Visualizing and Predicting Heart Diseases with an Interactive Dashboard

NALAIYA THIRAN PROJECT REPORT
2022

Submitted by

Boomika R B

Aathish R

Abinaya C

Deepthi N

Team ID: PNT2022TMID04303

Content

CHAPTER-1

INTRODUCTION

1.1 Project Overview

The leading cause of death in the developed world is heart disease. Therefore, there needs to be work done to help prevent the risks of having a heart attack or stroke. This project aims to create an interactive Dashboard using IBM Cognos Tool and dataset to predict which patients are most likely to suffer from a heart disease in the near future using the features given.

1.2 Purpose

Heart disease (HD) is a major cause of mortality in modern society. Medical diagnosis is an extremely important but complicated task that should be performed accurately and efficiently. Cardiovascular disease is difficult to detect due to several risk factors, including high blood pressure, cholesterol, and an abnormal pulse rate. Based on the analytics we can analyze which patients are most likely to suffer from heart disease in the near future and based on the patient details we will make decisions to cure them.
Various details are fed in the application and the heart disease associated with those details. Users can share their heart related issues with this application. It then processes user specific details to check for various illness that could be associated with it. After getting the result from the system,patient may contact the Doctors. The system can be use in case of emergency.

CHAPTER-2

LITERATURE SURVEY

2.1 Existing Problem

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in an attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithms.

2.2 References

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9.

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using

lazy associative classification. In 2013 International MutliConference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[10] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiologyy, 18(2), 361-7.

[11] Chen A H, Huang S Y, Hong P S, Cheng C H & Lin E J (2011, September). HDPS: Heart disease prediction system. In 2011 Computing in Cardiology (pp. 557- 60). IEEE.

[12] Parthiban, Latha and R Subramanian. "Intelligent heart disease prediction system using CANFIS and genetic algorithm." International Journal of Biological, Biomedical and Medical Sciences 3.3 (2008).

2.1 Problem Statement Definition

In India in 2016, CVDs (Cardiovascular Diseases) contributed to 28·1% of total deaths and 14·1% of total disability-adjusted life years (DALYs). Most persons with coronary heart disease who pass away are 65 years of age or older. Although both sexes can get heart attacks in old age,women have a higher mortality rate (within a few weeks). Risk for heart disease can be increasedby a number of medical issues, lifestyle, age, and family history. When a person is affected by heart disease, it causes side effects. Chest pain, chest tightness, chest pressure and chest discomfort Breathing difficulties, Neck, jaw, throat, upper abdomen, or back pain. Heart disease -and the conditions that lead to it - can happen at any age. High rates of obesity and high blood pressure among younger people (ages 35–64) are putting them at risk for heart disease earlier in life. CAD happens when coronary arteries struggle to supply the heart with enough blood, oxygen and nutrients. Cholesterol deposits, or plaques, are almost always to blame. Thesebuildups narrow your arteries, decreasing blood flow to your heart. This can cause chest pain, shortness of breath or even a heart attack.

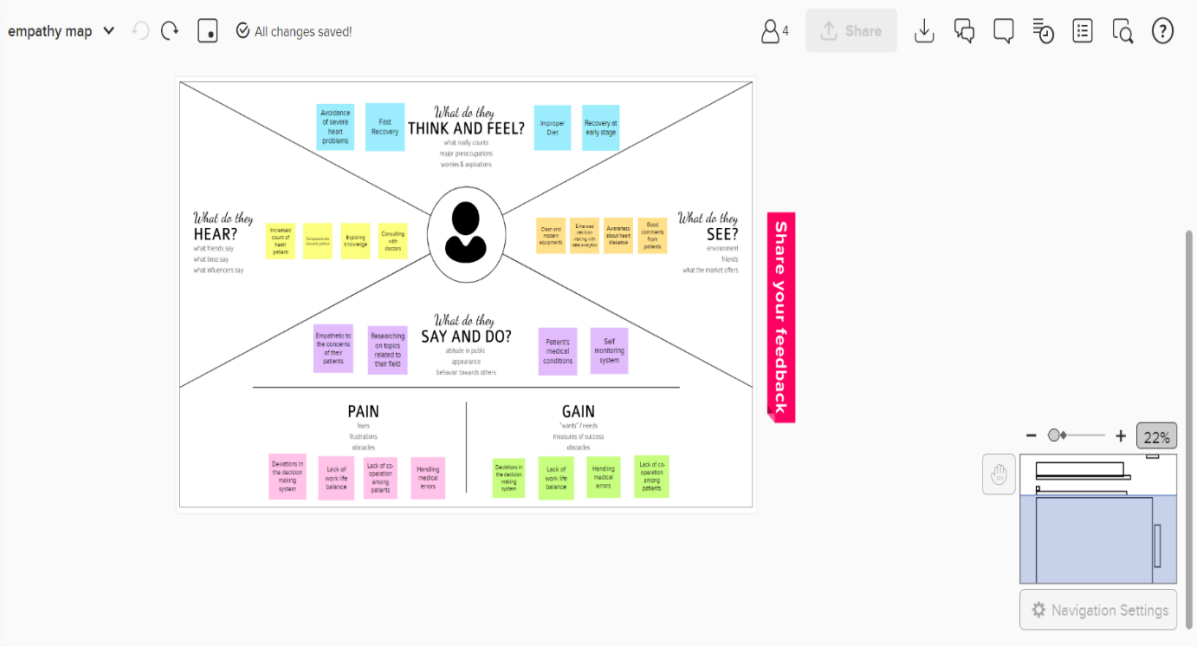
Therefore in order Predict if the patient suffers from heart disease- The health professional entersthe input values from the patient's health report. The data is fed into the project model which predicts the probability of having heart disease.

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

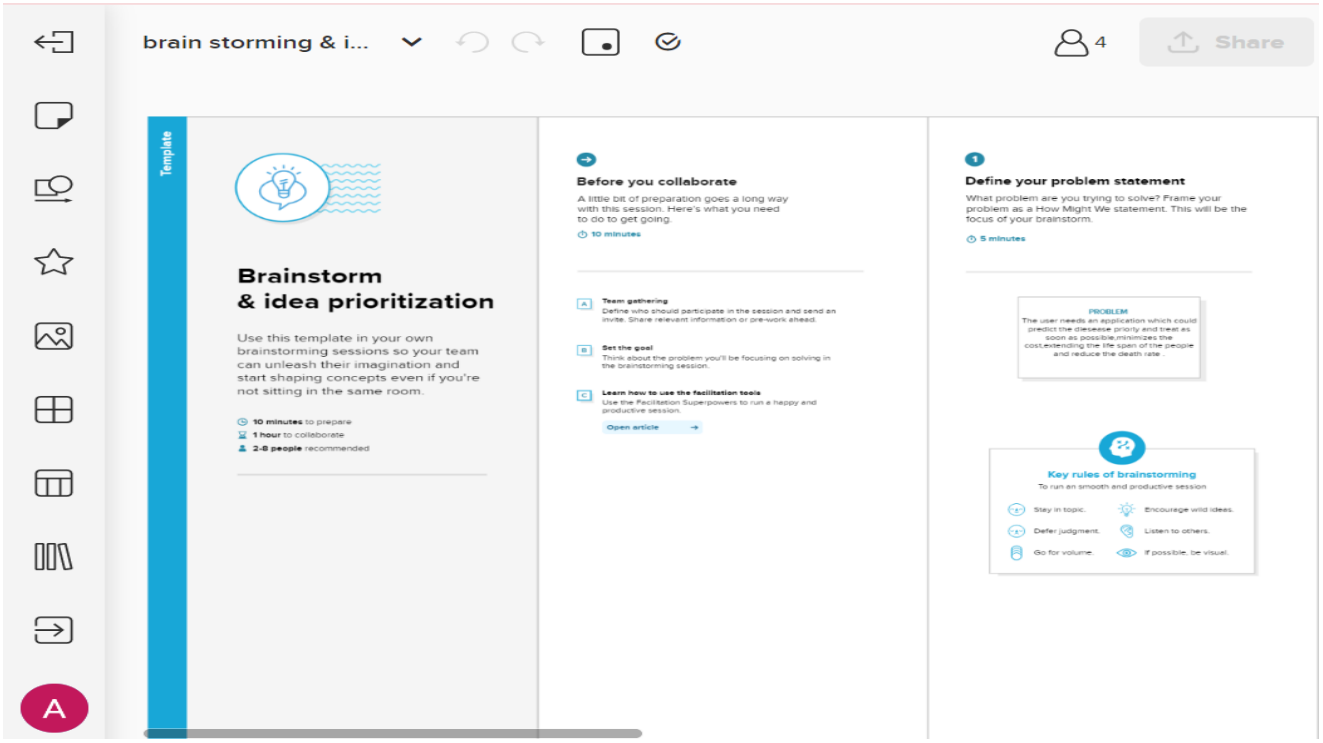
An empathy map canvas is a more in-depth version of the original empathy map, which helpsidentify and describe the user's needs and pain points. And this is valuable information for improving the user experience.



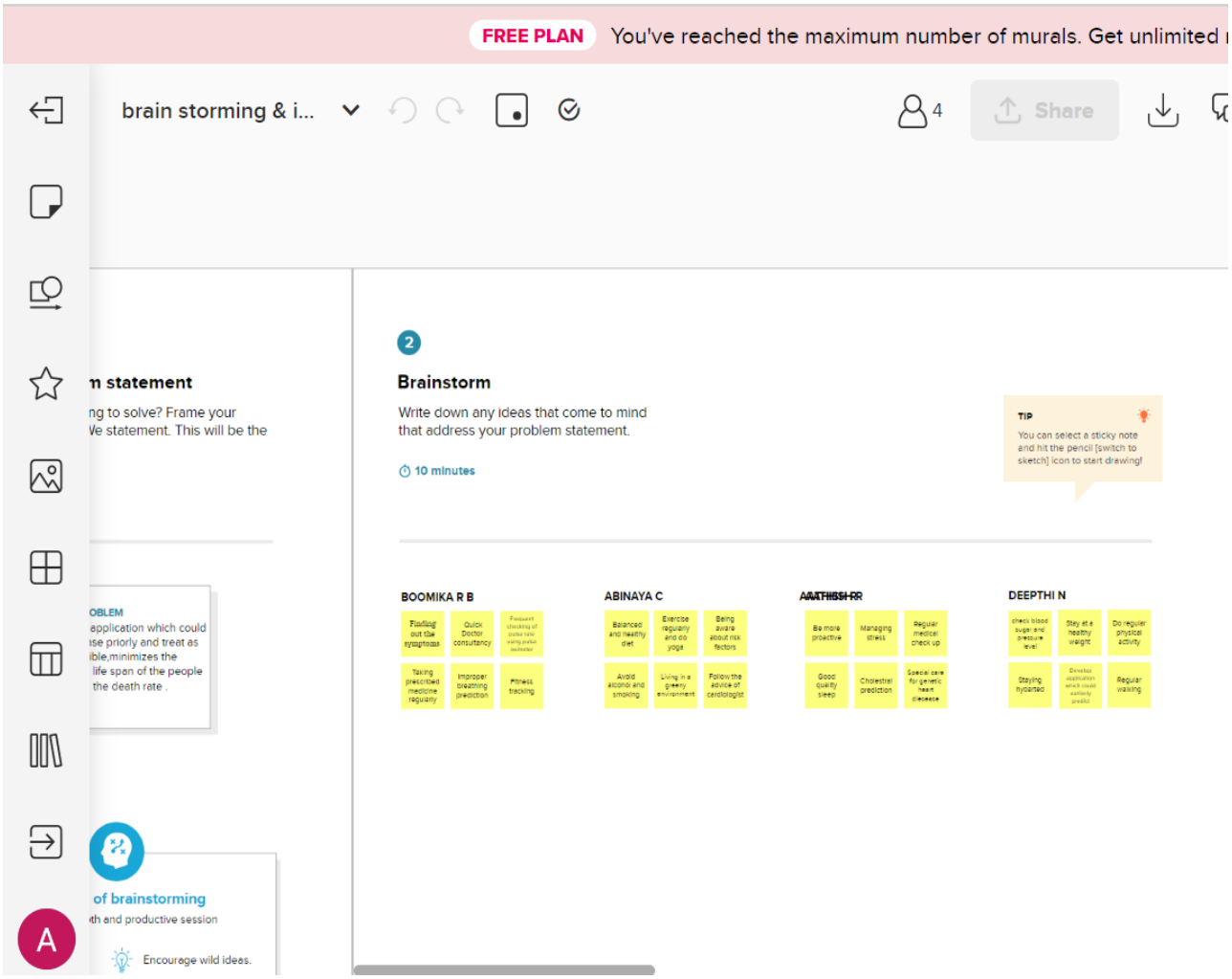
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volumeover value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich number of creative solutions.

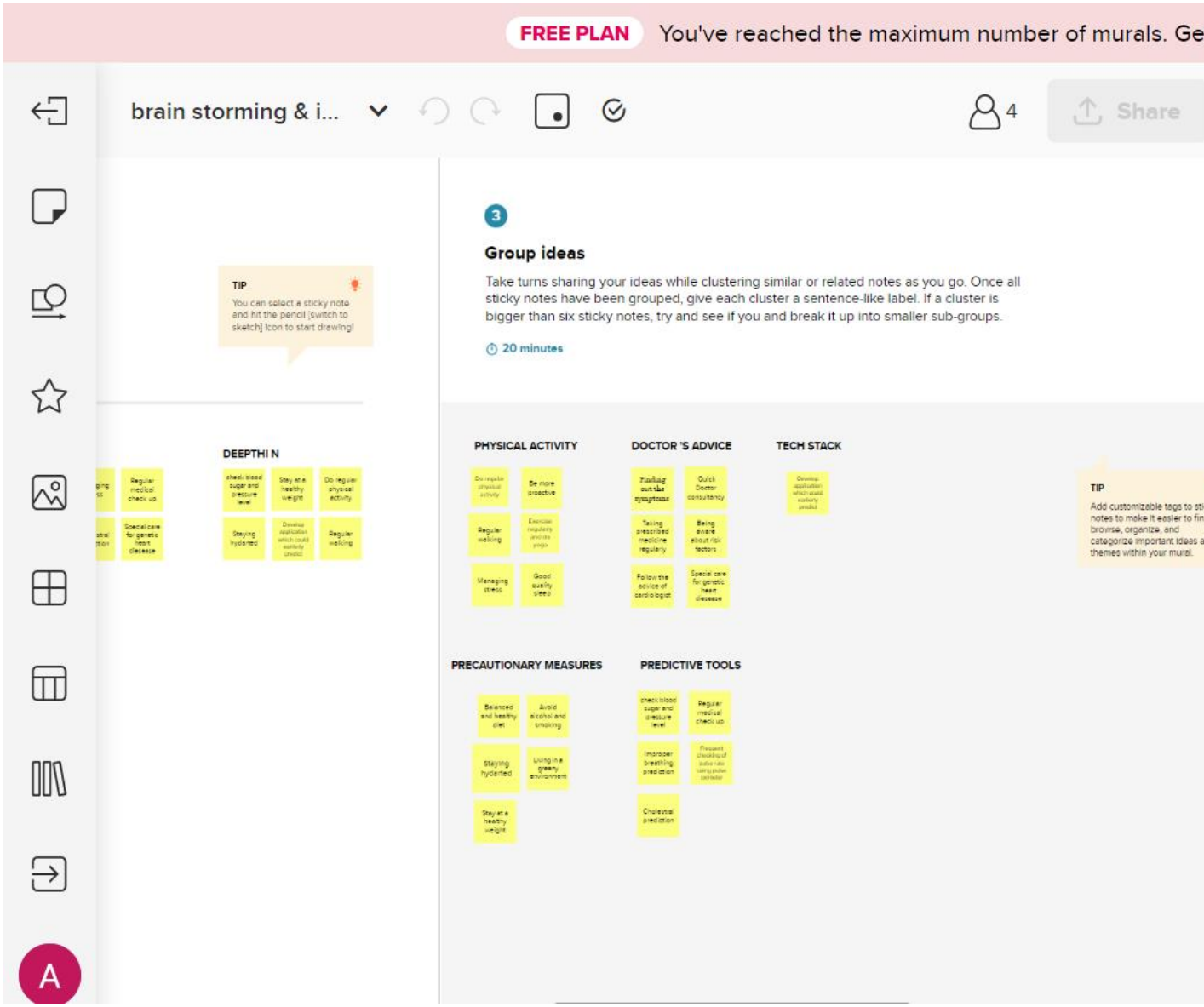
Step 1: Defining Problem Statement



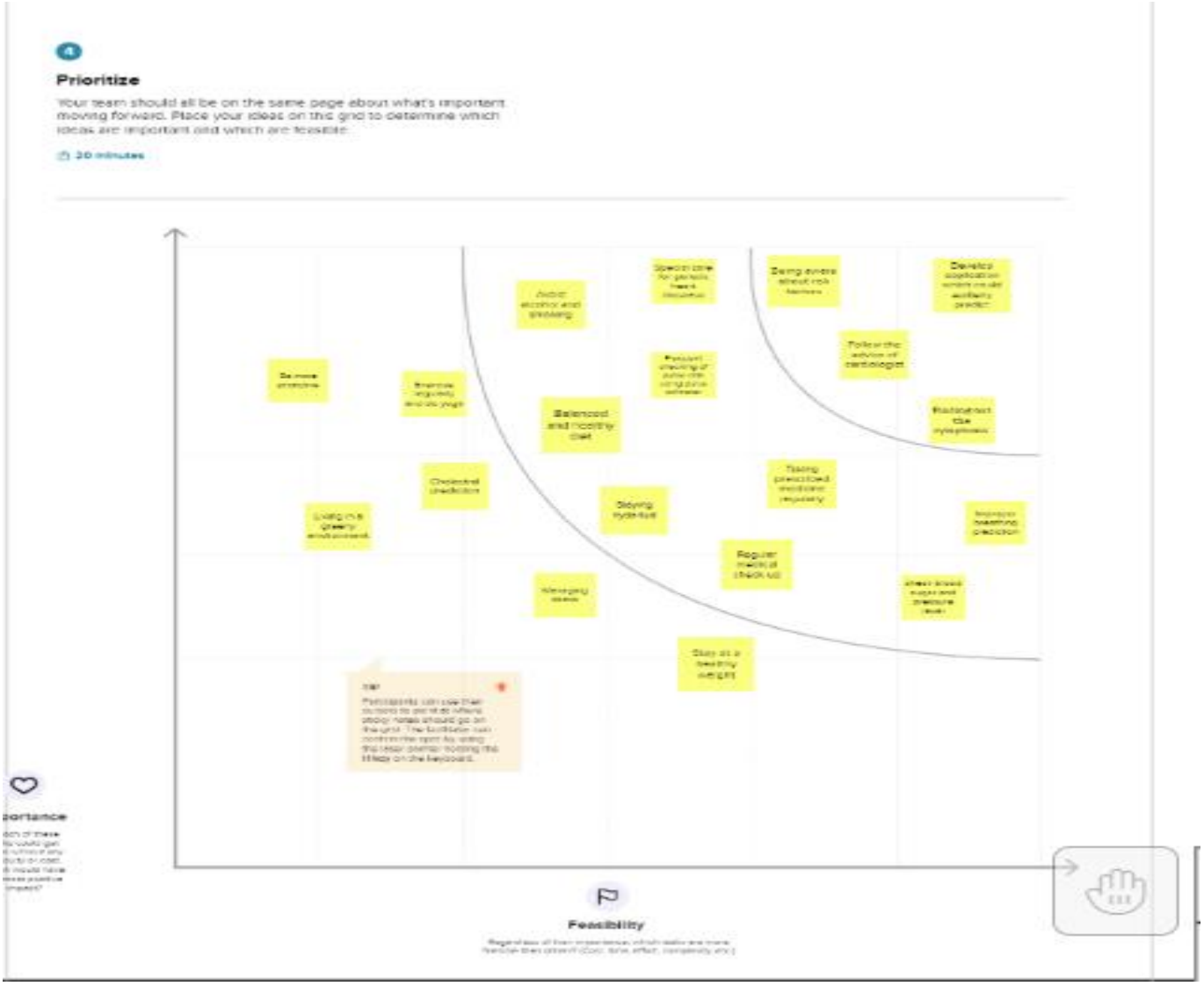
Step 2: Brainstorming ideas



Step 3: Grouping ideas

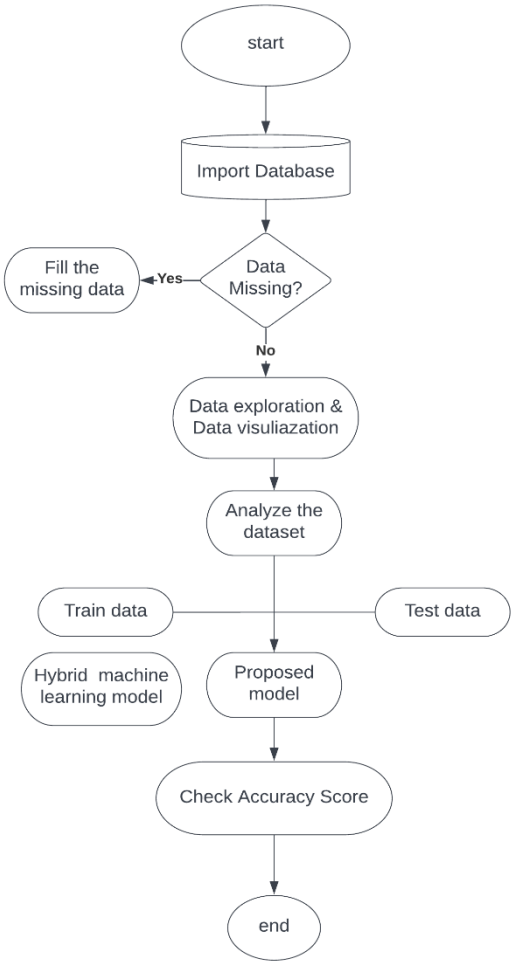


Step 5: Prioritizing ideas



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The leading cause of death in the developed world is heart disease. Therefore, there needs to be work done to help prevent the risks of having a heart attack or stroke.
2.	Idea / Solution description	The suggested solution is interactive dashboard for visualizing and forecasting cardiac ailments, where the user may see both analysis of their medical report and anticipated outcome. The dashboard will be made with IBM Cognos . The dataset is pre-processed to check missing values, noisy data and to clean the data. The dataset is explored and visualised and then machine learning model is used for prediction of heart disease.
3.	Novelty / Uniqueness	Machine learning algorithms are used for fast prediction of heart disease. The uniqueness of our proposal is to convey the availabilities to the customer with maximum accuracy.

4.	Social Impact / Customer Satisfaction	It assists in early illness diagnosis and often notifies the user of their current health state. The system's enhanced heart disease decision making is advantageous both user and physician.
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	This solution works well with long and small datasets. It can also be modified to predict various other disease depending on the dataset.

3.4 Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People with heart disease.

6. CUSTOMER CONSTRAINTS CC

- Reduced economic expenses.
- Interactive dashboard.
- Immediate treatment

5. AVAILABLE SOLUTIONS AS

- Avoid smoking
- Take healthy foods
- Visit cardiologist in case of any symptoms
- Maintaining healthy exercise

Explore AS, differentiate

Focus on J&P, tap into BE, understand

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Coronary artery disease is a common heart condition that affects the major blood vessels that supply the heart muscle. Cholesterol deposits (plaques) in the heart arteries are usually the cause of coronary artery disease.

9. PROBLEM ROOT CAUSE RC

A buildup of fatty plaques in the arteries (atherosclerosis) is the most common cause of coronary artery disease. Risk factors include a poor diet, lack of exercise, obesity and smoking.

7. BEHAVIOUR BE

Chest pain or discomfort, Shortness of breath, Slow heartbeat, Lightheadedness, Swelling in the legs, belly area or areas around the eyes.

Focus on J&P, tap into BE, understand

Identify strong TR & EM

3. TRIGGERS TR

The thing that triggers our customer is that they immediately want an answer by a prediction method which predicts by knowing the current health condition.

4. EMOTIONS: BEFORE / AFTER EM

When people use our product they get a clarification Of the health condition right now so that according to the prediction they can immediately go to a physicist for consultation.

10. YOUR SOLUTION SL

We are using a prediction method which uses various attributes for predicting the status of heart disease with the use of our machine learning model to predict the immediate results.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

They use our dashboard to predict the status of heart disease.

8.2 OFFLINE

After the results have been predicted using the status of dashboard they can take a copy of the results and get a consultation from the physicist incase they have a heart problem

Identify strong TR & EM

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 through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Account creation	User can fill their Gmail and password for account creation.
FR-4	Personal details for account	Apart from the basic details ,user need to enter details such as name,age,height,sex,weight,previous medical records etc.
FR-5	Regular medical condition updation in app	Entry present medical records, symptoms etc.,
FR-6	Doctor consultation	Expert doctor consultation through app

4.2 Non-Functional Requirement

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application will have a simple and user friendly graphical interface. Users will be able to 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 technique known as database replication should be used so that all the important data will be safe. By building the app it will provide security, privacy and compliance by considering authentication, privilege management, secure data storage and communication, compliance and testing and installation.
NFR-3	**Reliability**	The application has to be consistent at every scenario and has to work without failure in any environment.
NFR-4	**Performance**	The performance of this project is to reduce heart disease death rate by earlier accurate disease

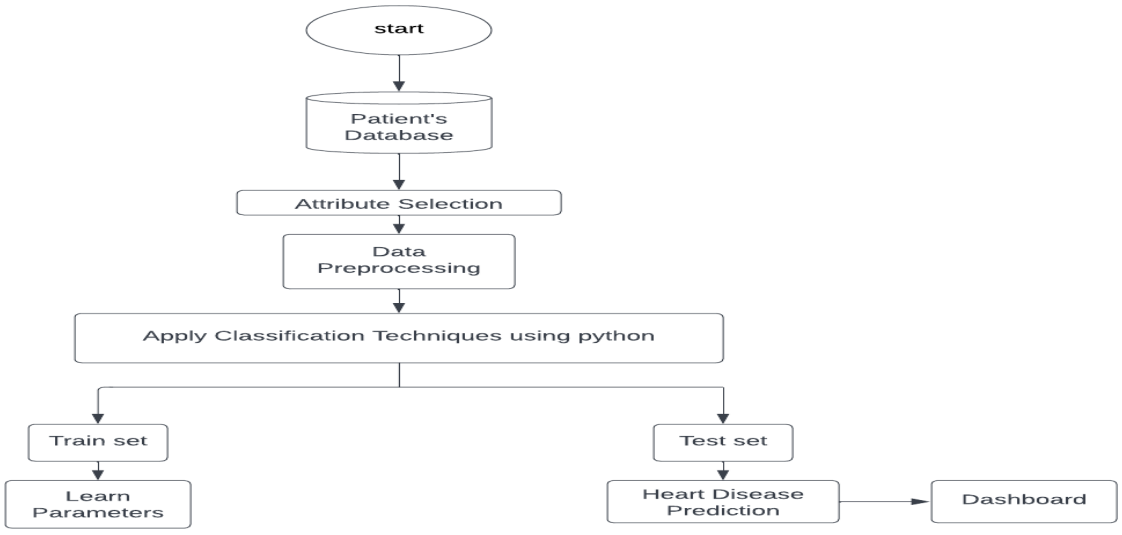
		prediction .Our solution offers services such as disease prevention, diagnosis and treatment, and rehabilitation.
NFR-5	**Availability**	Availability is important because, while there are often shortage in human resources, deployed providers are frequently inappropriately absent or present, are not actively delivering health care because they engaged with other duties. The application has available in 24 X 7 for users without any interruption.
NFR-6	**Scalability**	It can be integrated with smart electronic gadgets for further advancements which is very helpful for earlier prediction. Further, we can provide live doctor consultancy, keep up the old data records for increasing accurate prediction and to prevent heart disease .It notifies the people to nearby hospital when they are at risk.

CHAPTER 4

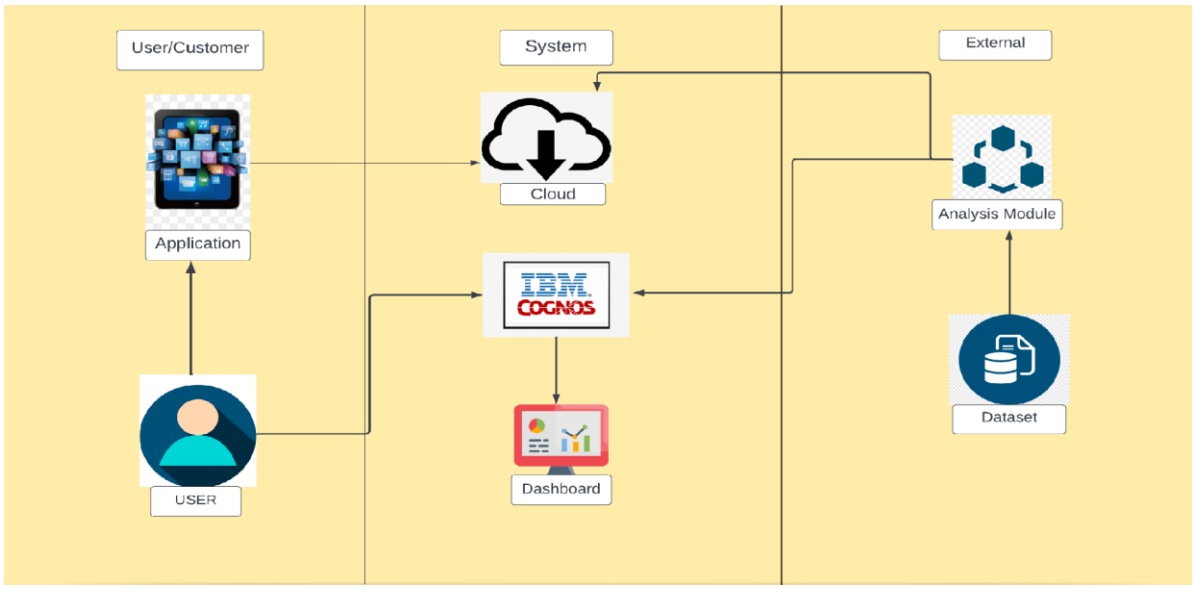
PROJECT DESIGN

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 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.



5.2 Solution and Technical Architecture



5.3 User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile 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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-1
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can register & access the dashboard with Gmail Login and password	High	Sprint-1
Customer (Web user)	Dashboard	USN-6	As a user , I 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-7	As a user, I can check the risk factors and prevention tips	I can read the prevention tips	High	Sprint-2
		USN-8	As a user, I can check the treatment options	I can read the treatment options	High	Sprint-2
Customer Care Executive	Helpdesk	USN-9	As a customer care executive,	I can post my queries in the dashboard	High	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			he/she can view the customer queries			
		USN-10	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-11	As an admin, he/she can update the health details of users.	I can view my updated health details	High	Sprint-4
		USN-12	As an admin, he/she can add or delete users	I can access my account/Dashboard when logged in	High	Sprint-4
		USN-13	As an admin, he/she can update the risk and prevention tips	I can update the risk factors and prevention tips	High	Sprint-4

CHAPTER 6

PROJECT PLANNING AND SCHEDULING

6.1 Script Planning and Execution

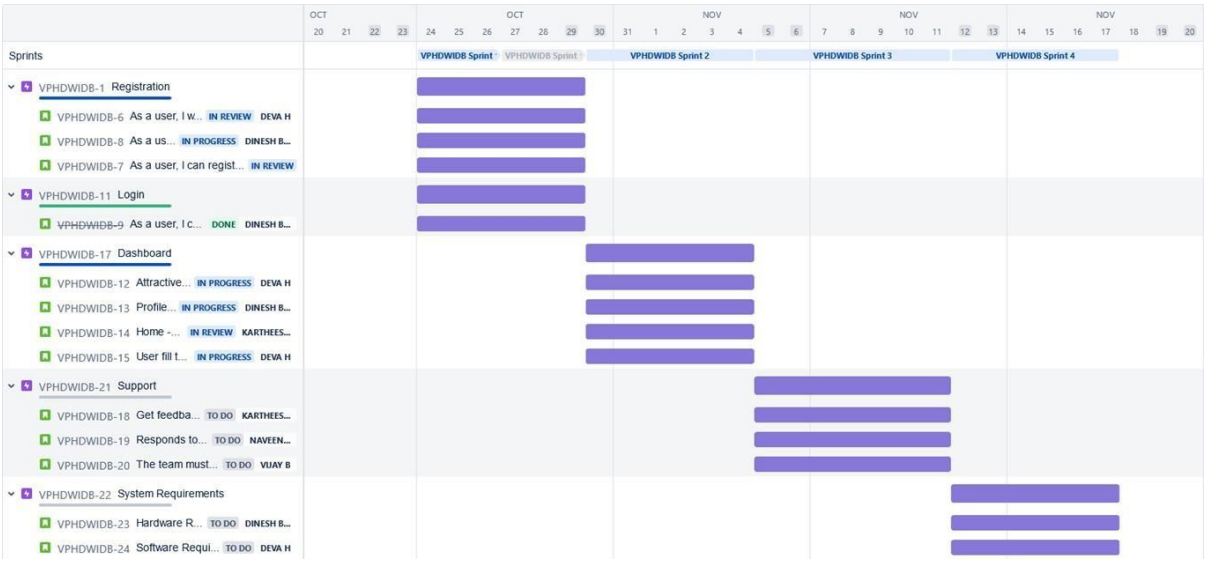
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint-1		USN-2	As a user, I will receive confirmation email onceI have registered for the application	1	High	2
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	2
Sprint-2		USN-4	As a user, I can register for the application through Gmail	2	Medium	2
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	1
Sprint-3	Dashboard	USN-6	View profile and update the profile	2	High	1
		USN-7	User can update password	1	Medium	1
		USN-8	User can analyse heart disease in home page	2	High	1
		USN-9	User has to enter the basic information to analyse their health status. User has to enter their age, gender, blood sugarlevel, chest pain type and other details	2	High	4
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
		USN-10	User can also view doctor details and book appointments	2	High	3
Sprint-3	Visualization	USN-11	User can visualize the results effectively	1	Medium	2

Sprint-4	Dashboard	USN-12	User can send feedback	2	Medium	2
		USN-13	User can contact toll free for any queries	1	Medium	2

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	19	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 Jira Report



CHAPTER 7
CODING AND SOLUTIONING

7.1 Machine Learning

PREDICTING HEART DISEASE USING MACHINE LEARNING ALGORITHMS

Step 1: Importing libraries Step 2: Loading the Dataset Step 3: Data Visualization Step 4: Data Preprocessing Step 5: Spliting into test and train set Step 6: Model Training Step 7: Performance Metrices Step 8: Predicting the best model based on accuracy metrix

Algorithms used

Logistic Regression, Random Forest, Light GBM, XGBoost, Decision tree

Step 1: Importing libraries

In [88]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Step 2: Loading the dataset

In [90]:
```
df = pd.read_csv('Heart.csv')
```

In [91]:
```
df.info()
```

```
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Age                     270 non-null    int64  
 1   Sex                     270 non-null    int64  
 2   Chest pain type         270 non-null    int64  
 3   BP                      270 non-null    int64  
 4   Cholesterol             270 non-null    int64  
 5   FBS over 120            270 non-null    int64  
 6   EKG results             270 non-null    int64  
 7   Max HR                  270 non-null    int64  
 8   Exercise angina         270 non-null    int64  
 9   ST depression           270 non-null    float64
 10  Slope of ST             270 non-null    int64  
 11  Number of vessels fluro  270 non-null    int64  
 12  Thallium                270 non-null    int64  
 13  Heart Disease           270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

In [92]:
```
df.describe()
```

Out[92]:

	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
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.00000	270.000000	270.000000	270.000000
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	1.05000	1.585185	0.670370	4.696296
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	1.14521	0.614390	0.943896	1.940659
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.00000	1.000000	0.000000	3.000000
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	0.00000	1.000000	0.000000	3.000000
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	0.80000	2.000000	0.000000	3.000000
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	1.60000	2.000000	1.000000	7.000000
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.20000	3.000000	3.000000	7.000000

In [93]:
```
df.head()
```

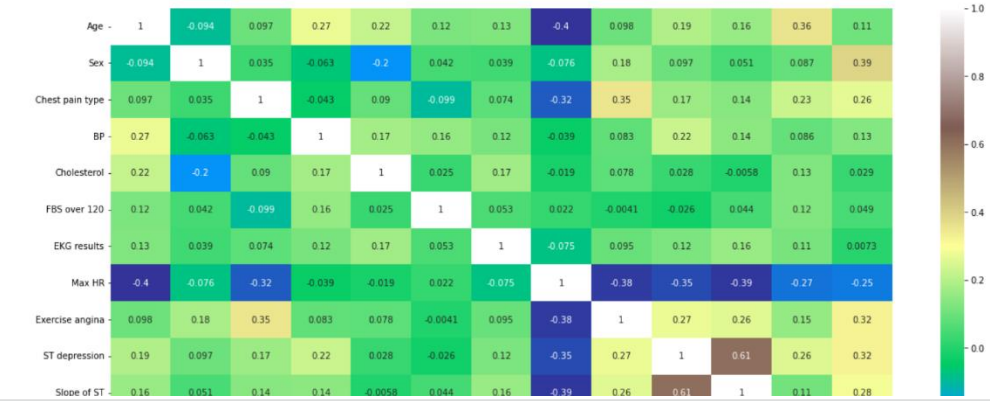
Out[93]:

	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	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

Step 3: Data Visualization

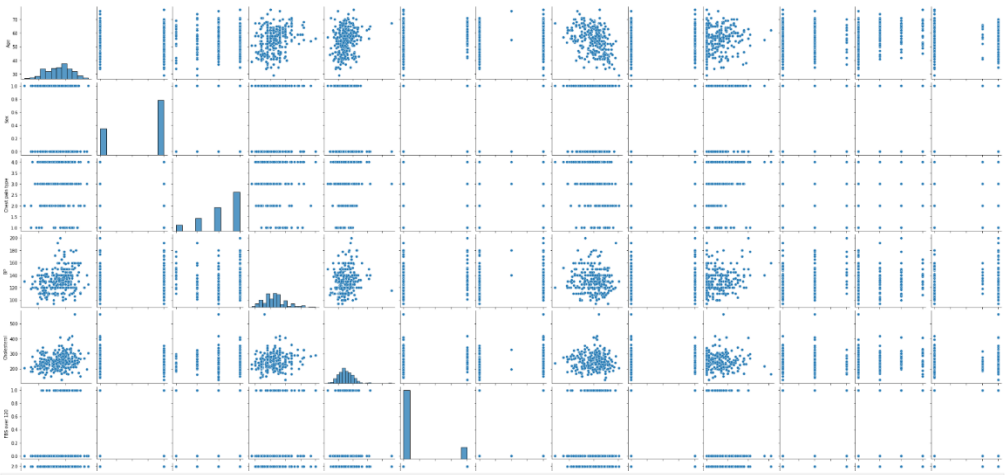
In [96]:
```
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(), annot=True, cmap='terrain')
```

Out[96]:

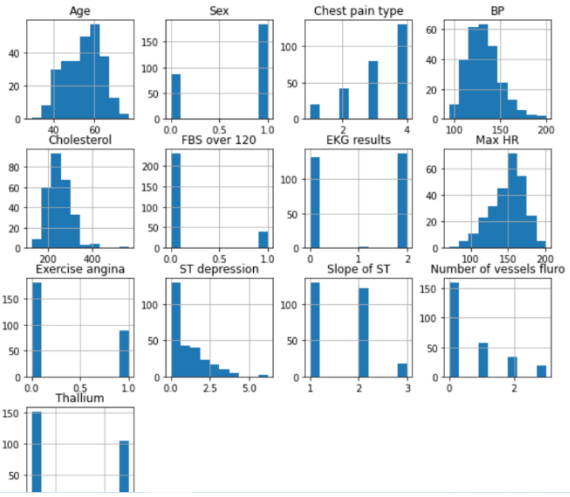


In [97]:
```
sns.pairplot(data=df)
```

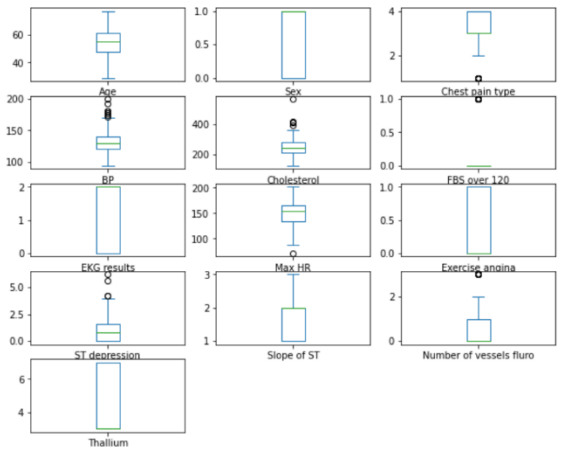
Out[97]:



In [98]:
```
df.hist(figsize=(10,12), layout=(5,4));
```

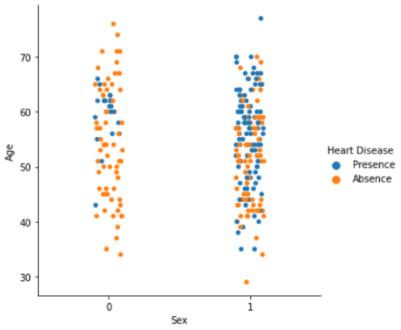


In [99]:
```
df.plot(kind='box', subplots=True, layout=(6,3), figsize=(10,10))
plt.show()
```

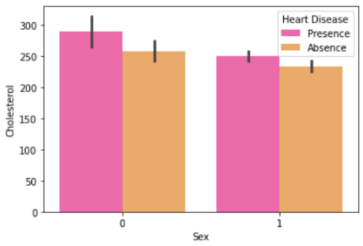


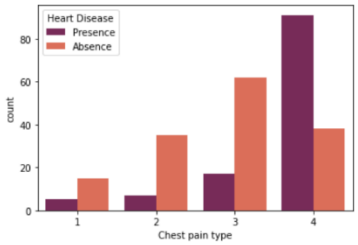
In [100…
```
sns.catplot(data=df, x='Sex', y='Age', hue='Heart Disease', palette='tab10')
```

Out[100…

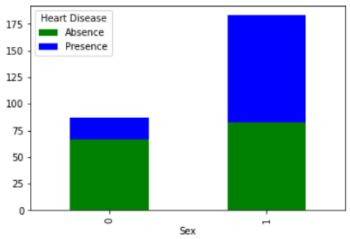


Out[101…





Out[105…



Step 4: Data Preprocessing

In [123…
```
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 [107…
```
df.head()
```

Out[107…

	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	Heart Disease
0	1.712094	1	4	130	1.402212	0	0.981664	109	0	2.4	2	2.472682	-0.875706	Presence
1	1.382140	0	3	115	6.093004	0	0.981664	160	0	1.6	2	-0.711535	1.189277	Absence
2	0.282294	1	2	124	0.219823	0	-1.026285	141	0	0.3	1	-0.711535	1.189277	Presence
3	1.052186	1	4	128	0.258589	0	-1.026285	105	1	0.2	2	0.349871	1.189277	Absence
4	2.152032	0	2	120	0.374890	0	0.981664	121	1	0.2	1	0.349871	-0.875706	Absence

In [132…
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
label = le.fit_transform(df['Heart Disease'])
label
```

Out[132…
```
array([1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0,
```

In [133…
```
df.drop("Heart Disease", axis=1, inplace=True)
df["Heart Disease"] = label
df
```

Out[133…

	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	Heart Disease
0	1.712094	1	4	130	1.402212	0	0.981664	109	0	2.4	2	2.472682	-0.875706	1
1	1.382140	0	3	115	6.093004	0	0.981664	160	0	1.6	2	-0.711535	1.189277	0
2	0.282294	1	2	124	0.219823	0	-1.026285	141	0	0.3	1	-0.711535	1.189277	1
3	1.052186	1	4	128	0.258589	0	-1.026285	105	1	0.2	2	0.349871	1.189277	0
4	2.152032	0	2	120	0.374890	0	0.981664	121	1	0.2	1	0.349871	-0.875706	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
265	-0.267629	1	3	172	-0.981951	1	-1.026285	162	0	0.5	1	-0.711535	1.189277	0
266	-1.147506	1	2	120	0.258589	0	-1.026285	173	0	0.0	1	-0.711535	1.189277	0
267	0.172309	0	2	140	0.859476	0	0.981664	153	0	1.3	2	-0.711535	-0.875706	0
268	0.282294	1	4	140	-1.117635	0	-1.026285	148	0	0.4	2	-0.711535	0.673032	0
269	1.382140	1	4	160	0.704409	0	0.981664	108	1	1.5	2	2.472682	-0.875706	1

270 rows × 14 columns

Step 5: Splitting into train and test data

In [135…
```
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.3, random_state=40)
```

In [136…
```
print('x_train-', x_train.size)
print('x_test-', x_test.size)
print('y_train-', y_train.size)
print('x_test-', x_test.size)
```

```
x_train- 2457
x_test- 1053
y_train- 189
x_test- 1053
```

Step 6: Model Training

Algorithm 1:Logistic Regression

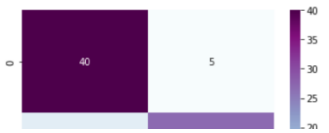
In [137…
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
model1=lr.fit(x_train,y_train)
prediction1=model1.predict(x_test)
```

In [138…
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,prediction1)
cm
```

Out[138…
```
array([[40,  5],
       [ 9, 27]], dtype=int64)
```

In [139…
```
sns.heatmap(cm, annot=True,cmap='BuPu')
```

Out[139…



In [143…
```
print('LR  :', accuracy_score(y_test, prediction1))
```

```
LR  : 0.8271604938271605
```

Algorithm 2: Random Forest

In [144…
```
import sklearn
from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators=2,min_samples_split=3,min_samples_leaf=2)
clf.fit(x_train,y_train)
pred=clf.predict(x_test)
```

In [145…
```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.84      0.77        45
           1       0.74      0.56      0.63        36

    accuracy                           0.72        81
   macro avg       0.72      0.70      0.70        81
weighted avg       0.72      0.72      0.71        81
```

Algorithm 3 : Light GBM

In [146…
```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in c:\users\dell\anaconda3\lib\site-packages (3.3.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.20.3)
Requirement already satisfied: wheel in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (2.2.0)
```

In [147…
```
from lightgbm import LGBMClassifier
lgbmc=LGBMClassifier()
lgbmc.fit(x_train,y_train)
y_pred=lgbmc.predict(x_test)
```

In [121…
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

      Absence       0.82      0.80      0.81        45
     Presence       0.76      0.78      0.77        36

    accuracy                           0.79        81
   macro avg       0.79      0.79      0.79        81
weighted avg       0.79      0.79      0.79        81
```

Algorithm 4: XGBoost

In [167…
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\dell\anaconda3\lib\site-packages (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.20.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.7.1)
```

In [169…
```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.80      0.82        45
           1       0.76      0.81      0.78        36

    accuracy                           0.80        81
   macro avg       0.80      0.80      0.80        81
weighted avg       0.80      0.80      0.80        81
```

Algorithm 5: Decision Tree

In [159…

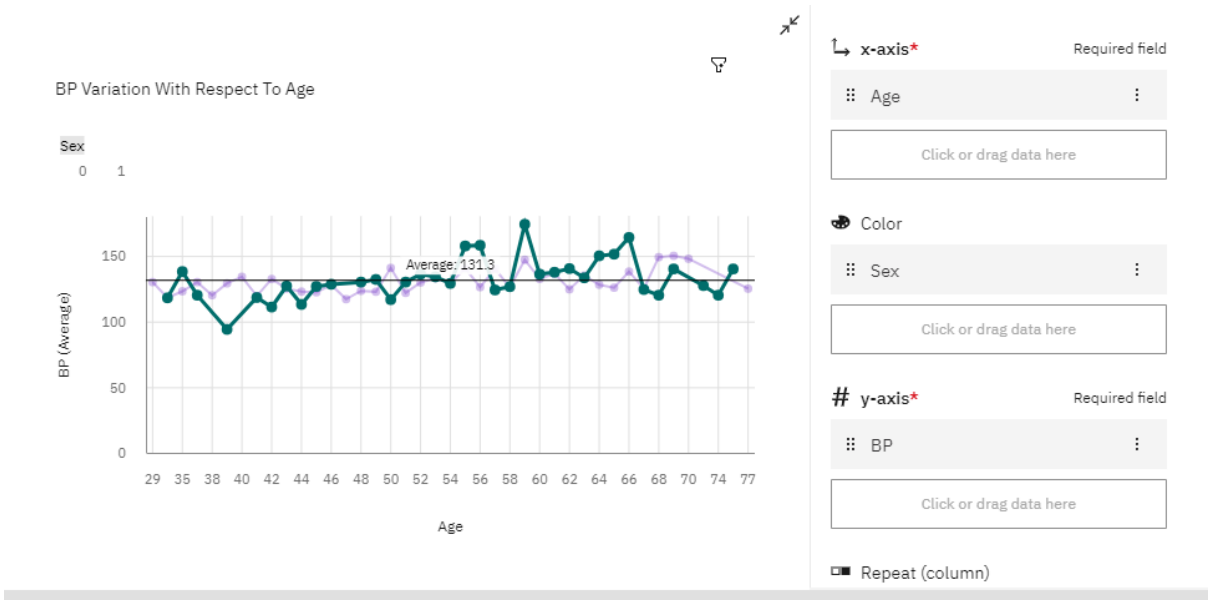
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(x_train, y_train)
predict1=regressor.predict(x_test)
```

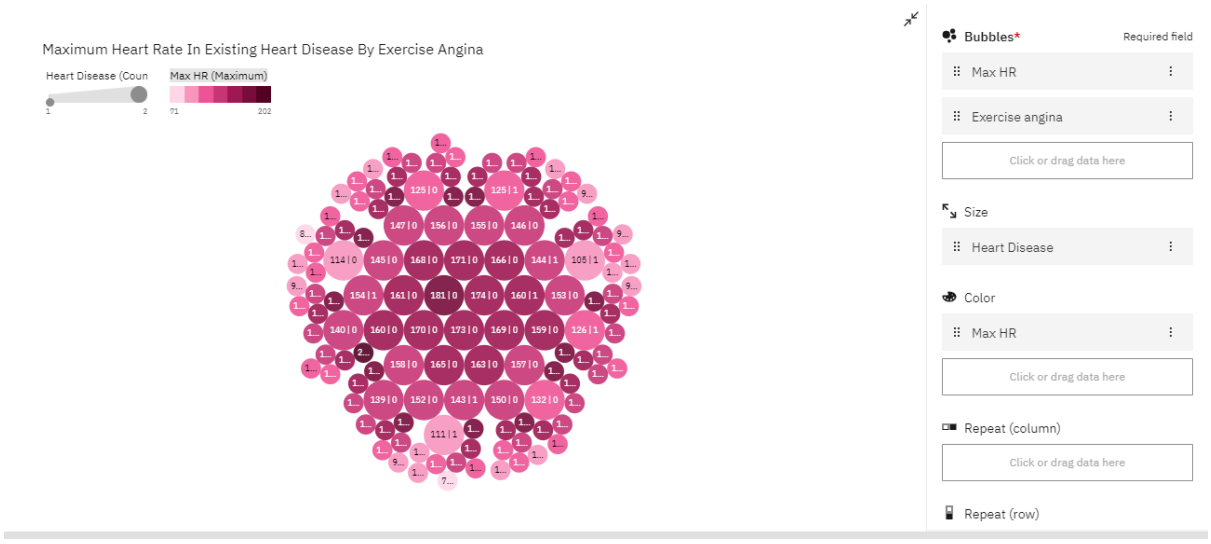
In [160…

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict1))
```

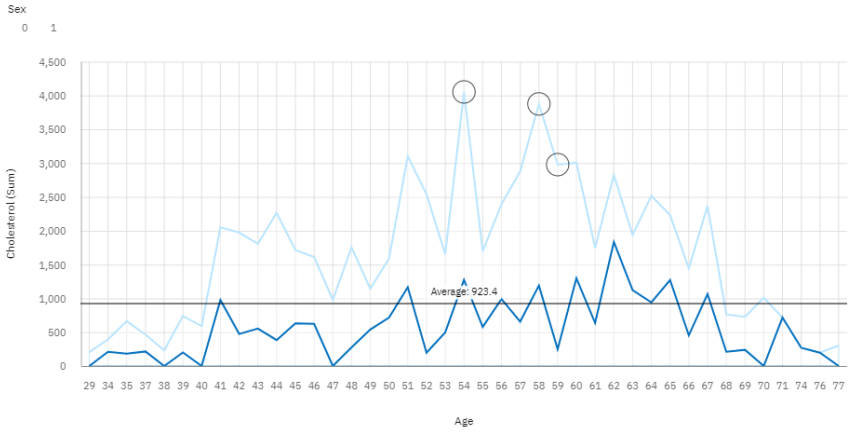
```
0.7037037037037037
```

7.2 Dashboard





Serum Cholesterol Levels Vs Age



Visualization properties

Visualization | General

Color

Current color palette

Cyan

Change color palette

Element color #0873C1

Axis

Legend

Chart

Data handling: Stacked

Average Age For Different Types Of Chest Pain In Existing Heart Diseases

Age	0					1	
	1	2	3	4	Summary	1	2
Absence	63.25	51	54.97	55	**54.58**	52.91	
Presence	(no value)	58	62	59.28	**59.35**	56.4	
Summary	**63.25**	**51.44**	**55.19**	**57.2**	**55.68**	**54**	

Columns

Sex

Chest pain type

Click or drag data here

Rows

Heart Disease

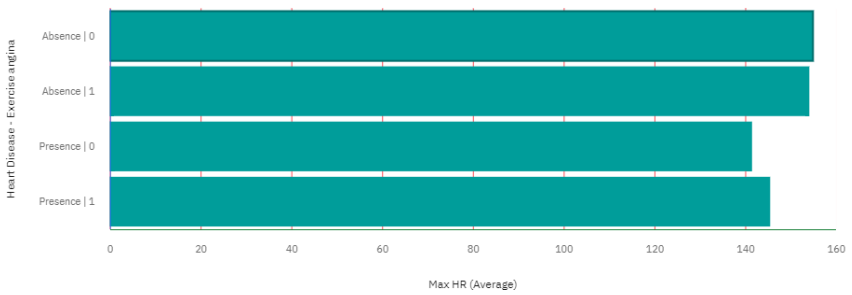
Click or drag data here

Values* Required field

Age

Click or drag data here

Effect Of Existing Heart Disease On Average Of Exercise Angina



Bars

Heart Disease

Exercise angina

Click or drag data here

Length* Required field

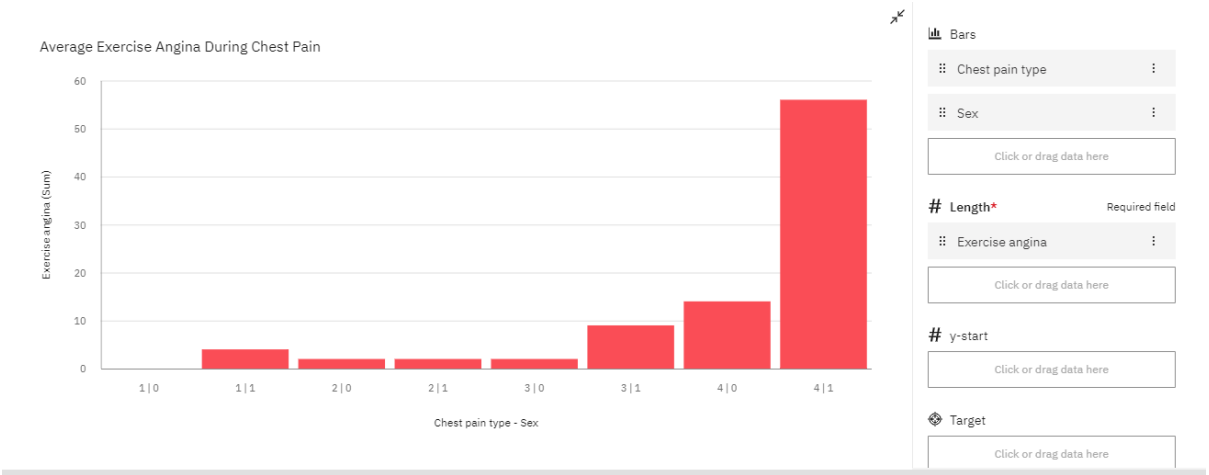
Max HR

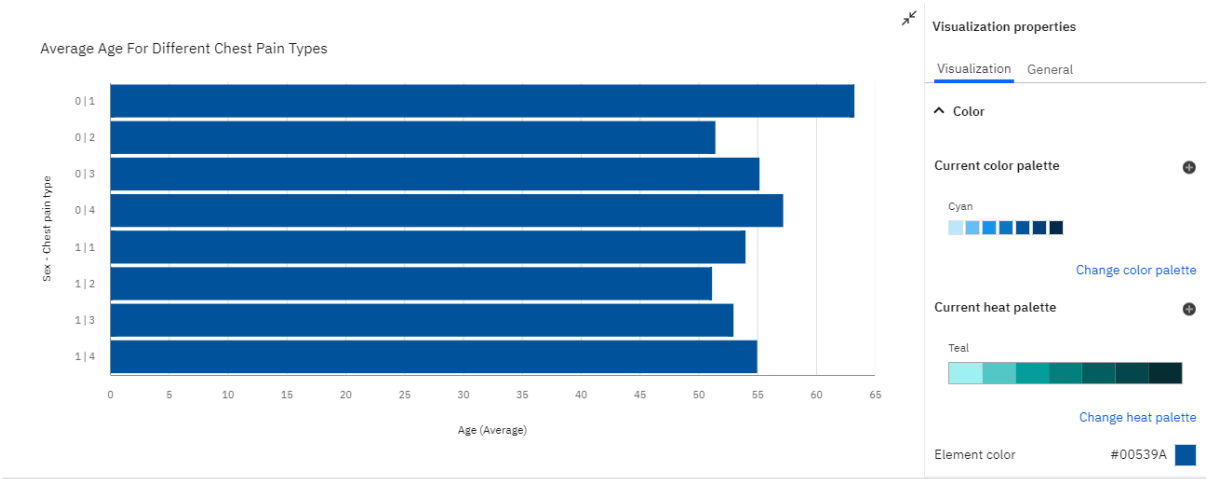
Click or drag data here

Color

Click or drag data here

Repeat (column)


Average Exercise Angina During Chest Pain
Exercise angina (Sum)
60
50
40
30
20
10
0
1 | 0
1 | 1
2 | 0
2 | 1
3 | 0
3 | 1
4 | 0
4 | 1
Chest pain type - Sex
Bars
Chest pain type
Sex
Click or drag data here
Length*
Required field
Exercise angina
Click or drag data here
y-start
Click or drag data here
Target
Click or drag data here


Average Age For Different Chest Pain Types
Sex - Chest pain type
0 | 1
0 | 2
0 | 3
0 | 4
1 | 1
1 | 2
1 | 3
1 | 4
0
5
10
15
20
25
30
35
40
45
50
55
60
65
Age (Average)
Visualization properties
Visualization
General
Color
Current color palette
Cyan
Change color palette
Current heat palette
Teal
Change heat palette
Element color
#00539A


Tab 1
+
Average Age For Different Chest Pain Types
Average Exercise Angina During Chest Pain
BP Variation With Respect To Age
Sex
0
1
Average: 131.3
Effect Of Existing Heart Disease On Average Of Exercise Angina
Average Age For Different Types Of Chest Pain In Existing Heart Diseases
Age
3
Presence
58
Serum Cholesterol Levels Vs Age
Sex
0
1
Average: 923.4
Maximum Heart Rate In Existing Heart Disease By Exercise Angina
Heart Disease (Coun
Bars
Sex
Chest pain type
Click or drag data here
Length*
Required field
Age
Click or drag data here
y-start
Click or drag data here
Target
Click or drag data here
Color

CHAPTER 8

TESTING

8.1 Test Cases

Testing the data model for various input values.

In [140…
```
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))
```

```
Testing Accuracy: 0.9382716049382716
```

In [141…
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,prediction1)
```

Out[141…
```
0.8271604938271605
```

Algorithm 2: Random Forest

In [144…
```
import sklearn
from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators=2,min_samples_split=3,min_samples_leaf=2)
clf.fit(x_train,y_train)
pred=clf.predict(x_test)
```

In [145…
```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.84      0.77        45
           1       0.74      0.56      0.63        36

    accuracy                           0.72        81
   macro avg       0.72      0.70      0.70        81
weighted avg       0.72      0.72      0.71        81
```

Algorithm 3 : Light GBM

In [146…
```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in c:\users\dell\anaconda3\lib\site-packages (3.3.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.20.3)
Requirement already satisfied: wheel in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (2.2.0)
```

In [147…
```
from lightgbm import LGBMClassifier
lgbmc=LGBMClassifier()
lgbmc.fit(x_train,y_train)
y_pred=lgbmc.predict(x_test)
```

In [121…
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

     Absence       0.82      0.80      0.81        45
    Presence       0.76      0.78      0.77        36

    accuracy                           0.79        81
   macro avg       0.79      0.79      0.79        81
weighted avg       0.79      0.79      0.79        81
```

Algorithm 4: XGBoost

In [167…
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\dell\anaconda3\lib\site-packages (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.20.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.7.1)
```

In [169…
```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.80      0.82        45
           1       0.76      0.81      0.78        36

    accuracy                           0.80        81
   macro avg       0.80      0.80      0.80        81
weighted avg       0.80      0.80      0.80        81
```

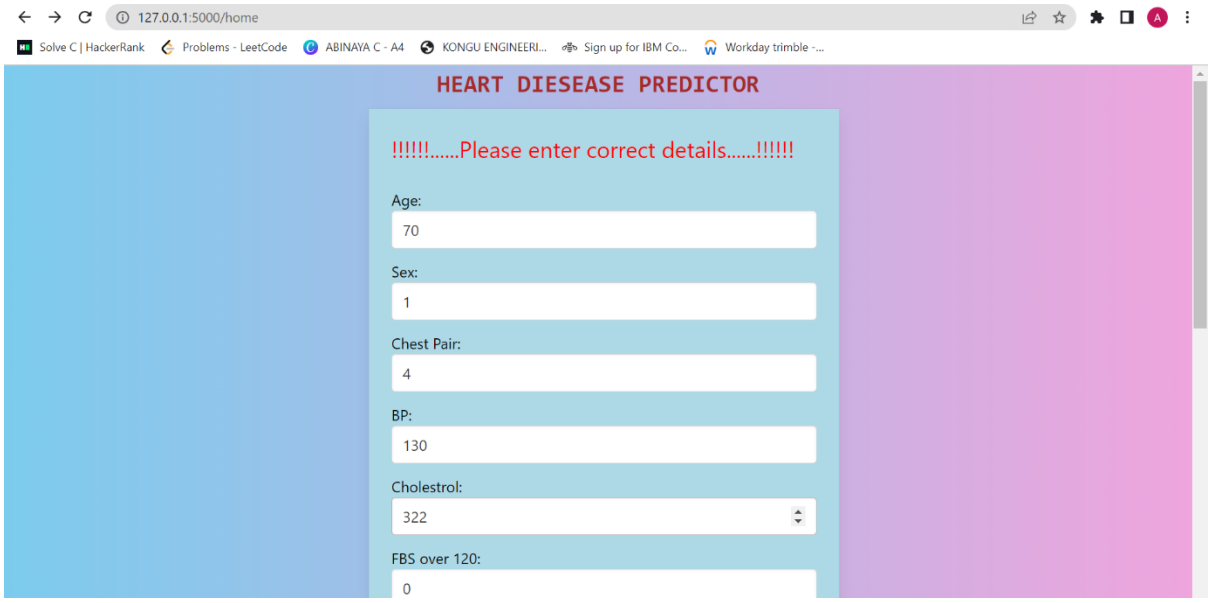
Algorithm 5: Decision Tree

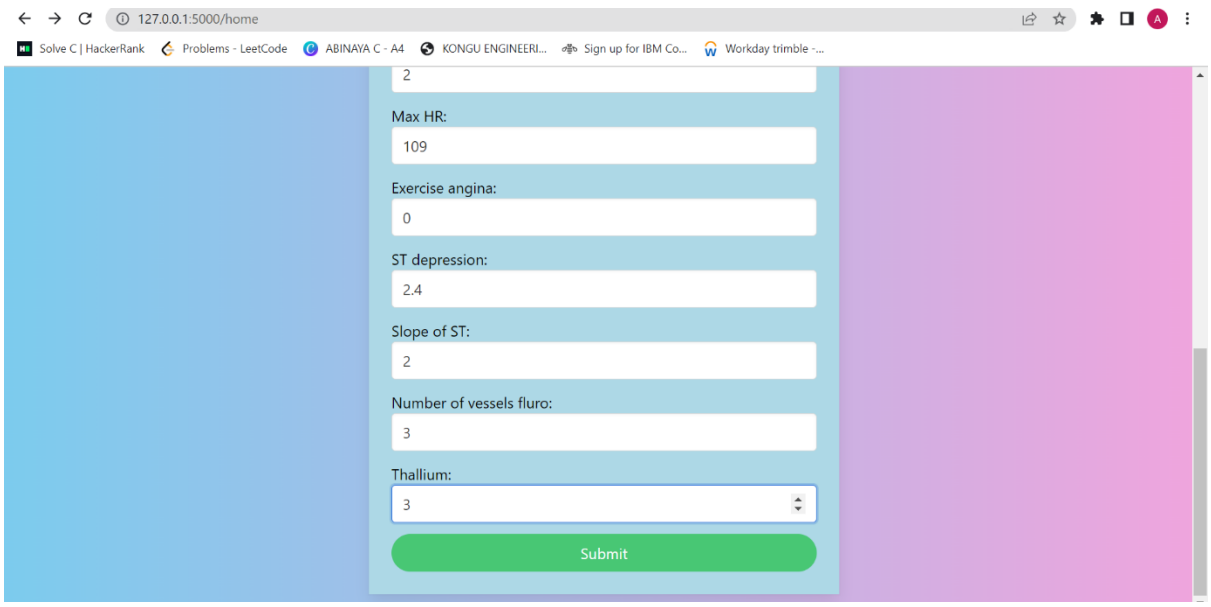
In [159…
```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(x_train, y_train)
predict1=regressor.predict(x_test)
```

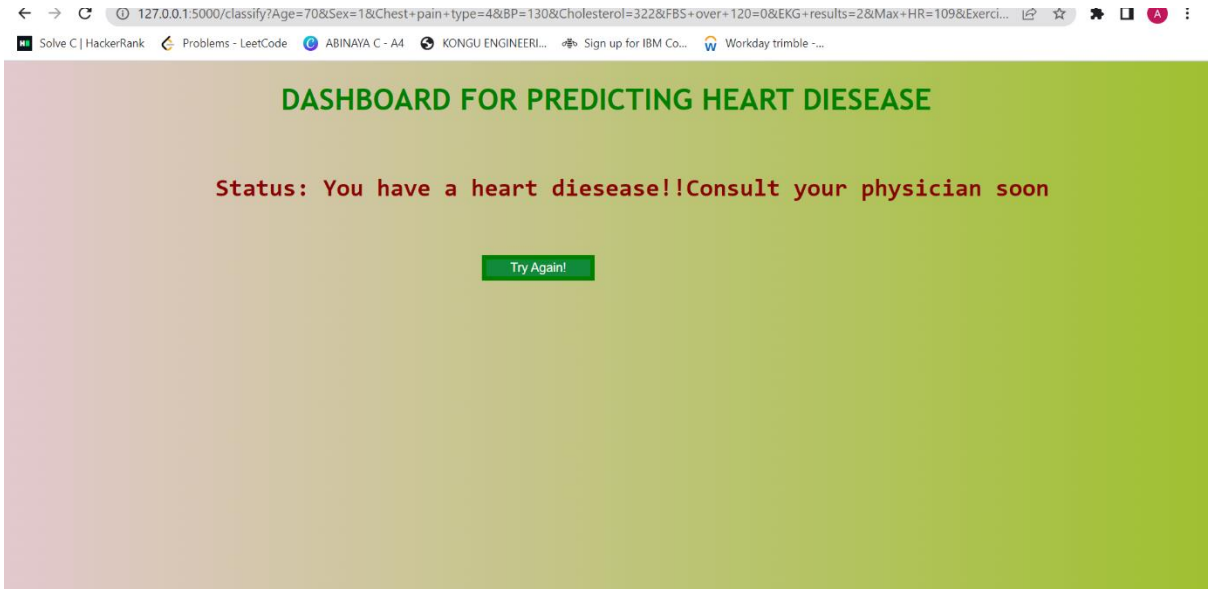
In [160…
```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict1))
```

```
0.7037037037037037
```

8.2 User acceptance Testing






127.0.0.1:5000/classify?Age=70&Sex=1&Chest+pain+type=4&BP=130&Cholesterol=322&FBS+over+120=0&EKG+results=2&Max+HR=109&Exerci...
Solve C | HackerRank
Problems - LeetCode
ABINAYA C - A4
KONGU ENGINEERI...
Sign up for IBM Co...
Workday trimble -...
DASHBOARD FOR PREDICTING HEART DIESEASE
Status: You have a heart diesease!!Consult your physician soon
Try Again!

CHAPTER 9
RESULTS

9.1 Performance Metrices

In [140…
```
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))
```

```
Testing Accuracy: 0.9382716049382716
```

In [141…
```
from sklearn.metrics import accuracy_score
accuracy_score(y_test,prediction1)
```

Out[141…
```
0.8271604938271605
```

Algorithm 2: Random Forest

In [144…
```
import sklearn
from sklearn.ensemble import RandomForestClassifier
clf=RandomForestClassifier(n_estimators=2,min_samples_split=3,min_samples_leaf=2)
clf.fit(x_train,y_train)
pred=clf.predict(x_test)
```

In [145…
```
from sklearn.metrics import classification_report
print(classification_report(y_test, pred))
```

```
              precision    recall  f1-score   support

           0       0.70      0.84      0.77        45
           1       0.74      0.56      0.63        36

    accuracy                           0.72        81
   macro avg       0.72      0.70      0.70        81
weighted avg       0.72      0.72      0.71        81
```

Algorithm 3 : Light GBM

In [146…
```
!pip install lightgbm
```

```
Requirement already satisfied: lightgbm in c:\users\dell\anaconda3\lib\site-packages (3.3.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (1.20.3)
Requirement already satisfied: wheel in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.37.0)
Requirement already satisfied: scikit-learn!=0.22.0 in c:\users\dell\anaconda3\lib\site-packages (from lightgbm) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda3\lib\site-packages (from scikit-learn!=0.22.0->lightgbm) (2.2.0)
```

In [147…
```
from lightgbm import LGBMClassifier
lgbmc=LGBMClassifier()
lgbmc.fit(x_train,y_train)
y_pred=lgbmc.predict(x_test)
```

In [121…
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

     Absence       0.82      0.80      0.81        45
    Presence       0.76      0.78      0.77        36

    accuracy                           0.79        81
   macro avg       0.79      0.79      0.79        81
weighted avg       0.79      0.79      0.79        81
```

Algorithm 4: XGBoost

In [167…
```
!pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\users\dell\anaconda3\lib\site-packages (1.7.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.20.3)
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.7.1)
```

In [169…
```
from xgboost import XGBClassifier
model = XGBClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.80      0.82        45
           1       0.76      0.81      0.78        36

    accuracy                           0.80        81
   macro avg       0.80      0.80      0.80        81
weighted avg       0.80      0.80      0.80        81
```

Algorithm 5: Decision Tree

In [159…

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(x_train, y_train)
predict1=regressor.predict(x_test)
```

In [160…

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predict1))
```

```
0.7037037037037037
```

CHAPTER 10
ADVANTAGES & DISADVANTAGES

Advantages:

- This is one of the fastest ways to determine if a person is likely to suffer from a heart diseaseor not.
- Useful for medical practitioners to easily classify their patients.
- User Friendly
- Easy to understand
- Secure
- Dashboard provides insightful informations

Disadvantages:

- Needs work
- Users need to know all the fields
- Does Not take null value as input
- Does not provide suggestions to user

CHAPTER 11

CONCLUSION

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients.

CHAPTER 12

FUTURE SCOPE

Future enhance of the HDPS is to predict a specific HD type such Heart attracts, CVD, CAD, etc. the potential of the HDPS in a different area are hospital, Clinic, smartphone, smart wear, hospital/police emergency system and integrate with fitness mobile application. We will integrate this model in hospital and clinic system to predict heart disease. We will implement this HDP Model into smart wears to detect essential attributes of HD and suggest to the precaution of HD. we will also apply this model into a mobile app to easily test ourselves HD. we will integrate smart wear to the hospital and police emergency system to save the life of the patient at the emergency condition.

CHAPTER 13

APPENDIX

13.1 Source code

Index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Login Page</title>


    <!-- Owl-Carousel -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.car
ousel.min.css"
        integrity="sha256-UhQQ4fxEeABh4JrcmAJ1+16id/1dnlOEVCFOxDef9Lw="
crossorigin="anonymous" />
    <link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets
/owl.theme.default.min.css"
        integrity="sha256-kksNxjDRxd/5+jGurZUJd1sdR2v+ClrCl3svESBaJqw="
crossorigin="anonymous" />

    <!-- Font Awesome CDN -->
    <script src="https://kit.fontawesome.com/23412c6a8d.js"></script>

    <!-- Custom Style-->
    <link rel="stylesheet" href="../css/Style.css">
</head>

<body>

    <div class="container">
        <div class="panel">
            <div class="row">
                <div class="col liquid">
                    <h4><i class="fas fa-drafting-compass"></i> HEALTH
CARE</h4>

                    <!-- Owl-Carousel -->

                    <div class="owl-carousel owl-theme">
```

```
                        <img src="../assets/loginimg.svg" alt=""
class="login_img">
                        <img src="../assets/heart.png" alt=""
class="login_img">
                        <img src="../assets/save.jpg" alt=""
class="login_img">
                    </div>

                    <!-- /Owl-Carousel -->

                    <!-- <div class="follow">
                        Follow us <i class="fab fa-facebook-f"></i>
                        <i class="fab fa-twitter"></i>
                    </div> -->
                </div>
                <div class="col login">

                    <form>
                        <div class="titles">
                            <h6>We keep everything</h6>
                            <h3>Ready to Login</h3>
                        </div>
                        <div class="form-group">
                            <input type="text" placeholder="Email"
class="form-input">
                            <div class="input-icon">
                                <i class="fas fa-user"></i>
                            </div>
                        </div>
                        <div class="form-group">
                            <input type="password" placeholder="Password"
class="form-input">
                            <div class="input-icon">
                                <i class="fas fa-user-lock"></i>
                            </div>
                        </div>
                        <br>
                        <a href="register.html">
                            <center><input type="button" value=" LOGIN "
style="border:5px solid green;background-color:#118a3b;"></input></center>
                        </a>


                    </form>

                </div>
            </div>
        </div>
```

```
    </div>

    <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/owl.carousel.min.js"
        integrity="sha256-pTxD+DSzIwmwhOqTFN+DB+nHjO4iAsbgfyFq5K5bcE0="
crossorigin="anonymous"></script>

    <script>
        $(document).ready(function () {

            $('.owl-carousel').owlCarousel({
                loop: true,
                autoplay: true,
                autoplayTimeout: 2000,
                autoplayHoverPause: true,
                items: 1
            });
        });
    </script>
</body>

</html>
```

Register.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Registration Form</title>
    <meta name="viewport" content="width=device-width,
      initial-scale=1.0"/>
    <link rel="stylesheet" href="../css/style_reg.css" />
  </head>
  <body>
    <div class="container">
        <h3 style="color:cyan; text-align: center;">Please complete the
registration process to continue!!</h3>
    <br>
      <h1 class="form-title">Registration</h1>
      <form action="#">
        <div class="main-user-info">
          <div class="user-input-box">
            <label for="fullName"> Name</label>
            <input type="text"
```

```
                id="fullName"
                name="fullName"
                placeholder="Enter your Name"/>
      </div>
      <div class="user-input-box">
        <label for="username">Blood Group</label>
        <input type="text"
                id="username"
                name="username"
                placeholder="Enter your blood group"/>
      </div>
      <div class="user-input-box">
        <label for="email">Email</label>
        <input type="email"
                id="email"
                name="email"
                placeholder="Enter your Email"/>
      </div>
      <div class="user-input-box">
        <label for="phoneNumber">Phone Number</label>
        <input type="text"
                id="phoneNumber"
                name="phoneNumber"
                placeholder="Enter your Phone Number"/>
      </div>
      <div class="user-input-box">
        <label for="password">Age</label>
        <input type="text"
                id="password"
                name="password"
                placeholder="Enter your Age"/>
      </div>

    </div>
    <div class="gender-details-box">
      <span class="gender-title">Gender</span>
      <div class="gender-category">
        <input type="radio" name="gender" id="male">
        <label for="male">Male</label>
        <input type="radio" name="gender" id="female">
        <label for="female">Female</label>
        <input type="radio" name="gender" id="other">
        <label for="other">Other</label>
      </div>
    </div>
  <center>
      <a href="http://127.0.0.1:5000/home">
```

```
          <input type="button" value=" REGISTER " style="border:5px solid
green;background-color:#118a3b;">
          </a>
        </center>

      </form>
    </div>
  </body>
</html>
```

Model.py

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
data = pd.read_csv('./Heart.csv')
variety_mappings = {0: 'Absence', 1: 'Presence'}

# Encoding the target variables to integers
data = data.replace(['Absence', 'Presence' ], [0, 1])

X = data.iloc[:, 0:-1]
y = data.iloc[:, -1]

logreg = LogisticRegression(max_iter=1000)
logreg.fit(X, y)

def classify(a, b, c, d,e,f,g,h,i,j,k,l,m):
    arr = np.array([a, b, c, d,e,f,g,h,i,j,k,l,m]) # Convert to numpy array
    arr = arr.astype(np.float64) # Change the data type to float
    query = arr.reshape(1, -1) # Reshape the array
    prediction = variety_mappings[logreg.predict(query)[0]] # Retrieve from
dictionary
    return prediction
```

Server.py

```
import model
import pandas as pd
import numpy as np
from flask import Flask, request, render_template

app = Flask(__name__,template_folder="templates")

# Default route set as 'home'
@app.route('/home')
def home():
    return render_template('home.html')

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

```
def classify_type():
    try:
        age = request.args.get('Age') # Get parameters for sepal length
        sex = request.args.get('Sex') # Get parameters for sepal width
        chest_pair = request.args.get('Chest pain type') # Get parameters for
petal length
        bp = request.args.get('BP') # Get parameters for petal width
        cholestrol = request.args.get('Cholesterol')
        FBS_over_120 = request.args.get('FBS over 120')
        EKG_results = request.args.get('EKG results')
        Max_HR = request.args.get('Max HR')
        Exercise_angina = request.args.get('Exercise angina')
        ST_depression = request.args.get('ST depression')
        Slope = request.args.get('Slope of ST')
        Num = request.args.get('Number of vessels fluro')
        Thallium = request.args.get('Thallium')
        result=""
        # Get the output from the classification model
        variety =
model.classify(age,sex,chest_pair,bp,cholestrol,FBS_over_120,EKG_results,Max_H
R,Exercise_angina,ST_depression,Slope,Num,Thallium)
        if variety == 'Presence':
            result+="Status: You have a heart diesease!!Consult your physician
soon"
        else:
            result+="Status: You are healthy and free from heart diesease!!"
        # Render the output in new HTML page
        return render_template('output.html', variety=result)
    except:
        return 'Error'

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

home.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Heart diesease predictor</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.9.0/css/bulma.min.css">
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <style>
        /* html{
            overflow: hidden;
```

```
        } */

        body{

            height: 1200px;
             background-size: cover;
            background-repeat: no-repeat;
            /* background-color: aquamarine; */
            background-image:linear-gradient(to right, rgb(124, 204, 238),
#ee9edbec);
             /* background: url('../templates/register1.jpg') ;
              */
        }

        #login-form-container{
            position: absolute;
            top:300px;
            margin-top:2%;
            margin-left: 3%;
            width: 1200px;
            height: 100%;
            display: flex;
            align-items: center;
            justify-content: center;

        }
        h1{
            text-align: center;
            color: brown;
            font-family: monospace;
            font-weight: bolder;
            font-size: xx-large;
        }
    </style>
</head>
<body>
    <h1>HEART DIESEASE PREDICTOR</h1>
    <br>
    <br>
    <br>
    <div id="login-form-container">
        <form action="classify" method="GET">
            <div class="card" style="width: 500px; background-color:lightblue;
color:black">
            <div class="card-content">
                <div class="media">
                <div class="is-size-4 has-text-centered"
style="color:red">!!!!!!.....Please enter correct details.....!!!!!!</div>
```

```
                </div>
                <div class="content">

                <div class="field">
                    <p class="control">
                    Age: <input class="input" type="number" value='0.00'
step='0.01' name="Age" id="Age">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                    Sex: <input class="input" type="number" value='0.00'
step='0.01' name="Sex" id="Sex">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                    Chest Pair: <input class="input" type="number"
value='0.00' step='0.01' name="Chest pain type" id="Chest pain type">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                    BP: <input class="input" type="number" value='0.00'
step='0.01' name="BP" id="BP">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                    Cholestrol: <input class="input" type="number"
value='0.00' step='0.01' name="Cholesterol" id="Cholesterol">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        FBS over 120: <input class="input" type="number"
value='0.00' step='0.01' name="FBS over 120" id="FBS over 120">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
```

```
                        EKG results: <input class="input" type="number"
value='0.00' step='0.01' name="EKG results" id="EKG results">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        Max HR: <input class="input" type="number"
value='0.00' step='0.01' name="Max HR" id="Max HR">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        Exercise angina: <input class="input" type="number"
value='0.00' step='0.01' name="Exercise angina" id="Exercise angina">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        ST depression: <input class="input" type="number"
value='0.00' step='0.01' name="ST depression" id="ST depression">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        Slope of ST: <input class="input" type="number"
value='0.00' step='0.01' name="Slope of ST" id="Slope of ST">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        Number of vessels fluro: <input class="input"
type="number" value='0.00' step='0.01' name="Number of vessels fluro"
id="Number of vessels fluro">
                    </p>
                </div>

                <div class="field">
                    <p class="control">
                        Thallium: <input class="input" type="number"
value='0.00' step='0.01' name="Thallium" id="Thallium">
                    </p>
                </div>
```

```
                <div class="field">

                    <button class="button is-fullwidth is-rounded is-
success">Submit</button>

                </div>
                </div>
            </div>
        </form>
    </div>
</body>
</html>
```

Output.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Heart diesease</title>
    <style>
        .media{

            text-align: center;
            color:darkred;
            width: 1000px;
            font-family: monospace;
            font-size:xx-large;
            font-weight:bolder;
            margin-top: 15%;
            margin-left: 40%;
        }
        body{
            background-image:linear-gradient(to right, #E3C9CEFF, #9FC131FF);
            color:green;
            text-align: center;
            font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Arial, sans-serif;
        }


    </style>
    </head>

<body>
    <h1>DASHBOARD FOR PREDICTING HEART DIESEASE</h1>
    <div id="login-form-container">
        <div class="card" style="width: 400px">
```

```
            <div class="card-content">
                <div class="media">
                    <div class="is-size-4 has-text-centered">
                        {{ variety }}
                    </div>
                </div>
                <br>
                <br>
                <br>

                <form action="home">
                    <div class="field">
                        <button class="button is-fullwidth is-rounded is-
success" style="border:5px solid green;color:white;background-color:#118a3b;
margin-left: 500px; width: 30%;">Try Again!</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</body>
```

GitHub link :
https://github.com/IBM-EPBL/IBM-Project-22328-1659849255

Project demo link:
https://drive.google.com/file/d/1nUmdtvlqWj2y1392dOPgp7wd4iLG_xtJ/view?usp=sharing