IBM – NALAIYA THIRAN PROJECT

HEART DISEASE VISUALISATION WITH AN INTERACTIVE DASHBOARD

INDUSTRY MENTOR : MAHIDHAR SAUMYA

FACULTY MENTOR : KOUSIKA N

TEAM ID	**:**	PNT2022TMID02838
TEAM LEAD	**:**	VARUNA K
TEAM MEMBER	**:**	SUBASHINI K
TEAM MEMBER	**:**	VIGNESHKANNA R
TEAM MEMBER	**:**	SURYAPRAKASH K

TABLE OF THE CONTENT

CHAPTER-1

INTRODUCTION

1.1 PROJECT OVERVIEW

Machine Learning is a way of Manipulating and extraction of implicit, previously unknown/known and potentially useful information about data. Machine Learning is a very vast and diverse field and its scope and implementation is increasing day by day. Machine learning Incorporates various classifiers of Supervised, Unsupervised and Ensemble Learning which are used to predict and Find the Accuracy of the given dataset. We can use that knowledge in our project of Heart Disease Prediction as it will help a lot of people. Cardiovascular diseases are very common these days, they describe a range of conditions that could affect your heart. The World health organization estimates that 17.9 million global deaths from (Cardiovascular diseases) CVDs . It is the primary reason for deaths in adults. Our project can help predict the people who are likely to be diagnosed 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. This project focuses on mainly three data mining techniques namely: (1) Logistic regression, (2) KNN and (3) Random Forest Classifier. The accuracy of our project is 87.5% which is better than the previous system where only one data mining technique was used. So, using more data mining techniques increased the HDPS accuracy and efficiency. Logistic regression falls under the category of supervised learning. Only discrete values are used in logistic regression. The objective of this project is to check whether the patient is likely to be diagnosed with any cardiovascular heart diseases based on their medical attributes such as gender, age, chest pain, fasting sugar level, etc. A dataset is selected from the Kaggle. By using this dataset, we predict whether the patient can have a heart disease or not. To predict this, we use 13 medical attributes of a patient and classify him if the patient is likely to have a heart disease. These medical attributes are trained under three algorithms: K Nearest Neighbour Classifier, Support Vector Classifier, Decision Tree Classifier and Random Forest Classifier. I varied parameters across each model to improve their scores. In the end, K Nearest Neighbors Classifier achieved the highest score of 87% with 8 nearest neighbors.

1.2 PURPOSE

As we all know, the heart is the most important part of our body other than the brain.It pumps blood through the blood vessels of the circulatory system. The circulatory system is extremely important because it transports blood, oxygen and other materials to the different organs of the body. Heart plays the most crucial role in the circulatory system. If the heart does not function properly then it will lead to serious health conditions including death.For having a healthy heart, there are many solutions available in the market. Exercisecan also play an important role for maintaining heart health. Apart from medical treatments, technology can also prove to be very useful in treating any heart disease. Any heart disease is predicted beforehand, then curing it would be not much complex. But predicting it would be a tough task. Medical science has made excellent use of technological breakthroughs to raise the standard of healthcare These technological developments have opened the path for precise illness diagnosis and prognosis Machine learning might be a great option for you obtain a high level of accuracy when it comes to forecasting heart illnesses with the help of algorithms.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Before we did the experiments, we did research on how people explored heart disease prediction so that we can broaden our horizons and learn from them. In 2011, Ujma Ansari[1] made use of the Decision Tree model to predict heart disease and get a high accuracy of 99%, which inspires us to use a better version of Decision Tree and it is Random Forest. Unfortunately, the paper uses a dataset with 3000 instances but does not provide a reference of how they get the data. The UCI website only provides 303 instances of dataset so we doubt where the author gets 3000 instances of dataset. In 2012, Chaitrali S. Dangare [2] made the prediction by using three models and such models are Naïve Bayes, Decision Trees and Neural Network. We are using the same dataset as he did. The difference between his work and ours is that he added 2 more features into the dataset, which means there are 15 features of his work while there are 13 features in our dataset. Though there is no big difference between 13 features and 15 features in his work, what he did on dataset inspires us to make useful changes to our dataset (Try normalization on dataset) to make our results comprehensive. However, during this paper there are only 3 models. More models need to be considered so that the results are comprehensive. In 2017, Kaan Uyar and Ahmet İlhan[3] did the same experiment and used the same dataset as we did for projects. During their analysis, “Class distributions are interpreted as 54% absence and 46% presence of a heart disease”. The dataset we download from Kaggle has 54% 1s and 46% 0s in the target column. From their analysis, we realize 1 indicates absence of heart disease and vice versa. To make it easily understood, we switched 1s and 0s in the target column so that 1 indicates presence of heart disease to show our confusion matrix[10] in our results. After reviewing paper [4] and [5], we have learned that a neural network has the advantage of fault tolerance and it has the ability to work with inadequate knowledge as human beings. Therefore, in our project we decide to spend some time working on neural network to detect heart diseases

2.2 REFERENCES

- https://en.wikipedia.org/wiki/Cardiovascular_disease.
- http://www.heart.org/HEARTORG/Conditions/HeartAttack/WarningSignsof a
- HeartAttack/Warning-Signs-of-aHeartAttack_UCM_002039_Article.jsp#.
- WNpKgPl97IU.
- www.who.int/cardiovascular_diseases/en/.
- http://food.ndtv.com/health/world-heart-day-2015-heart-disease-in-india-is-agrowing-concern-ansari-1224160.

2.3 PROBLEM STATEMENT DEFINITION

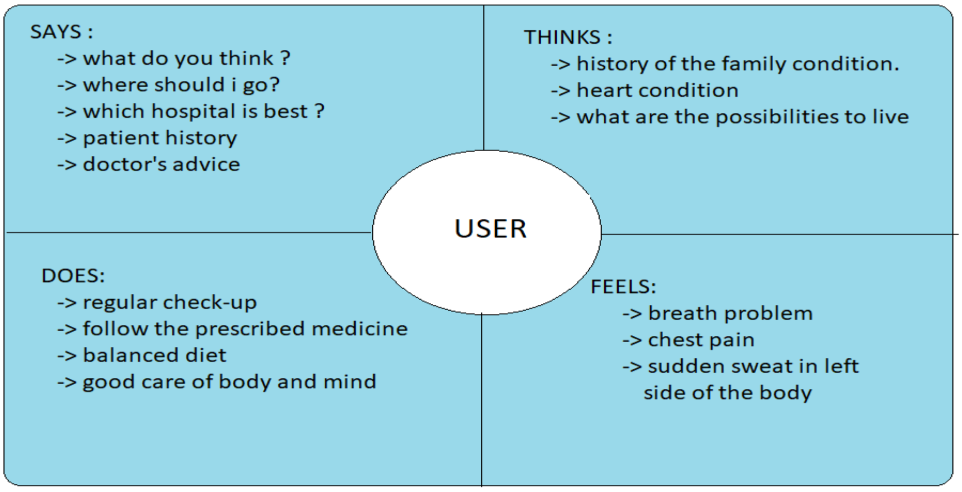
Visulaization and prediction of Heart Disease with an Interactive Dashboard. The given data consists of records of patients with their features like age, sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina and Oldpeak. Each patient has unique patient Id. Your task is to predict the target variable HeartDisease. Classify each patient as either 1: for possibility of heart disease or 0: for normal condition.

CHAPTER-3

IDEATION AND PROPOSE SOLUTION

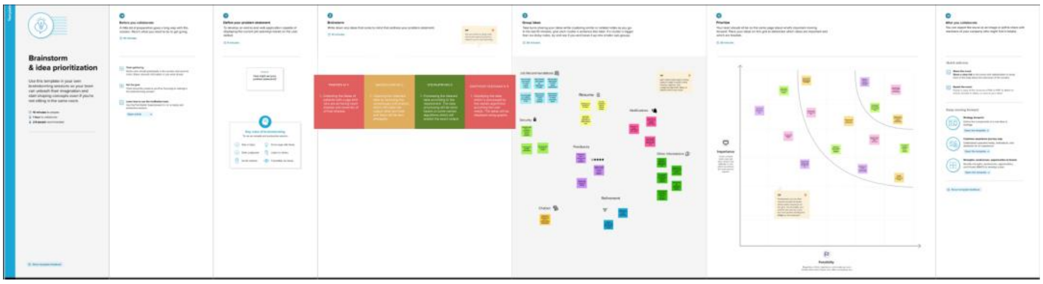
3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges. An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers.



3.2 IDEATION AND BRAIN STROMING

A group problem-solving technique that involves the spontaneous contribution of ideas fromall members of the group.The mulling over of ideas by one or more individuals in an attempt to devise or find asolution to a problem.



3.3 PROPOSED SOLUTION

Visualizing And Predicting Heart DiseasesWith An Interactive Dashboard

- The leading cause of death in the developed world is heart disease.Therefore, there needs to be work done to help prevent the risks ofhaving a heart attack or stroke.
- This database contains of 14 fields. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (nopresence) to 4.
- Use this dataset to predict which patients are most likely to suffer from aheart disease in the near future using the features given.

THE DATA DICTIONARY IS AS FOLLOWS

S.NO	FIELD NAME
1	Age
2	Sex
3	Chest pain type
4	BP
5	Cholesterol
6	FBS over 120
7	EKG results
8	Max HR
9	Exercise angina

10	ST depression
11	Slope of ST
12	Number of vessels fluro
13	Thallium
14	Heart Disease

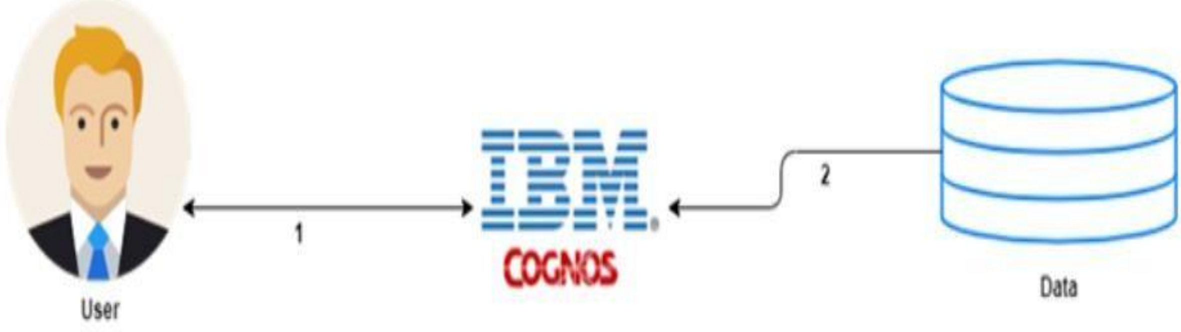
NOVELTY

- Heart diseases are the most common cause of death worldwide over the last few decades in the developed as well as underdeveloped and developing countries. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the mortality rate. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. In this project, we have developed and researched about models for heart disease prediction through the various heart attributes of the patient and we are going to create the interactive dashboard through which we can analyse the heart diseases based on age, sex, blood pressure of a person,etc.Dataset available publicly in Kaggle Website, further evaluating the results using confusion matrix and cross-validation.

FEASIBILITY OF THE IDEA

- Know fundamental concepts and can work on IBM Cognos Analytics, Gain a broad understanding of plotting different visualizations to provide a suitable solution. Able to create meaningful Visualizations and Dashboard(s). We consider a dataset which is having 14 fields by using that we are going to do explorations and building visualizations so that we can analyse the heart diseases of the patient.

BUSINESS MODELS :



SCALABILITY OF THE SOLUTION:

- We are going to do explorations and visualizations ,Exploration of bp versus chest paintype and gender ,Exploration of max heart rate duringthe chest pain,Exploration of BP by age,Exploration of cholestrol by ageand gender these are the explorations we are going to use.
- Average age for different chest pain types,Average exercise angina during chest pain,BP variation with respect to age,Effort of existing heartdisease on average of exercise angina , Average age for different types of chest pain in existing heart diseases , Maximum heart rate in existing heart disease by exercise angina these all are the visualizations.

3.4 PROBLEM SOLUTION FIT

Project Title: Visualizing and predicting heart disease with an interactive dashboard

Project Design Phase-I - Solution Fit Template

Team ID:PNT2022TMID02838

Define cc , fit into cs

CS

1. CUSTOMER SEGMENT
- Hospital
- Clinic
- People that who monitor regularly
- Scientist that who research on the dataset to find a medicine.

CC

6. CUSTOMER CONSTRAINT

the absence of data due to user confidentiality, collaborative dashboard, network connectivity, and ignorance of AI/ML technologies

AS

5.AVAILABLE SOLUTION
- Customers favour manual predictions and data visualisation.
- It is a difficult task to do because of the mathematical formula we must derive.

Define cc , fit into cs

Focus on J&P, tap into BE ,understand RC

JP

2. Jobs to be done / problems

Dataset :
Quality of the data that we are going to use is important .
If it is unreliable then the result will be not accurate while predicting.
Problem:
With the previous analysis of data, that we need to predict the heart disease with user entered current data.

RC

9. PROBLEM ROOT CAUSE
- Reason for heart disease will differs from person to person
- Few main reason are Cholesterol and usage of alcohol
- But their may be a similarity between some people
- In future root cause for heart disease may or may not finalize

BE

7.BEHAVIOUR
- Obtain a good,reliable dataset
- After a well understand difference between the field to make a comparison between them.

Focus on J&P, tap into BE ,understand RC

Identify the strong TR and EM

JP

3. TRIGGERS

inadequate method of analysing massive amounts of data and inability to determine the fundamental cause of heart disease and similarity between people with heart disease.

RC

10.YOUR SOLUTION

using ML technology to anticipate heart disease and IBM cognos to provide a user dashboard that allows for viewing and analysis of the condition

BE

ONLINE:
- Visualization
- exploration

OFFLINE:
- Collecting of dataset

Identify the strong TR and EM

EM

3. EMOTIONS: BEFORE/AFTER

BEFORE :
There is a great deal of uncertainty regarding the cause of heart disease.
AFTER:
Their may be a that to find root cause and it make better for predictions

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR NO.	FUNCTIONAL REQUIREMENT (EPIC)	SUB REQUIREMENT (STORY / SUB-TASK)
FR-1	User Registration	Registration for application through Gmail
FR-2	User Confirmation	User gets Confirmation via Email
FR-3	Visualizing data	User can view the Visualization of the available data
FR-4	Predicted Result	User can view the Predicted result based on the medical details

4.2 NON FUNCTIONAL REQUIREMENT

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

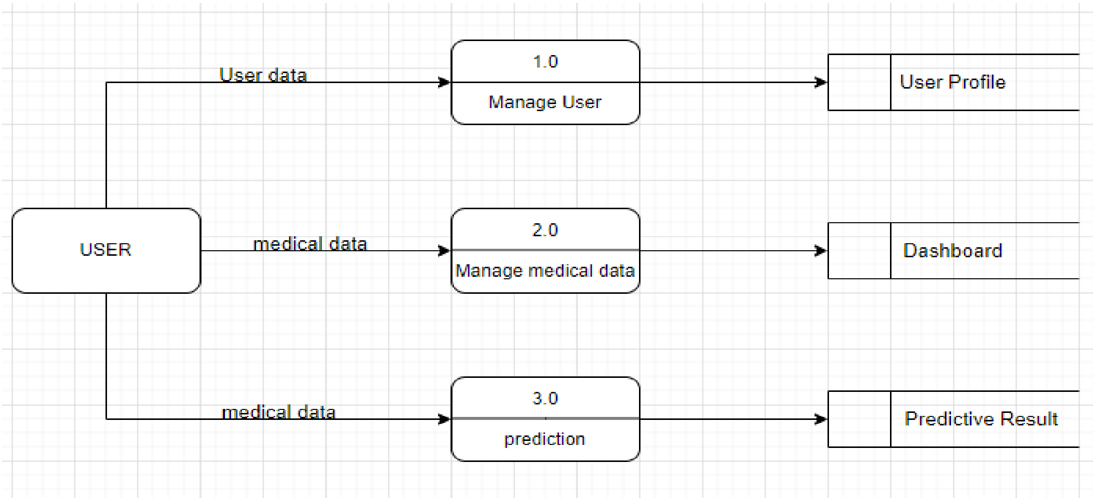
FR NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	**Usability**	The program will have a straightforward and user-friendly graphical user interface. The application's functionality will be simple for users to comprehend and utilize.
NFR-2	**Security**	Data replication
NFR-3	**Reliability**	The application must be reliable in any environment and consistent in every scenario.
NFR-4	**Performance**	Response time and data submission speed affect how well an application performs.
NFR-5	**Availability**	The application has availability of 24x7 for users
NFR-6	**Scalability**	The program can handle an increase in the number of users.

CHAPTER-5

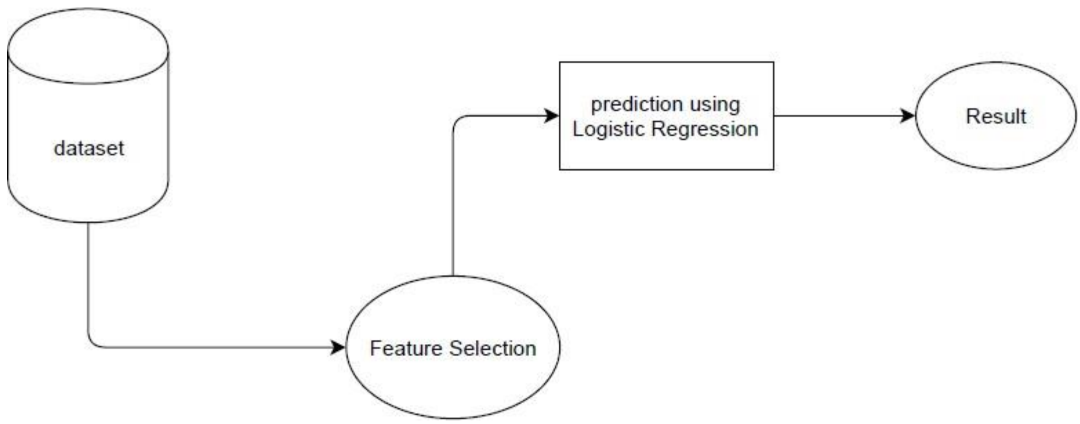
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



Heart disease is the main cause of death in the developed world.Therefore, efforts must be made to reduce the likelihood of suffering a heart attack or stroke. Using the provided attributes, this dataset can identify which patients are most likely to have a heart condition in the near future. One of the leading causes of morbidity and mortality among the global population is heart disease. One of the most crucial topics in the clinical data analysis subsection is the prediction of cardiovascular disease. In the healthcare sector, there is an enormous amount of data. The vast amount of unprocessed healthcare data is transformed via data mining into knowledge that may be used to make forecasts and educated judgments. The dataset consists of 270 individual's data. There are 14 columns in the dataset, which are described below.

IMPORT DATASET

read_csv() is used to read the CSV data with the pandas package, and then with the sklearn package we can work with some models for the prediction process .

5.3 USER STORIES

USER TYPE	FUNCTIONAL REQUIREMENT (EPIC)	USER STORY NUMBER	USER STORY / TASK
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, andpassword, and confirming my password.
		USN-2	As a user, I will receive a confirmationemail once I have registered for the application
	Login	USN-3	As a user, I can log into the application with the user-id andpassword
Customer (Web user)	Dashboard	USN-4	As a user, I can give medical data inthe application
Customer care	Helpdesk	USN-5	As a staff, I can view the customerqueries
staff			
		USN-6	As a staff, I can answer the customer queries
Administr ator	Profile	USN-7	As an admin, I can add or delete Users
		USN-8	As an admin, I can manage user details

ACCEPTANCE CRITERIA	PRIORITY	RELEASE
I can access my account / dashboard	High	Sprint-1
I can receive a confirmation email &click confirm	High	Sprint-1
I can access my account / dashboard after logging into theapplication	High	Sprint-1
I can view the visualization of trainedmedical data with user data	High	Sprint-2
I can post queries	Low	Sprint-3
I can get support	Low	Sprint-3
I can access the account / dashboard after logging into an application	High	Sprint-4
	High	Sprint-4

CHAPTER-6

PROJECT PLANNING & SCHEDULING

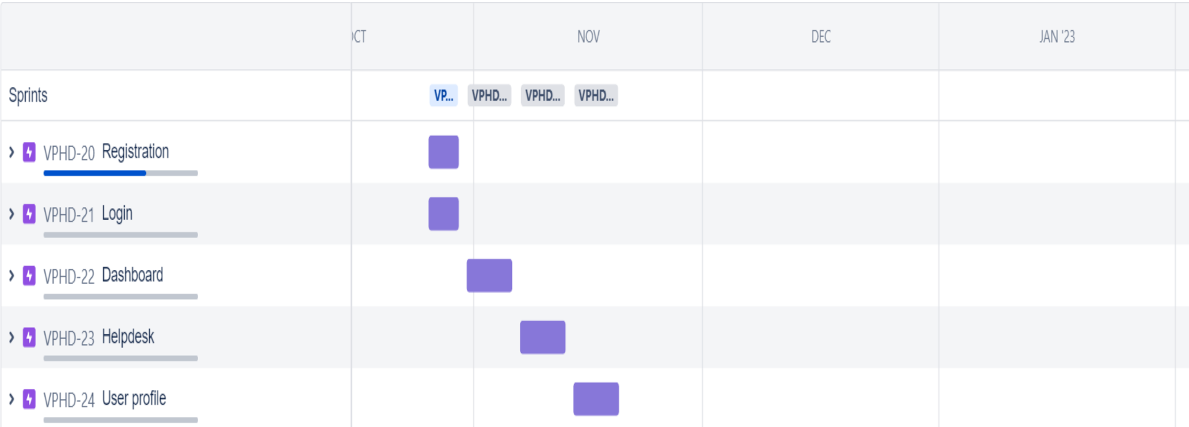
6.1 SPRINT PLANNING AND ESTIMATION

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 theapplication by entering my email, password, and confirming my password.	2	High	Varuna K Subashini K
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	2	High	Varuna K Subashini K
Sprint-1		USN-3	As a user, I can register for theapplication through Email, Google account and mobile number	2	Medium	VigneshKannaR Surya Prakash K
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email &password	2	High	Subashini K Surya Prakash K
Sprint-2	Dashboard	USN-5	As a user, I can update my profile and medical records foranalysis	5	High	Varuna K Vignesh Kanna R
Sprint-2		USN-6	As a user, I can view the accuracy of occurrence of heart disease through thereport generation	3	High	Varuna K Surya Prakash K
Sprint-3	Helpdesk	USN-7	As a user, I can post myqueries	3	Medium	Subashini K Vignesh Kanna R
Sprint-3		USN-8	As a customer care executive,he/she can view and answer the customer queries.	5	High	Subashini K Surya Prakash K
Sprint-4	User profile	USN-9	As an admin, he/she can update the health details of theusers	1	High	Varuna K
Sprint-4		USN-10	As an admin, he/she can addor delete users	2	High	Subashini K Vignesh Kanna R
Sprint-4		USN-11	As an admin, he/she canmanage the user details	5	High	Vignesh Kanna R Surya Prakash K

6.2 SPRINT DELIVERY AND SCHEDULE

SPRINT	TOTAL STORY POINTS	DURATION	SPRINT START DATE	SPRINT ENDDATE (PLANNED)	STORY POINTS COMPLETED (AS ON PLANNED END DATE)	SPRINT RELEASE DATE (ACTUAL)
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	06 Nov 2022
Sprint-2	8	6 Days	31 Oct 2022	05 Nov 2022	8	09 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	8	6 Days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

6.3 REPORTS FROM JIRA



CHAPTER-7

CODING AND SOLUTIONING

7.1 FEATURES 1

FRONT END

- CSS
- BOOTSTRAP

FRONT END CODE

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600&display=swap');


*{
  font-family: 'Poppins', sans-serif;
  margin:0; padding:0;
  box-sizing: border-box;
  outline: none; border:none;
  text-decoration: none;
}
.container{
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  padding:20px;
```

```
  padding-bottom: 60px;
}
.container .content{
  text-align: center;
}
.container .content h3{
  font-size: 30px;
  color:#333;
}
.container .content h3 span{
  background: crimson;
  color:#fff;
  border-radius: 5px;
  padding:0 15px;
}
.container .content h1{
  font-size: 50px;
  color:#333;
}
.container .content h1 span{
  color:crimson;
}
.container .content p{
  font-size: 25px;
```

```
  margin-bottom: 20px;
}
.container .content .btn{
  display: inline-block;
  padding:10px 30px;
  font-size: 20px;
  background: #333;
  color:#fff;
  margin:0 5px;
  text-transform: capitalize;
}
.container .content .btn:hover{
  background: crimson;
}
.form-container{
  min-height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  padding:20px;
  padding-bottom: 60px;
  background: #eee;
}
.form-container form{
```

```
  padding:20px;
  border-radius: 5px;
  box-shadow: 0 5px 10px rgba(0,0,0,.1);
  background: #fff;
  text-align: center;
  width: 500px;
}
.form-container form h3{
  font-size: 30px;
  text-transform: uppercase;
  margin-bottom: 10px;
  color:#333;
}
.form-container form input,
.form-container form select{
  width: 100%;
  padding:10px 15px;
  font-size: 17px;
  margin:8px 0;
  background: #eee;
  border-radius: 5px;
}
.form-container form select option{
  background: #fff;
```

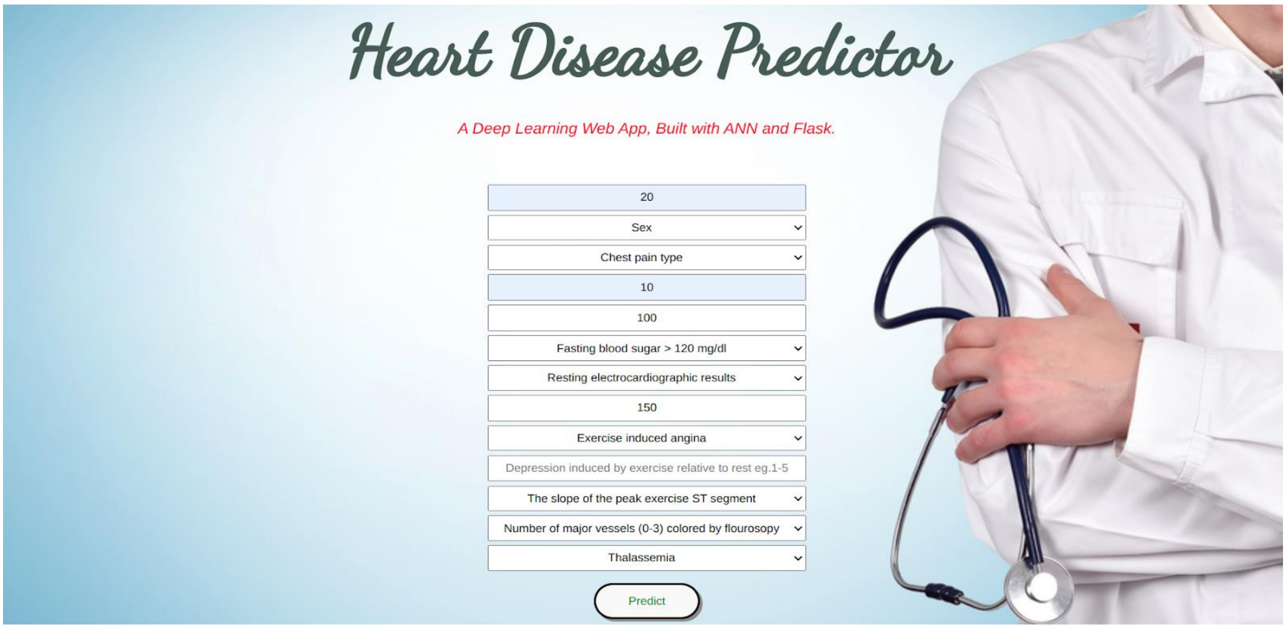
```
}
.form-container form .form-btn{
  background: #fbd0d9;
  color:crimson;
  text-transform: capitalize;
  font-size: 20px;
  cursor: pointer;
}
.form-container form .form-btn:hover{
  background: crimson;
  color:#fff;
}
.form-container form p{
  margin-top: 10px;
  font-size: 20px;
  color:#333;
}
.form-container form p a{
  color:crimson;
}
.form-container form .error-msg{
  margin:10px 0;
  display: block;
  background: crimson;
```

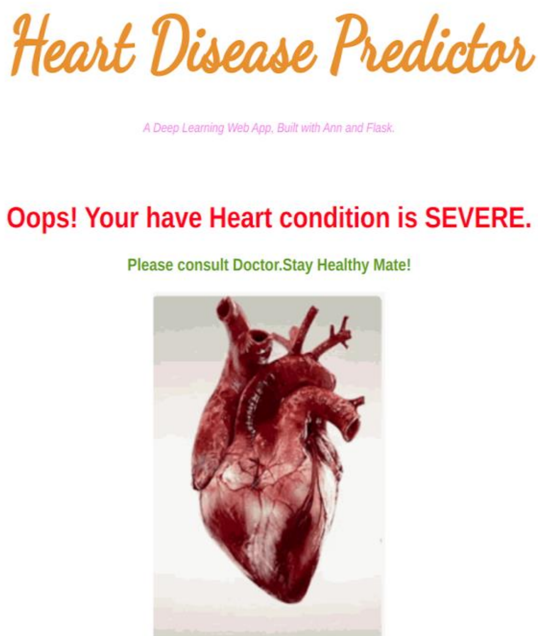
```
    color:#fff;
    border-radius: 5px;
    font-size: 20px;
    padding:10px;
  }
  function myfunction(){
     var x =document.getElementById("pass");

     if(x.type === "password"){
        x.type = "text";
     }
     else{
        x.type = "password";
     }
  }
  function validate(){
     var password = document.getElementById("pass");
     var length = document.getElementById("length");
     if(password.value.length >= 8){
        alert("Login Successful");
        window.location.replace("heart .html");
        return false;
     }
     else{
```

```
            alert("Login Failed");
        }
    }
    function page(){
        window.location.replace("Landingpage.html")
    }
      window.watsonAssistantChatOptions = {
        integrationID: "3f5d7446-04cb-4796-8496-1351f8d8acfd", // The ID of this integration.
        region: "au-syd", // The region your integration is hosted in.
        serviceInstanceID: "58af1a9a-a26e-4a48-ae6f-0881b0be4c0d", // The ID of your service instance.
        onLoad: function(instance) { instance.render(); }
      };
      setTimeout(function(){
        const t=document.createElement('script');
        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
      });
    function myfunction(){
        var x =document.getElementById("pass");
        if(x.type === "password"){
            x.type = "text";
        }
```

```
    else{
        x.type = "password";
    }
}
function validate(){
    var password = document.getElementById("pass");
    var length = document.getElementById("length");
    if(password.value.length >= 8){
        alert("Login Successful");
        window.location.replace("heart .html");
        return false;
    }
    else{
        alert("Login Failed");
    }
}

function page(){
    window.location.replace("Landingpage.html")
}
```

SCREENSHOTS





7.2 FEATURE 2

BACK END CODE:

```
#!/usr/bin/env python

# coding: utf-8

# In[2]:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

# In[4]:

df=pd.read_csv('Downloads/Heart_Disease_Prediction.csv')

# In[5]:

df.head()

# In[6]:

df.isnull().sum()

# In[7]:

print(df.info())

# In[9]:

plt.figure(figsize=(20,10))

sns.heatmap(df.corr(), annot=True, cmap='terrain')

# In[10]:
```

```
sns.pairplot(data=df)

# In[11]:

df.hist(figsize=(10,12), layout=(5,4));

# In[13]:

df.plot(kind='box', subplots=True, layout=(6,3), figsize=(10,10))

plt.show()

# In[19]:

sns.catplot(data=df, x='Sex', y='Age', hue='Heart Disease', palette='tab10')

# In[20]:

sns.barplot(data=df, x='Sex', y='Cholesterol', hue='Heart Disease',
palette='spring')

# In[21]:

df['Sex'].value_counts()

# In[25]:

gen.plot(kind='bar', stacked='True', color=['green','blue'],grid=False)

# In[ ]:

# In[ ]:

# In[22]:

df['Chest pain type'].value_counts()

# In[23]:

sns.countplot(x='Chest pain type', hue='Heart Disease' , data=df,
palette='rocket')

# In[24]:

gen = pd.crosstab(df['Sex'], df['Heart Disease'])
```

```
print(gen)
    #!/usr/bin/env python
# coding: utf-8
# In[1]:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
# In[2]:
df=pd.read_csv('Downloads/Heart_Disease_Prediction.csv')
# In[3]:
df.head()
# In[4]:
df.isnull().sum()
# In[5]:
print(df.info())
# In[6]:
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(), annot=True, cmap='terrain')
# In[7]:
sns.pairplot(data=df)
# In[8]:
```

```
df.hist(figsize=(10,12), layout=(5,4));

# In[9]:

df.plot(kind='box', subplots=True, layout=(6,3), figsize=(10,10))

plt.show()

# In[10]:

sns.catplot(data=df, x='Sex', y='Age', hue='Heart Disease', palette='tab10')

# In[11]:

sns.barplot(data=df, x='Sex', y='Cholesterol', hue='Heart Disease',
palette='spring')

# In[12]:

df['Sex'].value_counts()

# In[13]:

df['Chest pain type'].value_counts()

# In[14]:

sns.countplot(x='Chest pain type', hue='Heart Disease' , data=df,
palette='rocket')

# In[15]:

gen = pd.crosstab(df['Sex'], df['Heart Disease'])

print(gen)

# In[16]:

gen.plot(kind='bar', stacked='True', color=['green','blue'],grid=False)

# In[17]:

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler
```

```
StandardScaler = StandardScaler()

columns_to_scale=['Age', 'EKG results', 'Cholesterol', 'Thallium', 'Number of vessels fluro']

df[columns_to_scale] = StandardScaler.fit_transform(df[columns_to_scale])

# In[18]:

df.head()

# In[19]:

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

StandardScaler = StandardScaler()

columns_to_scale=['Age', 'EKG results', 'Cholesterol', 'Thallium', 'Number of vessels fluro']

df[columns_to_scale] = StandardScaler.fit_transform(df[columns_to_scale])

# In[20]:

df.head()

# In[21]:

x=df.drop(['Heart Disease'], axis=1)

y=df['Heart Disease']

# In[22]:

x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.3, random_state=40)

# In[23]:

print('x_train-', x_train.size)

print('x_test-', x_test.size)

print('y_train-', y_train.size)
```

```
print('x_test-', x_test.size)

# In[24]:

from sklearn.linear_model import LogisticRegression

lr=LogisticRegression()

model1=lr.fit(x_train,y_train)

prediction1=model 1.predict(x_test)

# In[25]:

from sklearn.metrics import confusion_matrix

cm=confusion_matrix(y_test,prediction1)

cm

# In[26]:

sns.heatmap(cm, annot=True,cmap='BuPu')

# In[27]:

TP=cm[0][0]

TN=cm[1][1]

FN=cm[1][0]

FP=cm[0][1]

print('Testing Accuracy:', (TP+TN+FN)/(TP+TN+FN+FP))

# In[28]:

from sklearn.metrics import accuracy_score

accuracy_score(y_test,prediction1)

l=accuracy_score(y_test,prediction1)

# In[29]:

from sklearn.metrics import classification_report
```

```
print(classification_report(y_test, prediction1))

# In[30]:

import pandas as pd

from sklearn import neighbors,metrics

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

import numpy as np

import pickle

from sklearn.ensemble import RandomForestClassifier

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

import seaborn as sns

import matplotlib.pyplot as plt

# In[31]:

from sklearn.metrics import accuracy_score

# In[32]:

dataset = pd.read_csv("Downloads/Heart_Disease_Prediction.csv")

# In[33]:

KX = dataset[['Age','Sex','Chest pain type','BP','Cholesterol','FBS over 120','EKG results','Max HR','Exercise angina','ST depression','Slope of ST','Number of vessels fluro','Thallium']].values

# In[34]:

KY = dataset[['Heart Disease']].values

# In[35]:

KX
```

```
# In[36]:

KY = KY.flatten()

print(KY)

# In[37]:

KX_train , KX_test , KY_train , KY_test =
train_test_split(KX,KY,test_size=0.2,random_state=4)

# In[38]:

knn = KNeighborsClassifier(n_neighbors = 20)

knn.fit(KX_train, KY_train)

print(knn.score(KX_test, KY_test))

# In[39]:

pickle.dump(knn,open('heart_knn_model.sav','wb'))

# In[40]:

predict_knn = knn.predict(KX_test)

accuracy_knn  = metrics.accuracy_score(KY_test,predict_knn)

# In[41]:

predict_knn

# In[42]:

accuracy_knn

# In[43]:

k=accuracy_knn

# In[45]:

import csv

import pandas as pd
```

```
import numpy as np

from sklearn.naive_bayes import GaussianNB

from sklearn.model_selection import train_test_split

from sklearn import metrics

from sklearn.metrics import confusion_matrix, f1_score, roc_curve, auc

import matplotlib.pyplot as plt

from itertools import cycle

from scipy import interp

# In[46]:

df = pd.read_csv('Downloads/Heart_Disease_Prediction.csv', header = None)

# In[47]:

training_x=df.iloc[1:df.shape[0],0:13]

# In[48]:

training_y=df.iloc[1:df.shape[0],13:14]

# In[49]:

nx=np.array(training_x)

ny=np.array(training_y)

# In[52]:

for z in range(5):

    print("\nTest Train Split no. ",z+1,"\n")

    nx_train,nx_test,ny_train,ny_test =
train_test_split(nx,ny,test_size=0.25,random_state=None)

    # Gaussian function of sklearn

    gnb = GaussianNB()
```

```
    gnb.fit(nx_train, ny_train.ravel())

    ny_pred = gnb.predict(nx_test)

# In[61]:

print("\n Naive Bayes model accuracy(in %):", metrics.accuracy_score(ny_test, ny_pred))

# In[62]:

n=metrics.accuracy_score(ny_test, ny_pred)

# In[64]:

import pandas as pd

from sklearn import neighbors,metrics

from sklearn.model_selection import train_test_split

from sklearn.neighbors import KNeighborsClassifier

import numpy as np

import pickle

from sklearn.ensemble import RandomForestClassifier

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

import seaborn as sns

import matplotlib.pyplot as plt

# In[65]:

from sklearn.metrics import accuracy_score

# In[67]:

dataset = pd.read_csv("Downloads/Heart_Disease_Prediction.csv")

# In[69]:
```

```
DX = dataset[['Age','Sex','Chest pain type','BP','Cholesterol','FBS over 120','EKG results','Max HR','Exercise angina','ST depression','Slope of ST','Number of vessels fluro','Thallium']].values

# In[70]:

dy = dataset[['Heart Disease']].valueE

# In[71]:

DX

# In[72]:

dy = dy.flatten()

print(dy)

# In[73]:

DX_train , DX_test , dy_train , dy_test = train_test_split(DX,dy,test_size=0.2,random_state=4)

# In[74]:

from sklearn.tree import DecisionTreeClassifier

max_accuracy = 0

# In[75]:

for x in range(200):

    dt = DecisionTreeClassifier(random_state=x)

    dt.fit(DX_train,dy_train)

    dy_pred_dt = dt.predict(DX_test)

    current_accuracy = round(accuracy_score(dy_pred_dt,dy_test)*100,2)

    if(current_accuracy>max_accuracy):

        max_accuracy = current_accuracy
```

```
        best_x = x
    # In[85]:
    dt = DecisionTreeClassifier(random_state=best_x)
    dt.fit(DX_train,dy_train)
    dy_pred_dt = dt.predict(DX_test)
    # In[88]:
    score_dt = (accuracy_score(dy_pred_dt,dy_test))
    # In[89]:
    print("The accuracy score achieved using Decision Tree is: "+str(score_dt))
    # In[90]:
    d=(accuracy_score(dy_pred_dt,dy_test))
    # In[91]:
    print('Logistic Regression :',l)
    print('KNN :',k)
    print('Naive Bayes :',n)
    print('Decision Tree :' ,d)
    # In[93]:
    print('Logistic Regression :',l*100,'%')
    print('KNN :',k*100,'%')
    print('Naive Bayes :',n*100,'%')
    print('Decision Tree :' ,d*100,'%')
    # In[ ]:
    import js2py
    import os
```

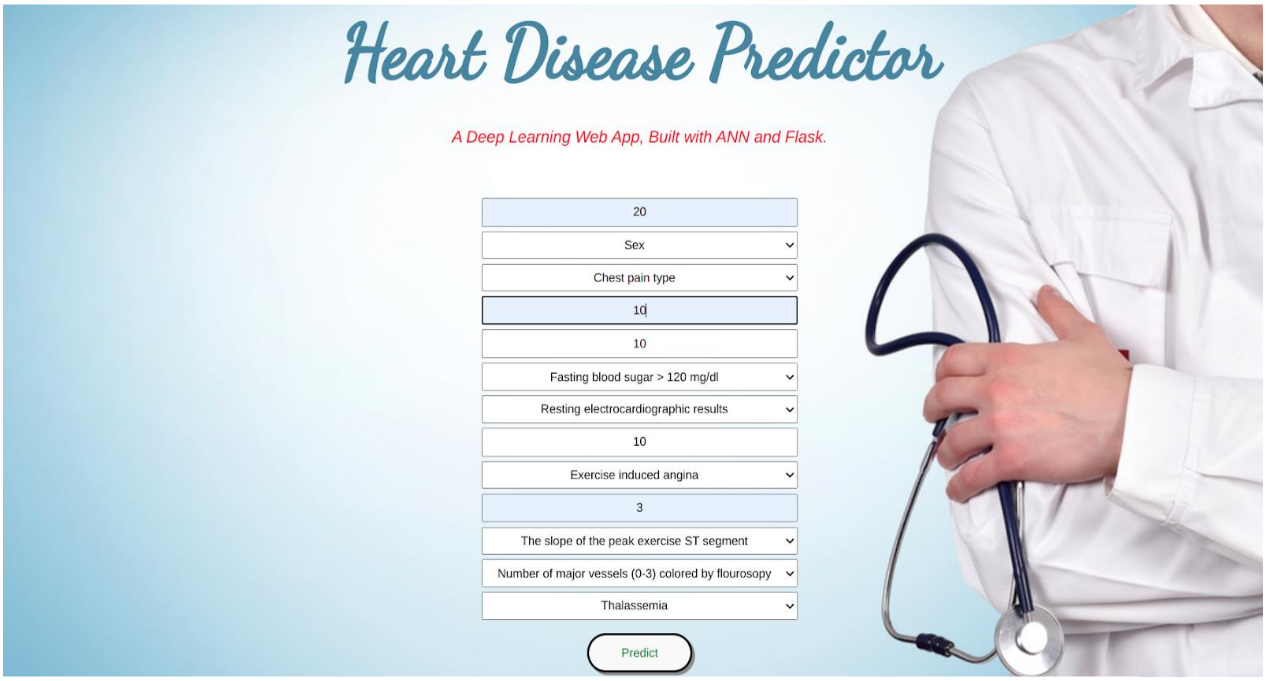
```
from flask import Flask,request,jsonify, json, Response, make_response,
render_template

from flask_pymongo import PyMongo

from flask_bcrypt import Bcrypt

from flask_cors import CORS

import db

js2py.run_file(bot.js)

app = Flask(__name__)

bcrypt = Bcrypt(app)

CORS(app)

@app.route('/')

@app.route("/test")

def test():

    return "Connected to the database!"

class UserAuthUtil:

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

    def hello_world():

        return "Working"

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

    def login_user():

        try:

            if request.method == 'POST':

                form_data = request.get_json()

                email = form_data['email']
```

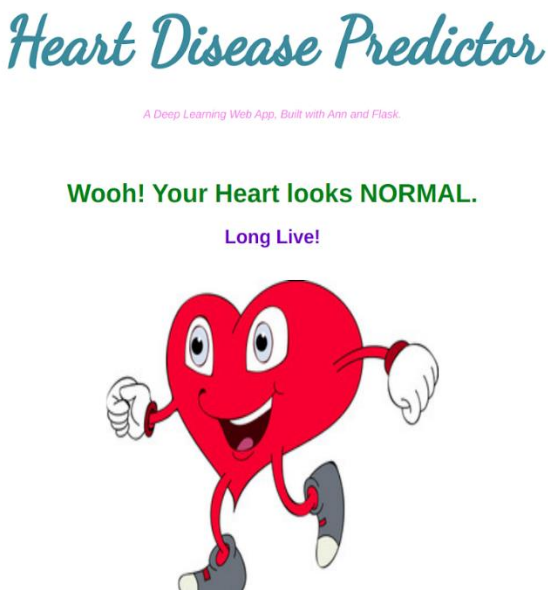
```
                password = form_data['password']

                if(email != '' and password != ''):

                    data = list(db.users.find({'email': email}))

                    if(len(data) == 0):

                        return Response(status=404, response=json.dumps({'message': 'user does not exist'}), mimetype='application/json')

                    else:

                        data = data[0]

                        if(bcrypt.check_password_hash(data['password'], password)):

                            #token =jwt.encode({'email': email}, app.config['SECRET_KEY'])

                            return make_response(jsonify({'message':'User logged in successfully'}), 201)

                        else:

                            return Response(status=402, response=json.dumps({'message': 'Invalid password'}), mimetype='application/json')

                else:

                    return Response(status=400, response=json.dumps({'message': 'Bad request'}), mimetype='application/json')

            else:

                return Response(status=401, response=json.dumps({'message': 'invalid request type'}), mimetype='application/json')

        except Exception as Ex:

            print('\n\n\n*********************************')

            print(Ex)

            print('*********************************\n\n\n')

            return Response(response=json.dumps({'message': "Internal Server
```

```
error"}), status=500, mimetype="application/json")


        @app.route("/register", methods=['POST'])
        def register_user():
            try:
                if request.method == "POST":
                    user_details = request.get_json()
                    full_name = user_details["fullName"]
                    email = user_details["email"]
                    password = user_details["password"]
                    password_hash =
bcrypt.generate_password_hash(password).decode('utf-8')
                    if (full_name != '' and email != '' and password_hash != ''):
                        db.users.insert_one({'fullName':full_name,'email':email,'password':
password_hash})
                        return Response(response=json.dumps({'message': 'User created
successfully'}), status=200, mimetype="application/json")
                    else:
                        return Response(status=400, response=json.dumps({'message':
'Please enter your details'}), mimetype='application/json')
                else:
                    return Response(status=400, response=json.dumps({'message': 'Bad
request'}), mimetype='application/json')
            except Exception as Ex:
                print('\n\n\n*********************************')
                print(Ex)
```

```
            print('********************************\n\n\n')

            return Response(response=json.dumps({'message': "Internal Server Error"}), status=500, mimetype="application/json")

    if __name__ == '__main__':

        app.run(port=8000)
```

SCREENSHOTS





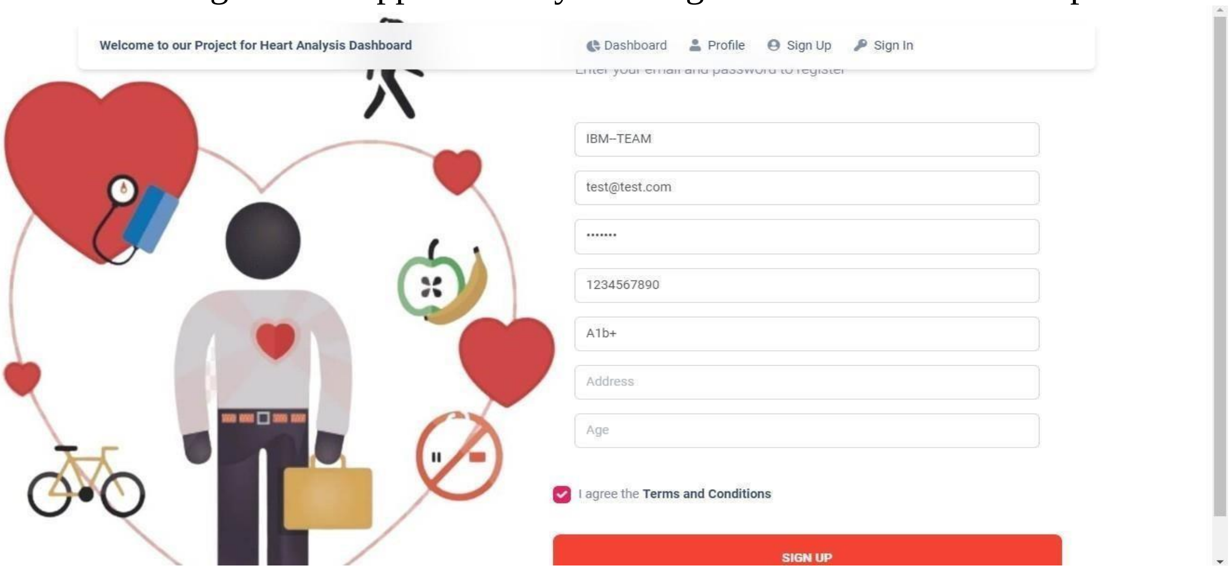
CHAPTER-8
TESTING

8.1 TEST CASES

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

LOGIN

Can't log into the application by entering email/PhoneNumber & password



CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS

Performance Analysis is the one used for predicting the algorithm based on various metrics such as accuracy, precision, recall/F1-Score, etc. Performance Analysis aims at comparing the accuracy and performance of machine learning models. The metrics is evaluated with four measures.

TP [True Positive]

If the positive input (from dataset) is given to the classifier, it gives positive output (predicted value). It predicts the total true positive cases identified correctly. True positive values will be 1(True) for heart failured patients. The predicted value will be value 1 if the true positive value is 1.

TN [True Negative]

If the negative input (from dataset) is given to the classifier, it gives negative output (predicted value). It predicts the total true negative cases identified correctly. The true negative values will be 0(False) for heart failured patients. The predicted value will be 1(True) if the true negative value is 0(false).

FP [False Positive]

If the negative input (from dataset) is given to the classifier, it gives positive output (predicted value). It predicts the total false positive cases identified incorrectly. The false positive values will be 0 for heart failured patients. The predicted value will be 1 if the false positive value is 0.

FN [False Negative]

If the positive input (from dataset) is given to the classifier, it gives negative output (predicted value). It predicts the total false negative cases identified incorrectly. The false negative values will be 0 for heart failured patients. The predicted value will be 0 if the values predicted are false.

The important metrics for performance analysis are accuracy, recall (F1-Score) and precision. These above measures are used to define the metric.

Accuracy (A)

Accuracy is a performance metric that has the correct predictions for the test data. It gives the percentage of correct predictions for testing the data. In machine learning, accuracy is calculated using the formula as shown in equation (1),

$$\mathbf{(TP+TN)/(TP+TN+FP+FN)}$$

Recall (R) or F1-Score

Recall metric is used to predict the number of correct samples (all samples identified as positive). It is the fraction of values (results) returned to the total number of values that can be returned. It is the ratio between true positives and all the actual positives. It measures the correctly identified positive samples out of all the actual positive samples. The recall is calculated using the formula as shown in equation (2),

$$\mathbf{TP/(TP+FN)}$$

Precision (P)

Precision which is also called as “positive predictive value”. It gives the percentage of true positives. It is the ratio between true positives and all the predicted positives. It measures the correctly identified positive samples out of all positively predicted samples. The precision is calculated using the formula as shown in equation (3),

$$\mathbf{TP/(TP+FP)}$$

CHAPTER-10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Increased accuracy for effective heart disease diagnosis.
- Handles roughest(enormous) amount of data using random forest algorithm and feature selection.
- Reduce the time complexity of doctors.
- Cost effective for patients.

DISADVANTAGES

- Prediction of cardiovascular disease results is not accurate.
- Data mining techniques do not help to provide effective decision making.
- Cannot handle enormous datasets for patient records.

CHAPTER-11

CONCLUSION

we proposed a method for heart disease prediction using machine learning techniques, these results showed a great accuracy standard for producing a better estimation result. By introducing new proposed Random forest classification, we find the problem of prediction rate without equipment and propose an approach to estimate the heart rate and condition. Sample results of heartrate are to be taken at different stages of the same subjects, we find the information from the above input via ML Techniques. Firstly, we introduced a support vector classifier based on datasets.

CHAPTER-12

FUTURE SCOPE

The future scope of this system aims at giving more sophisticated prediction models, risk calculation tools and feature extraction tools for other clinical risks. Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modeling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions

CHAPTER 13

APPENDIX

13.1 SOURCE CODE

https://github.com/IBM-EPBL/IBM-Project-17756-1659675857/tree/main/Final%20Deliverables/Final%20Code

13.2 GITHUB AND PROJECT DEMO LINK

GITHUB

https://github.com/IBM-EPBL/IBM-Project-17756-1659675857

PROJECT DEMO LINK

https://drive.google.com/file/d/1CU3q7weEDA13MVK9kc47TcYLmVwO8WOs/view?usp=share_link