export default HomePage;
```

LOGIN

```jsx
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
    }
```

```
        .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;
```

RESULT

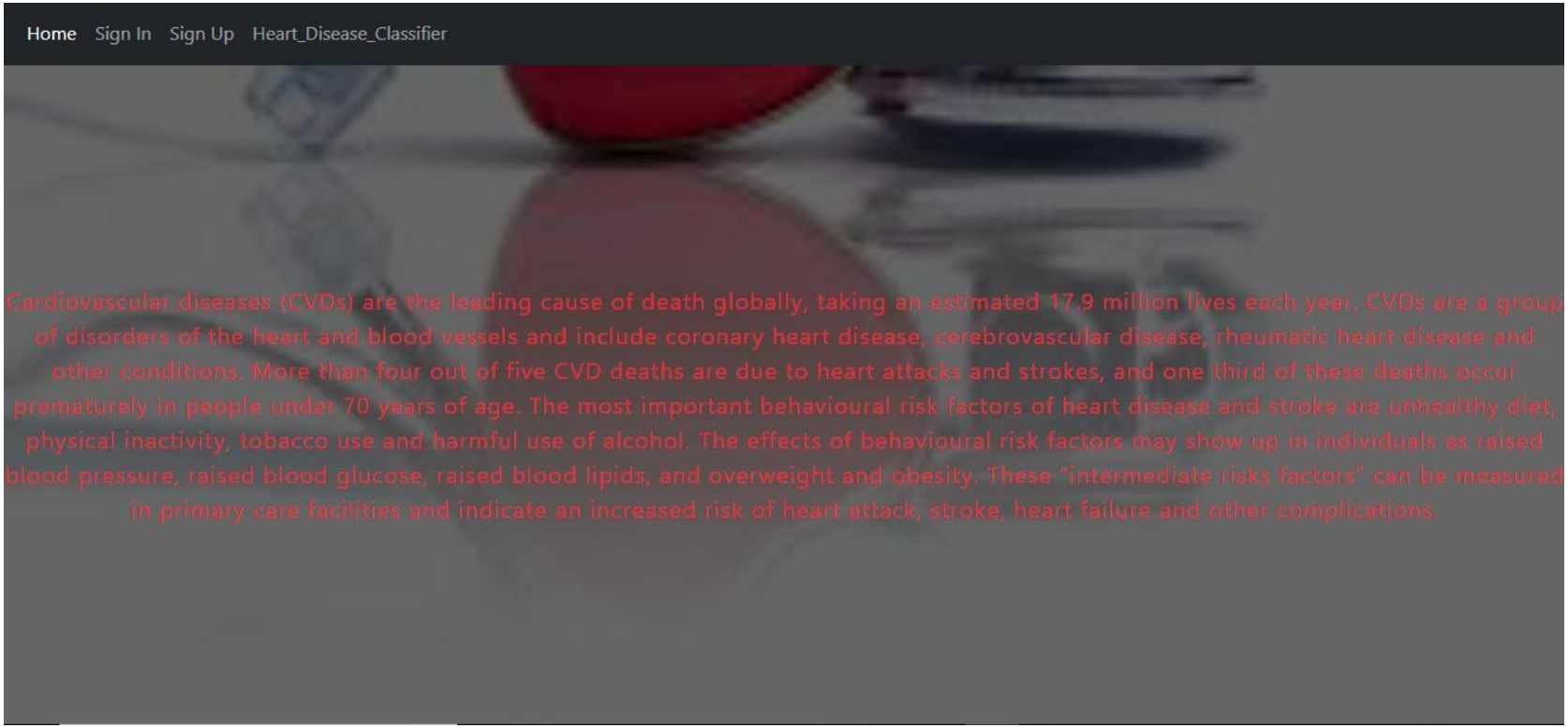
38 lines (34 sloc) | 1.19 KB

Raw Blame

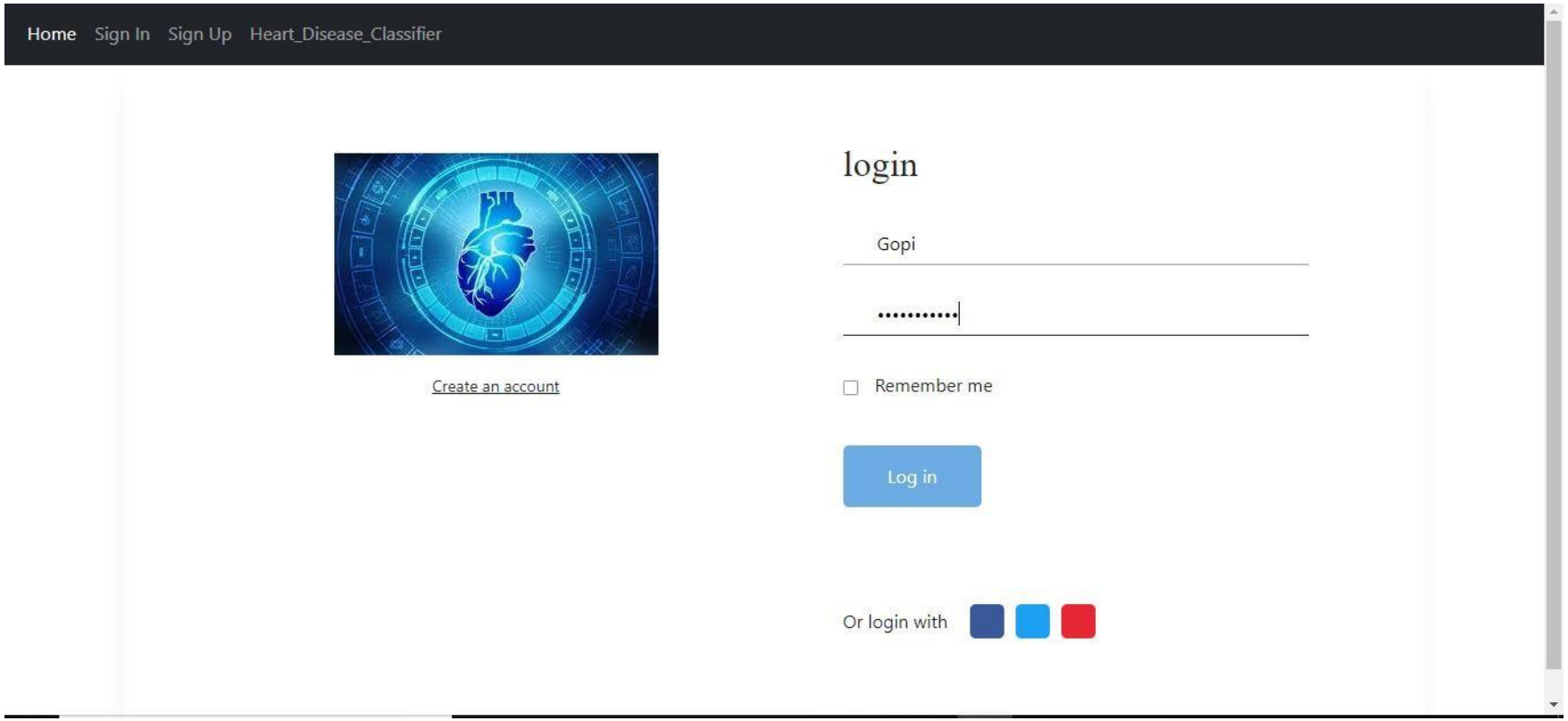
```jsx
import axios from 'axios';
import React, { useState } from 'react';
import './Results.css';
import { Link } from 'react-router-dom';


function Results() {
  const predictionObject = JSON.parse(sessionStorage.getItem('prediction'));
  const [saveState, setSaveState] = useState(false);
  const headers = {
    Authorization: `Bearer ${sessionStorage.getItem('token')}`,
  };
  function saveResult() {
    axios.put('http://127.0.0.1:8000/save_result', predictionObject, { headers })
      .then(() => setSaveState(true))
      .catch((ex) => console.log(ex)); // eslint-disable-line no-console
  }

  return (
    <div className="main">
      <h1>Based on the parameters provided, our ML model has predicted the {predictionObject.prediction} of heart disease</h1>
      <div className="button-container">
        {saveState ? (<h3>Your result has been successfully saved</h3>)
          : (
            <>
              <h3>Do you want to save the results?</h3>
              <button type="submit" onClick={saveResult}>Save</button>
            </>
          )}
        <Link to="/dashboard">
          <button type="submit">Dashboard</button>
        </Link>
      </div>
    </div>
  );
}

export default Results;
```

8.Output:

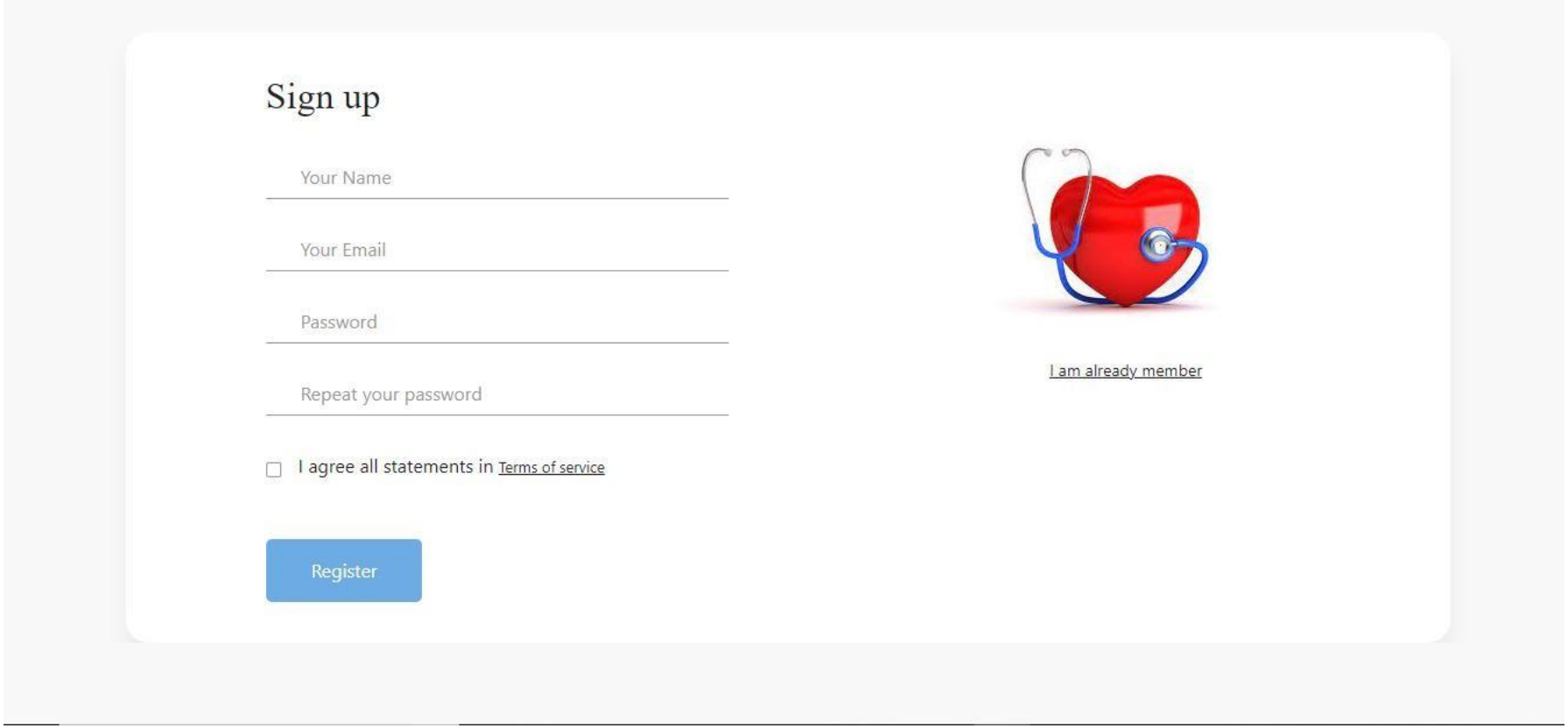
Home.html:



Sign in.html:


Home
Sign In
Sign Up
Heart_Disease_Classifier
Create an account
login
Gopi
Remember me
Log in
Or login with

SignUp.html:



Heart_Disease_Classifier.html

Heart Disease Test Form

Age: 35

Sex: Male

Chest Pain Type: Atypical Angina

Resting Blood Pressure in mm Hg: 23

Serum Cholestoral in mg/dl: 456

Fasting Blood Sugar > 120 mg/dl: False

Resting ECG Results: Having ST-T wave abnormal

Maximum Heart Rate: 134

Exercise Induced Angina: Yes

ST Depression Induced: 67

Slope of the Peak Exercise ST Segment: Downsloping

Number of Vessels Colored by Flourosopy: 2

Thalassemia: Normal

Result