Visualizing and Predicting Heart Diseases with an Interactive Dash Board Technology

1. **INTRODUCTION**

1.1 Project Overview

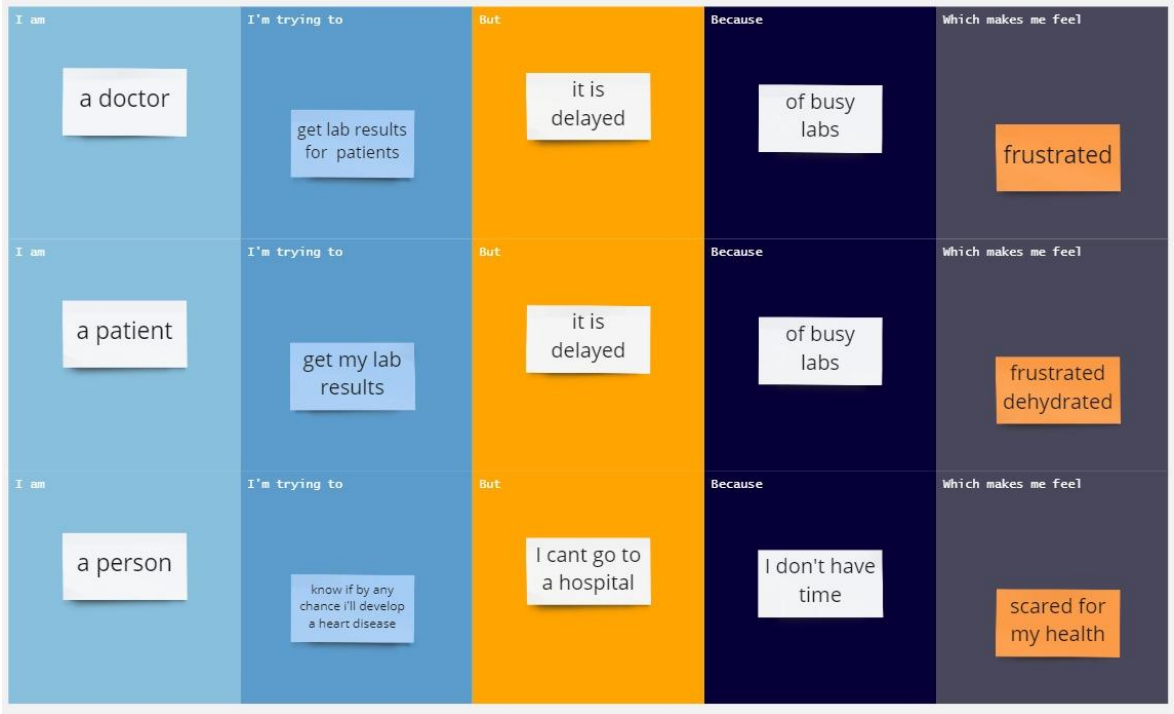
Predicting and diagnosing heart disease is the biggest challenge in the medical industry and it is based on factors like physical examination, symptoms and signs of the patient etc. As per this project we will be using data analytics and creating a dashboard with visualizations that will help us predict heart disease.

1.2 Purpose

We are creating a web application to predict the diseases an individual suffering based on the
symptoms by accepting the symptoms as input and checking from the already predict results
obtained by visualization that have been stored as a dictionary.So when the user gives the input ,the application might show the prediction results.This we will be obtained from web scraping using ml algos like random forest/decision tree with python as backend.The novelty
feature here is the analysis will be provided for the diseases in heatmaps for different combination of inputs by considering the attributes or here let us say symptoms at every stage
from medical research to patient experience and outcomes.
With the rapid advancement of technology and data, the health care domain is most important
in the field in this era.Our objective is to predict diseases based on symptoms by making use of Big Data & analysis and Machine Learning algorithms as we have already mentioned above.As we take enormous amount of data it will be tough to manage so we will be using Big Data Analytics to make it easier for managing the information. This knowledge will give the health care specialists insights that were not available before.

2. **LITERATURE SURVEY**

2.1 Existing problem



2.2 References

1. Effective Heart Disease Prediction using Hybrid Machine Learning Algorithms
Published in IEEE
Link : https://ieeexplore.ieee.org/abstract/document/8740989/
Objective :
Aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classification techniques.
Result :
The prediction models are developed using 13 features and the accuracy is calculated for modeling techniques. The model compares the accuracy, classification error, precision, F-measure, sensitivity and specificity. An enhanced performance level with an accuracy level of 88.7% is achieved through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM).

2. A novel approach for heart disease prediction using strength scores with significant predictors
Published in BMC Part of Springer Nature
Link:
https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01527-5
Objective:
Cardiovascular disease is the leading cause of death in many countries. Physicians

often diagnose cardiovascular disease based on current clinical tests and previous experience of diagnosing patients with similar symptoms. Patients who suffer from heart disease require quick diagnosis, early treatment and constant observations. To address their needs, many data mining approaches have been used in the past in diagnosing and predicting heart diseases. Previous research was also focused on identifying the significant contributing features to heart disease prediction, however, less importance was given to identifying the strength of these features.
This paper is motivated by the gap in the literature, thus proposes an algorithm that measures the strength of the significant features that contribute to heart disease prediction. The study is aimed at predicting heart disease based on the scores of significant features using Weighted Associative Rule Mining.
Result:
A set of important feature scores and rules were identified in diagnosing heart disease and cardiologists were consulted to confirm the validity of these rules. The experiments performed on the UCI open dataset, widely used for heart disease research yielded the highest confidence score of 98% in predicting heart disease.

3. Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction
Published in International Journal of Computer Applications
Link :
https://www.academia.edu/download/79534142/5a18f6653b56138cd5196d20e2f39de18
9e3.pdf
Objective :
Intends to provide a survey of current techniques of knowledge discovery in databases using data mining techniques that are in use in today's medical research particularly in Heart Disease Prediction.
Result :
The outcome of predictive data mining technique on the same dataset reveals that Decision Tree outperforms and some time Bayesian classification is having similar accuracy as that of decision tree but other predictive methods like KNN, Neural Networks, Classification based on clustering are not performing well. The second conclusion is that the accuracy of the Decision Tree and Bayesian Classification further improves after applying a genetic algorithm to reduce the actual data size to get the optimal subset of attributes sufficient for heart disease prediction.

4. **Enhanced Heart Disease Analysis and Prediction System [EHDAPS] Using Data Mining**
Published in Semantic Scholar
Link:
https://www.semanticscholar.org/paper/Improved-Study-of-Heart-Disease-Prediction-System-Dangare-Apte/f4de0213b4a5777ff39d5a94cd574713799ca221

Objective:

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining techniques are used for variety of applications. Data mining techniques have been very effective in designing clinical support systems because of their ability to discover hidden patterns and relationships in clinical data. One of the most important applications of such systems is in diagnosis of heart disease. The main objective of Enhanced Heart Disease Analysis and Prediction System (EHDAPS) is predicting the heart disease using historical heart database. To develop this system, medical terms such as sex, blood pressure, and cholesterol like seventeen input attributes are used. In this paper association among various attributes which are the causative factors of heart diseases are analyzed.

Result:

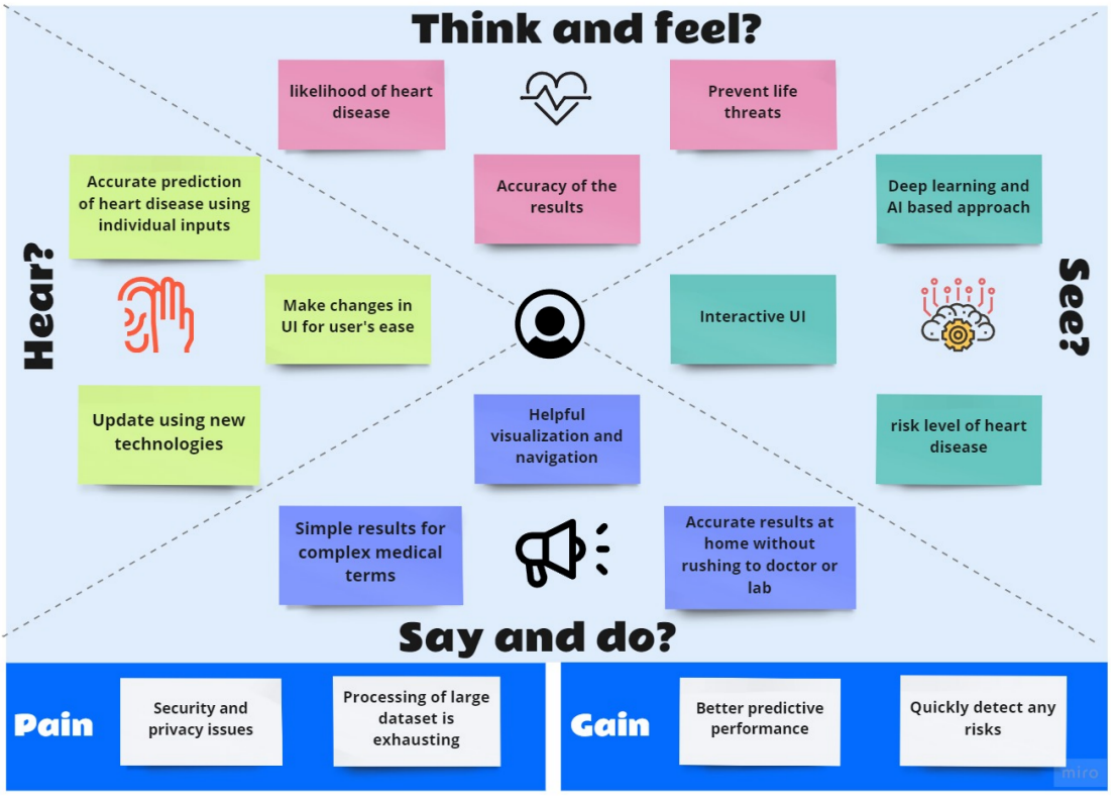
The patient's records are observed before prediction and the factors are grouped as per its severity level. In this system the level of causative factors are categorized using K-Means clustering technique and it distinguishes the risky and non risky factors. Frequent risk factors are mined from the clinical heart database using Apriori algorithm. The risk factors are taken for this study to predict the risk level and find the coordination among the factors that helps the medical people to predict the disease with minimum tests and treatments.

2.3 Problem Statement Definition

Predicting and diagnosing heart disease is the biggest challenge in the medical industry and it is based on factors like physical examination, symptoms and signs of the patient etc. As per this project we will be using data analytics and creating a dashboard with visualizations that will help us predict heart disease.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



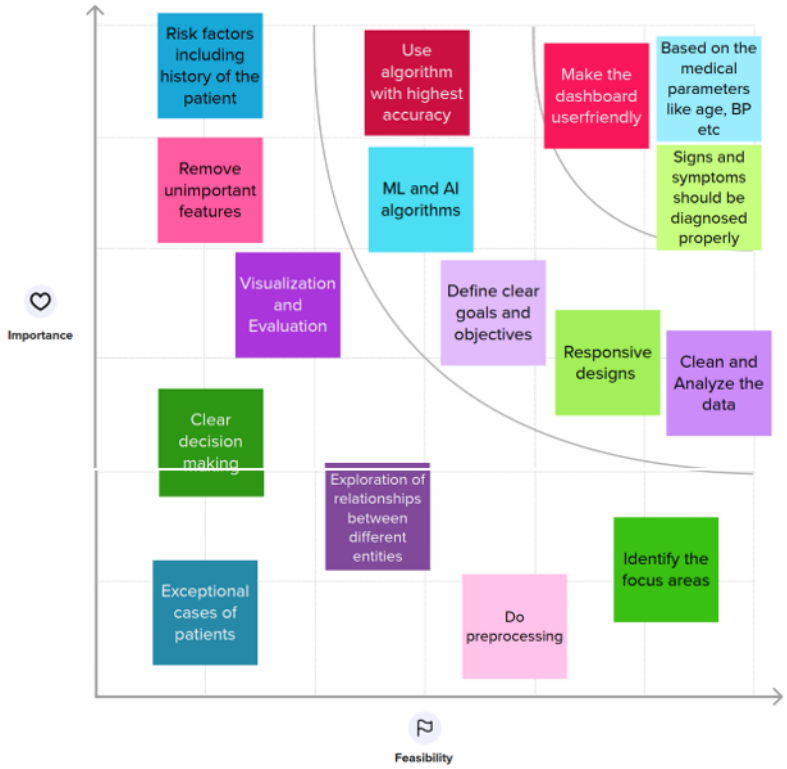
3.2 Ideation & Brainstorming

1. Brainstormed ideas for problem statement

Soumyah K M	Priyanka S	Heevitha R	Ubhanisha Sri C
Based on the medical parameters like age, BP etc	Signs and symptoms should be diagnosed properly	Define clear goals and objectives	Do preprocessing
ML and AI algorithms	Responsive designs	Clean and Analyze the data	Remove unimportant features
Risk factors including history of the patient	Identify the focus areas	Visualization and Evaluation	Make the dashboard userfriendly
Exceptional cases of patients	Clear decision making	Exploration of relationships between different entities	Use algorithm with highest accuracy

2. Grouping up of ideas based on importance, internal and external factors, etc.

3. Prioritize the ideas based on the factors and importance



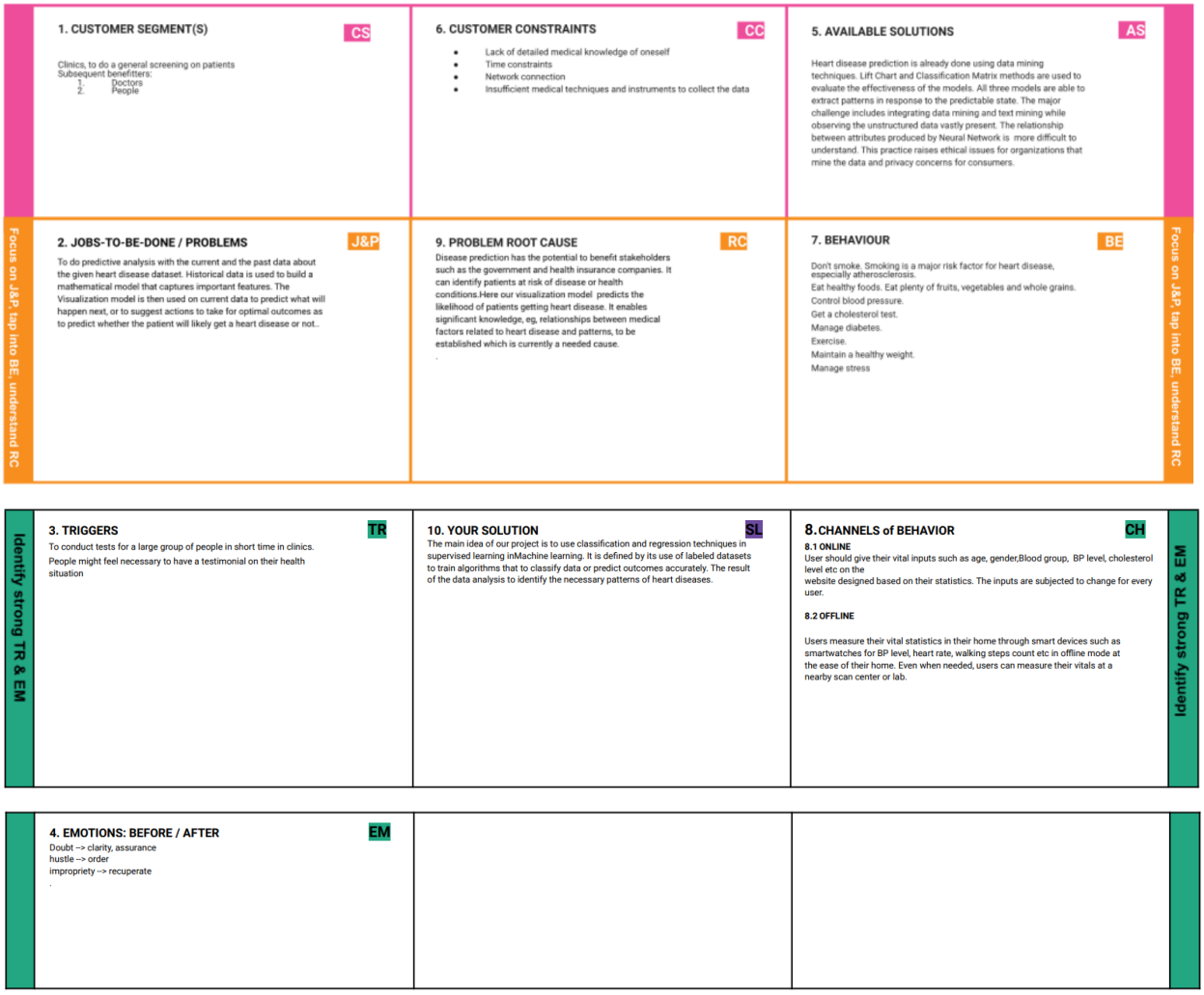
3.3 Proposed Solution

The main idea of our project is to use classification and regression techniques in supervised learning in Machine learning. It is defined by its use of labeled datasets to train algorithms

that to classify data or predict outcomes accurately. The result of the data analysis to identify the necessary patterns for predicting heart diseases.

The proposed system gets inputs directly from the user for parameters such as age, BP level, cholesterol level, smoker history,heart rate etc. These inputs can be tracked by them daily using smart devices. The supervised Machine Learning algorithms are used for learning relationships among input parameters, answer complex queries, better accuracy and provide optimal solutions.

3.4 Problem Solution fit



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
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Training	Dataset upload to train the model
FR-4	Test and output	Based on user trained model, output is displayed

4.2 Non-Functional requirements

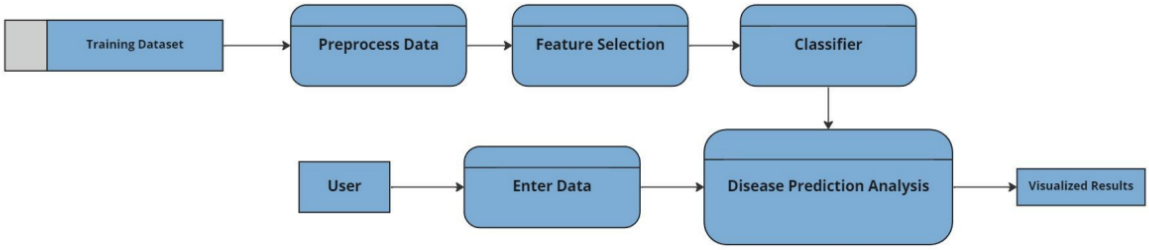
Following are the non-functional requirements of the proposed solution.

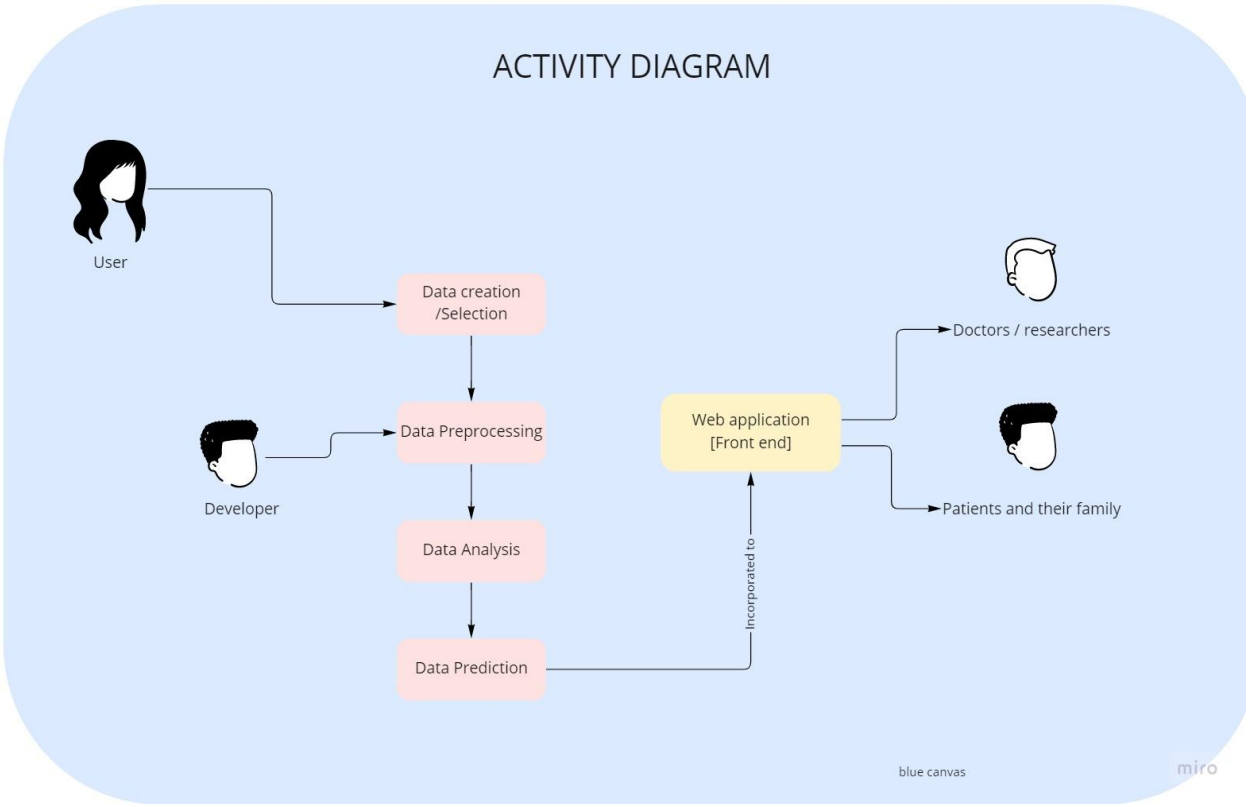
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Wearable devices should be comfortable. All of the pages on a website or mobile application should have the same look and feel. Navigations and UI should be simple and easy to use.
NFR-2	**Security**	The user credentials and their personal information on their body vitals should be kept confidential.
NFR-3	**Reliability**	Our system's cloud component needs to be operational at least 99.5% of the time in order to respond to requests from websites and mobile devices.Fault tolerance should be high.
NFR-4	**Performance**	All sensor data connected to the wearable device should be able to reach the fixed device in less than a second. The wearable gadget should be able to keep up with the rate of sensor data flow.
NFR-5	**Availability**	The system's monitoring and maintenance should be fundamentally focused. It shouldn't be the case that there are too many jobs running on several machines, making it difficult to monitor if they are uninterrupted.
NFR-6	**Scalability**	maintaining multiple users data, sensors accuracy, data transmission rate, increase or decrease of storage etc are monitored.

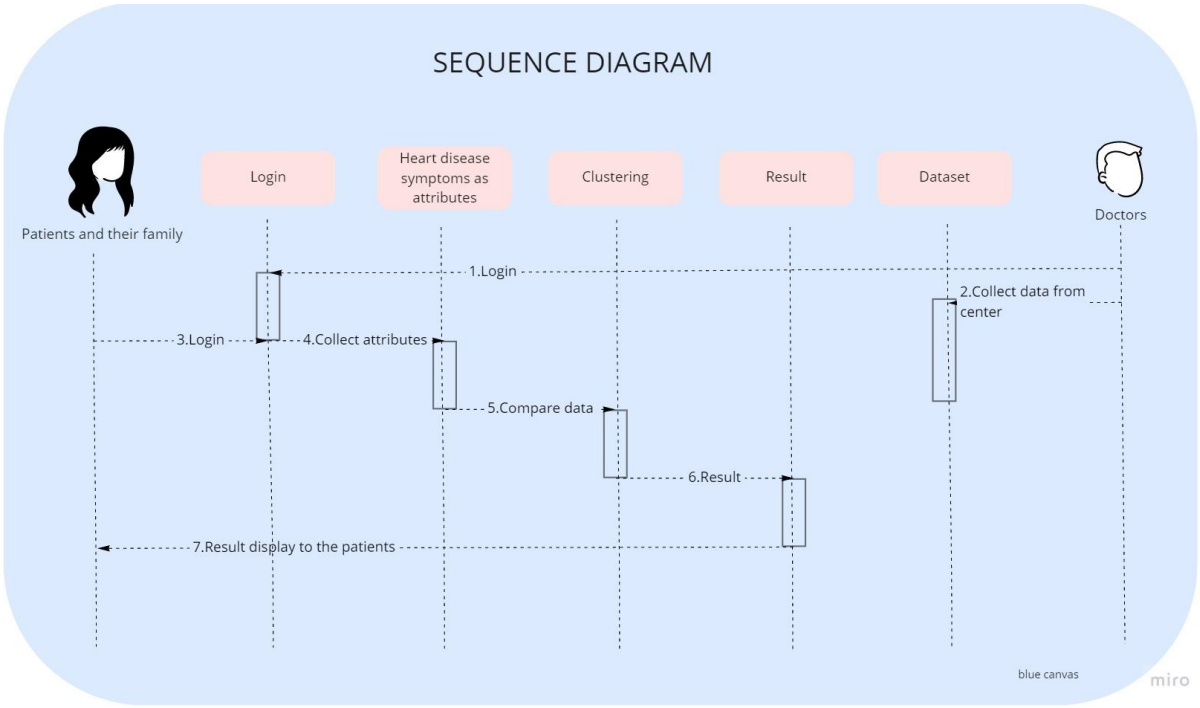
5. PROJECT DESIGN

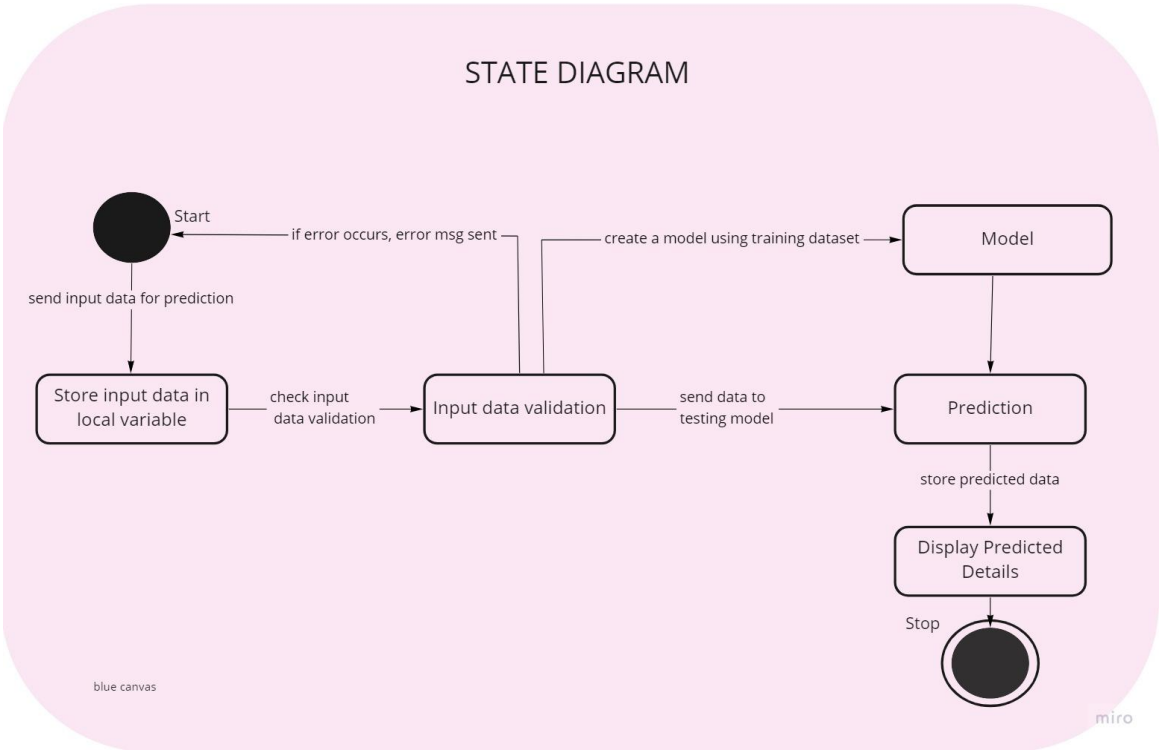
5.1 Data Flow Diagrams

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

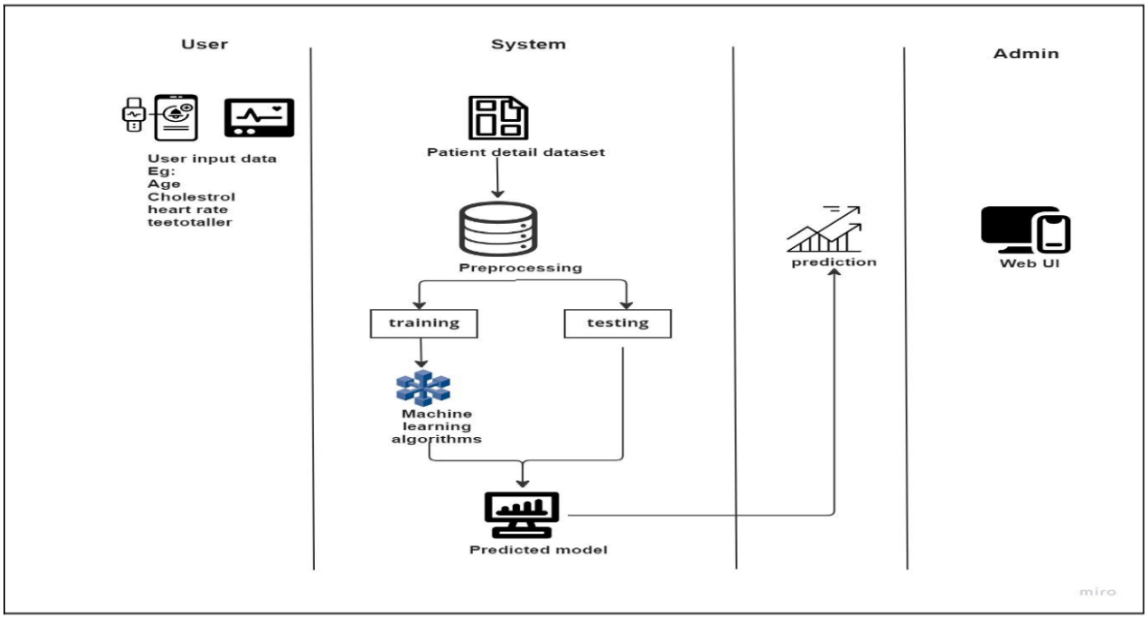

Training Dataset
Preprocess Data
Feature Selection
Classifier
User
Enter Data
Disease Prediction Analysis
Visualized Results


ACTIVITY DIAGRAM
User
Data creation /Selection
Developer
Data Preprocessing
Data Analysis
Data Prediction
Incorporated to
Web application [Front end]
Doctors / researchers
Patients and their family
blue canvas
miro





5.2 Solution & 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-2
Customer (web user)		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 get to the dashboard after signing in	High	Sprint-1
	Dashboard	USN-6	As a user, I can enter my data to the website securely	I can enter data only within the constraints	High	Sprint 4

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-7	As a user, I should give all my personal health vitals as inputs	I can enter data only within the constraints	High	Sprint4
		USN-8	As a user, the navigation of the website should be easier.	I should be able to navigate from all the pages	Medium	Sprint 4
		USN-9	As a user, I can view my results in the dashboard	I can get the visual representation of the results	High	Sprint 4
Administrator	Preprocessing	USN-10	As a administrator, I can add new predictions to training dataset	New records are visible in the updated dataset	Low	Sprint 3
		USN-11	As a administrator, I can remove incomplete records	Updations are visible in the updated dataset	Low	Sprint 3
		USN-12	As a administrator, I can remove unimportant features	Updations are visible in the updated dataset	High	Sprint 3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Soumyah K M
Sprint-1	Login	USN-2	As a user, I will receive confirmation email once I	1	High	Priyanka S

			have registered for the application			
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Ubhanisha Sri C
Sprint-1	Dashboard	USN-4	As a user, I can register for the application through Gmail	2	Medium	Soumyah K M
Sprint-1		USN-5	As a user, I can log into the application by entering email & password	1	High	Ubhanisha Sri C
Sprint 4	Preprocessing	USN-6	As a user, I can enter my data to the website securely	2	High	Priyanka S
Sprint4		USN-7	As a user, I should give all my personal health vitals as inputs	2	High	Heevitha R
Sprint 4		USN-8	As a user, the navigation of the website should be easier.	1	Medium	Soumyah K M
Sprint 4		USN-9	As a user, I can view my results in the dashboard	2	High	Heevitha R

Sprint 3		USN-10	As a administrator, I can add new predictions to training dataset	1	Low	Ubhanisha Sri C
Sprint 3		USN-11	As a administrator, I can remove incomplete records	1	Low	Soumyah K M
Sprint 3		USN-12	As a administrator, I can remove unimportant features	2	High	Priyanka S
Sprint 4	Visualization	USN-13	Having a view with geographic data	2	High	Heevitha R
Sprint 4		USN-14	Analysis of data with IBM cognos analytics	2	High	Priyanka S

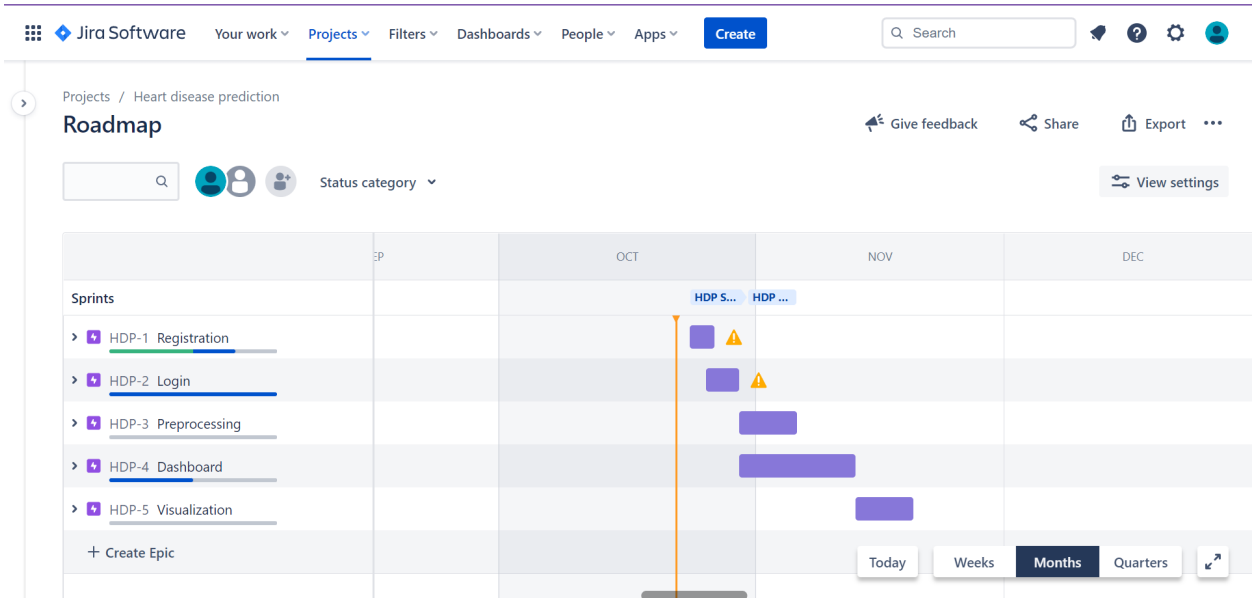
6.2 Sprint Delivery Schedule

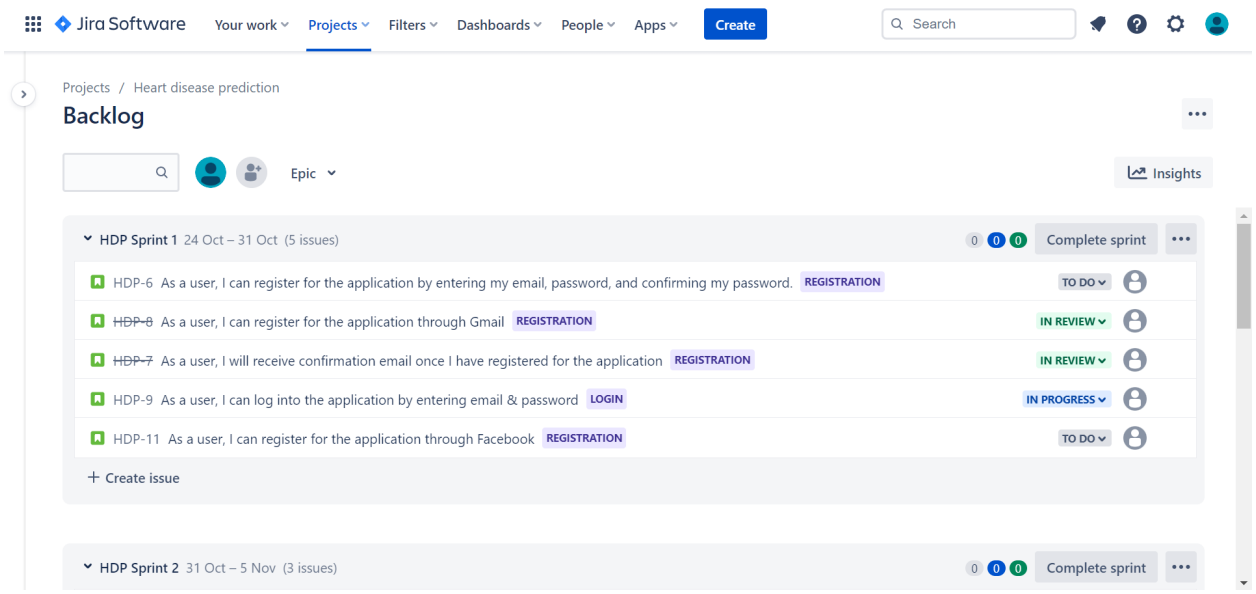
Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022

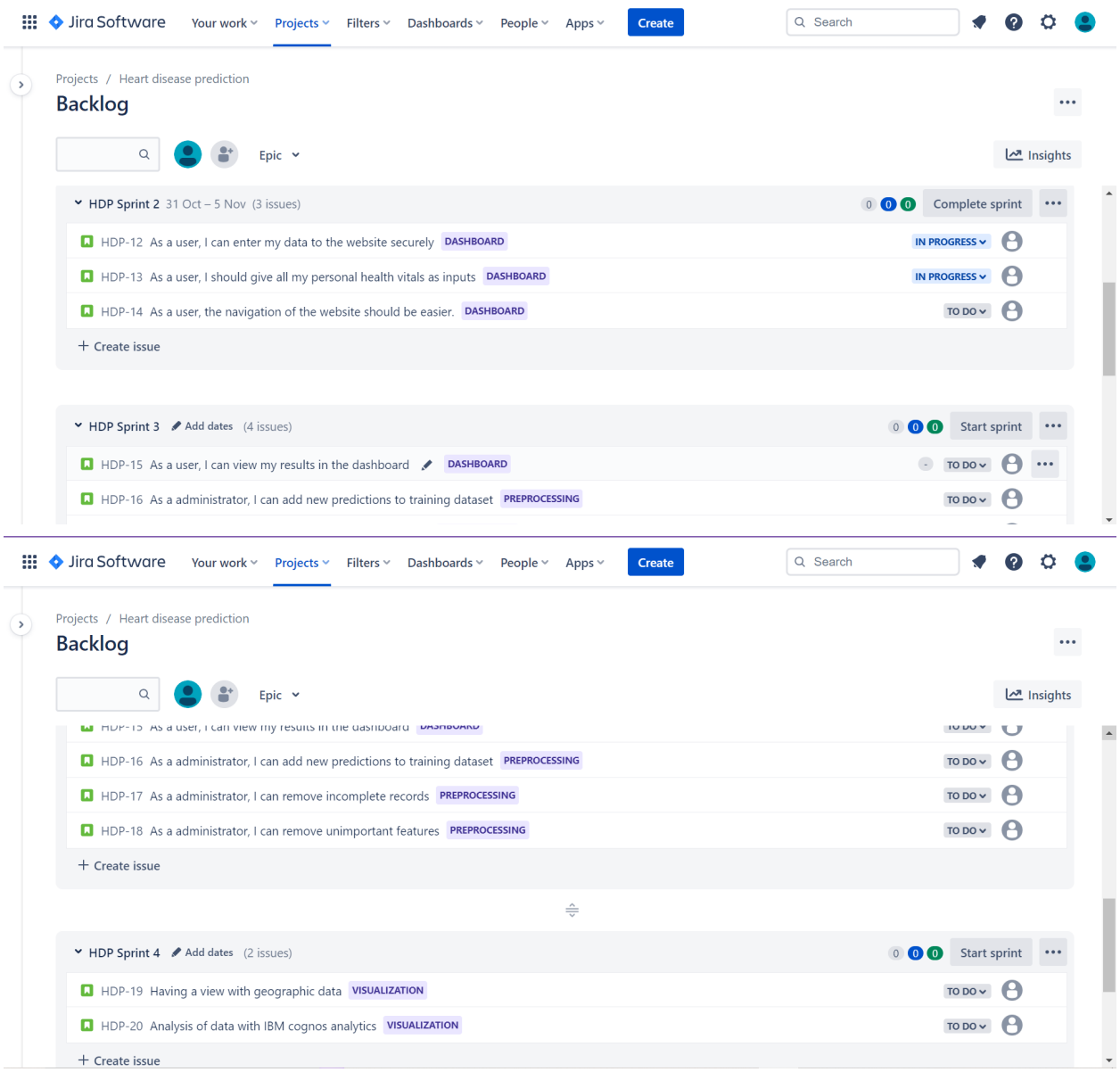
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

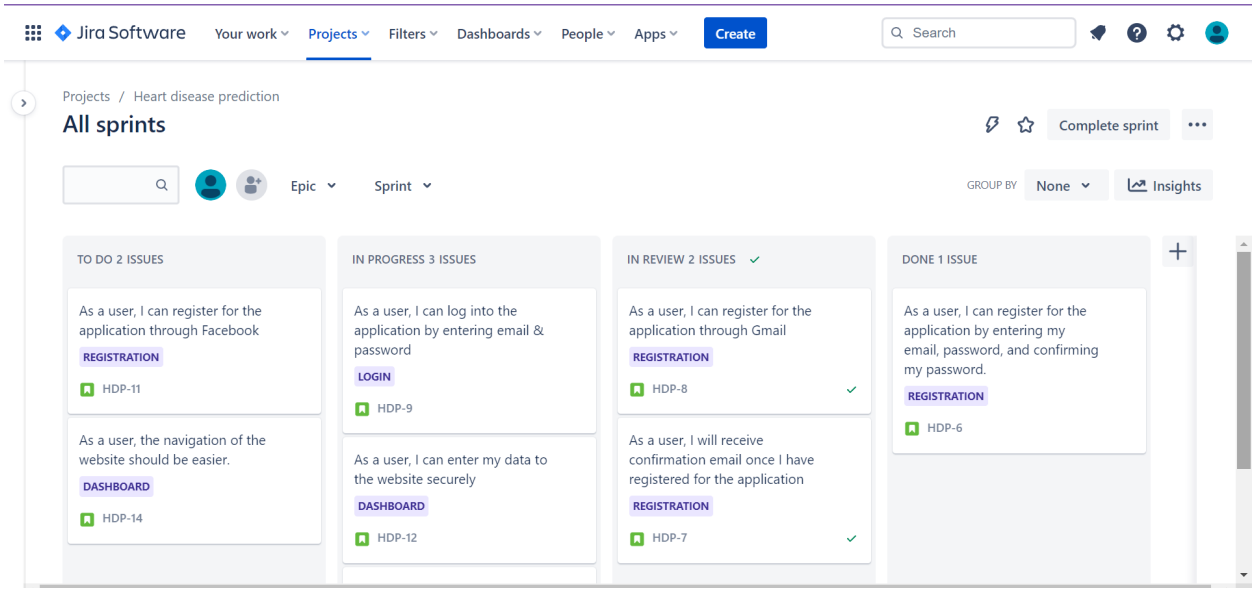
The tool used here is: JIRA Software (or Atlassian)

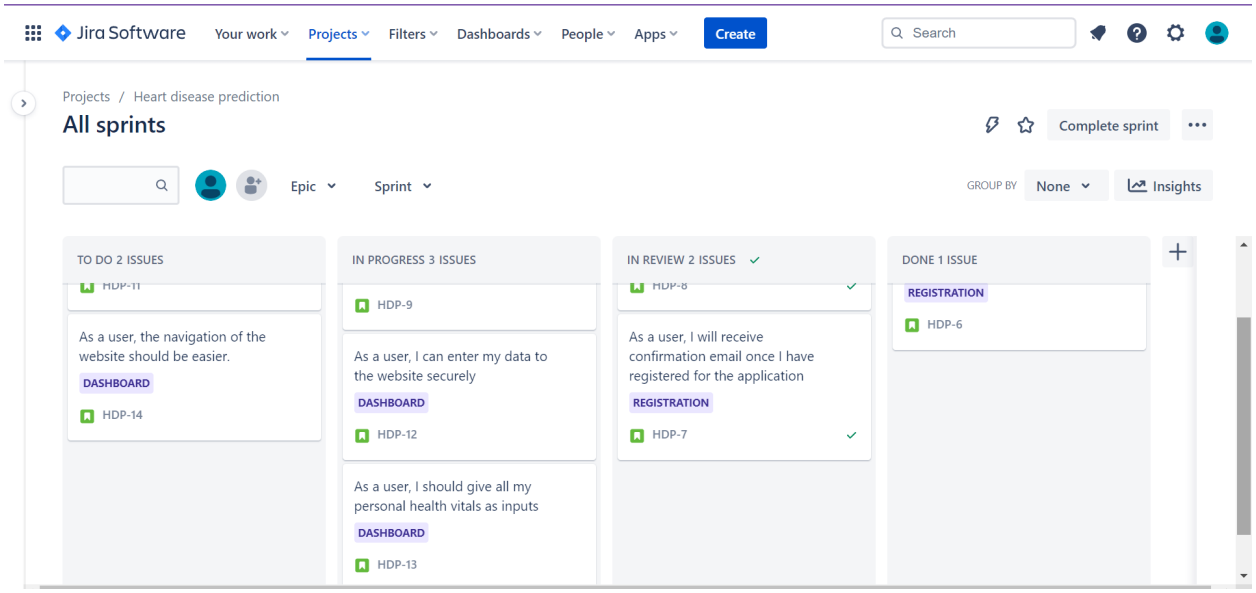

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
Roadmap
Give feedback
Share
Export
Status category
View settings
OCT
NOV
DEC
Sprints
HDP S...
HDP ...
HDP-1 Registration
HDP-2 Login
HDP-3 Preprocessing
HDP-4 Dashboard
HDP-5 Visualization
Create Epic
Today
Weeks
Months
Quarters


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
Backlog
Epic
Insights
HDP Sprint 1 24 Oct – 31 Oct (5 issues)
Complete sprint
HDP-6 As a user, I can register for the application by entering my email, password, and confirming my password. REGISTRATION TO DO
HDP-8 As a user, I can register for the application through Gmail REGISTRATION IN REVIEW
HDP-7 As a user, I will receive confirmation email once I have registered for the application REGISTRATION IN REVIEW
HDP-9 As a user, I can log into the application by entering email & password LOGIN IN PROGRESS
HDP-11 As a user, I can register for the application through Facebook REGISTRATION TO DO
Create issue
HDP Sprint 2 31 Oct – 5 Nov (3 issues)
Complete sprint


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
Backlog
Epic
Insights
HDP Sprint 2 31 Oct – 5 Nov (3 issues)
Complete sprint
HDP-12 As a user, I can enter my data to the website securely DASHBOARD IN PROGRESS
HDP-13 As a user, I should give all my personal health vitals as inputs DASHBOARD IN PROGRESS
HDP-14 As a user, the navigation of the website should be easier. DASHBOARD TO DO
Create issue
HDP Sprint 3 Add dates (4 issues)
Start sprint
HDP-15 As a user, I can view my results in the dashboard DASHBOARD TO DO
HDP-16 As a administrator, I can add new predictions to training dataset PREPROCESSING TO DO

Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
Backlog
Epic
Insights
HDP-16 As a administrator, I can add new predictions to training dataset PREPROCESSING TO DO
HDP-17 As a administrator, I can remove incomplete records PREPROCESSING TO DO
HDP-18 As a administrator, I can remove unimportant features PREPROCESSING TO DO
Create issue
HDP Sprint 4 Add dates (2 issues)
Start sprint
HDP-19 Having a view with geographic data VISUALIZATION TO DO
HDP-20 Analysis of data with IBM cognos analytics VISUALIZATION TO DO
Create issue


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 2 ISSUES
As a user, I can register for the application through Facebook
REGISTRATION
HDP-11
As a user, the navigation of the website should be easier.
DASHBOARD
HDP-14
IN PROGRESS 3 ISSUES
As a user, I can log into the application by entering email & password
LOGIN
HDP-9
As a user, I can enter my data to the website securely
DASHBOARD
HDP-12
IN REVIEW 2 ISSUES
As a user, I can register for the application through Gmail
REGISTRATION
HDP-8
As a user, I will receive confirmation email once I have registered for the application
REGISTRATION
HDP-7
DONE 1 ISSUE
As a user, I can register for the application by entering my email, password, and confirming my password.
REGISTRATION
HDP-6


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Projects / Heart disease prediction
All sprints
Complete sprint
Epic
Sprint
GROUP BY
None
Insights
TO DO 2 ISSUES
As a user, the navigation of the website should be easier.
DASHBOARD
HDP-14
IN PROGRESS 3 ISSUES
HDP-9
As a user, I can enter my data to the website securely
DASHBOARD
HDP-12
As a user, I should give all my personal health vitals as inputs
DASHBOARD
HDP-13
IN REVIEW 2 ISSUES
As a user, I will receive confirmation email once I have registered for the application
REGISTRATION
HDP-7
DONE 1 ISSUE
REGISTRATION
HDP-6

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

1a) Used Firebase for Authentication and storage

```
import pyrebase
firebaseConfig = {
  'apiKey': "AIzaSyBW9IPsQGqk9qrXsgyL8TZpnQ4MacWgc70",
  'authDomain': "test-firestore-streamlit-13160.firebaseapp.com",
  'projectId': "test-firestore-streamlit-13160",
'databaseURL':
"https://console.firebase.google.com/u/0/project/test-firestore-streamlit-13160/database/test-firestore-streamlit-13160-default-rtdb/data/~2F",
  'storageBucket': "test-firestore-streamlit-13160.appspot.com",
  'messagingSenderId': "703800935011",
  'appId': "1:703800935011:web:60fd49fa5c4d09eb002e56",
  'measurementId': "G-GR57GQVJC3"
 }

#firebase authentication
firebase = pyrebase.initialize_app(firebaseConfig)
auth = firebase.auth()
#database
db = firebase.database()
storage = firebase.storage()
```

1b)Validated every field of login and registration forms

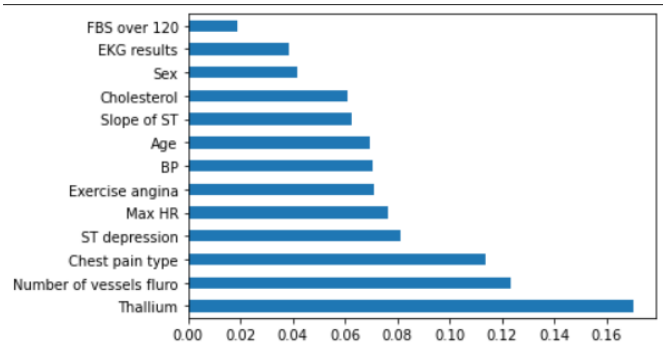
```
from email_validator import validate_email, EmailNotValidError
choice = st.sidebar.selectbox('login/Signup', ['Login', 'Sign up'])
email = st.sidebar.text_input("Please enter your email address")
password = st.sidebar.text_input("Please enter your password")
if choice == 'Sign up':
  handle = st.sidebar.text_input("Please input your username", value = 'Default')
  submit = st.sidebar.button("Create my Account")
  if submit:
    try:
      if email=='' or password=='':
        st.error("Please fill the empty email or password")
      else:
        try:
          validation = validate_email(email)
          user = auth.create_user_with_email_and_password(email, password)
```

```
                st.success("Your account is created successfully")
                st.balloons()
                st.title("Welcome "+handle+" !!")
                st.header("Login to view dashboard")
            except:
                st.error("enter valid email")
        except:
            st.error("email already exists")
if choice == 'Login':
    login = st.sidebar.button('Login')
    if login:
        try:
            user = auth.sign_in_with_email_and_password(email,password)
```

7.2 Feature 2

2a)Dataset is preprocessed and at least three unimportant features are removed as they have larger margin from other features.

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X ,y)
print(model.feature_importances_) #use inbuilt class feature_importances of tree based
    classifiers
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(13).plot(kind='barh')
plt.show()
```




```
hdp.drop(['FBS over 120', 'EKG results', 'Sex'], axis = 1, inplace = True)
```

2b)Random forest algorithm is applied on the dataset, and achieved 85.158% accuracy

```
a = hdp.drop('Heart Disease',axis=1)
b = hdp['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(a,b,test_size=0.2)
oversample = RandomOverSampler(sampling_strategy='minority')
X_over, y_over = oversample.fit_resample(X_train,y_train)
rf = RandomForestClassifier()
rf.fit(X_over,y_over)
preds = rf.predict(X_test)
print(accuracy_score(y_test,preds))
```

```
0.8518518518518519
```

7.3 Database Schema

Database schema is applicable only for users logins and the columns included are email, password, username.

8. TESTING

8.1 Test Cases

8.1.1 User should able to choose from login or signup

8.1.2 The UI elements should correspond to the appropriate fields such as email, password and username that gets stored in the database

8.1.3 The login page is valid only if the user has already registered

8.1.4 The registered user should login to view the dashboard

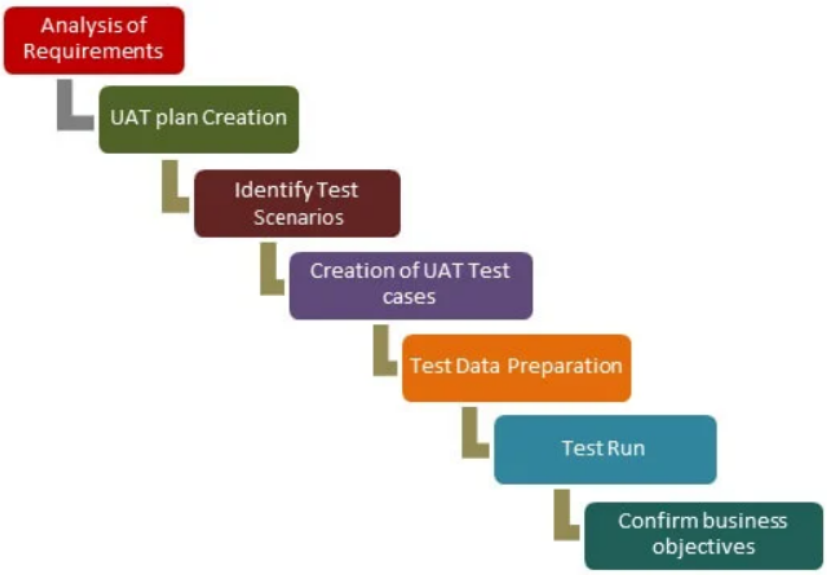
8.1.5 The credentials should match for both login and register

8.1.6 The email account should be valid

8.2 User Acceptance Testing

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is a kind of black box testing where two or more end-users will be involved.



UAT Process

UAT Tools

There are several tools in the market used for User acceptance testing and we have tried the subsequent tools.

Fitnesse tool: It is a java tool used as a testing engine. It is easy to create tests and record results in a table. Users of the tool enter the formatted input and tests are created automatically. The tests are then executed and the output is returned back to the user.

Watir : It is a toolkit used to automate browser-based tests during User acceptance testing. Ruby is the programming language used for inter-process communication between ruby and Internet Explorer.

9. RESULTS

9.1 Performance Metrics

The accuracy for individual algorithms has to be measured and whichever algorithm is giving the best accuracy,that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered.

```
print ('Classification Report : ')
print (classification_report(y_test, y_pred))
```

```
Classification Report :
              precision    recall  f1-score   support

     Absence       0.86      0.91      0.88        33
    Presence       0.84      0.76      0.80        21

    accuracy                           0.85        54
   macro avg       0.85      0.84      0.84        54
weighted avg       0.85      0.85      0.85        54
```

10. ADVANTAGES & DISADVANTAGES

Advantages

1. Increased accuracy for effective heart disease diagnosis because that somewhat exactly tells us if we are likely to get a heart disease or not in the future. This step is essential as the user is completely dependent on this result and it should live up to their expectations.

2. Handles roughest (enormous) amount of data using random forest algorithm and feature selection to cut down the unnecessary features in the dataset given. Only the most important ones are included as that improves the accuracy of the result and a one that can be relied upon.

3. Reduce the time complexity of doctors as they need not spend a lot of time diagnosing a patient as that can be complex at times.

4. Cost effective for patients.They need not spend tons of money on various kinds of tests. They can just perform one sample test and benefit from it. The world is changing and so is the advancement of technology.

Disadvantages:

1. Data mining techniques do not help to provide effective decision making since it cannot handle enormous datasets for patient records.
2. Slower for real-time predictions
3. Fast to train but slower to create predictions
4. Large amount of storage needed for real time

11. CONCLUSION

The overall aim is to define various machine learning algorithms and techniques useful in effective heart disease prediction. Efficient and accurate prediction of whether a person is likely

to have a heart disease with a lesser number of attributes and tests is the goal of this research. The data were pre-processed and then used in the model. Random Forest with 85.18% and SVM with 78.69% are the most efficient algorithms. However, K-Nearest Neighbor performed with the worst accuracy with 57.83%. We can further expand this research incorporating other machine learning techniques such as Naïve Bayes, Decision Tree,XGBoost on the UCI dataset.

Considering the limitations of this study, there is a need to implement more complex and combination of models to get higher accuracy for early prediction of heart disease.

12. FUTURE SCOPE

Most of the papers on prediction showed that hybrid techniques outperform a single classification technique in terms of accuracy. They have concluded that neural networks are an efficient technique for prediction. When the system is trained properly along with genetic algorithms, the system shows very promising results. This method can also be used to select the proper treatment methods for a patient in future, instead of just predicting the chances of developing a heart disease among the patients.

Eventually, an intelligent system may be developed that can lead to selection of proper treatment methods for a patient diagnosed with heart disease. A lot of work has been done already in making models that can predict whether a patient is likely to develop heart disease or not. There are several treatment methods for a patient once diagnosed with a particular form of heart disease. Machine learning algorithms along with Data mining can be of very good help in deciding the line of treatment to be followed by extracting knowledge from such suitable databases.

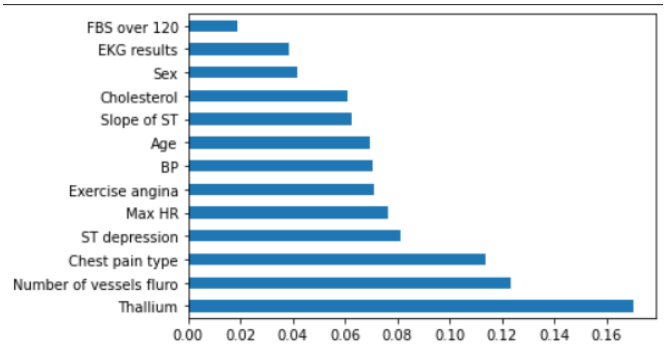
13. APPENDIX

Source Code

Code to generate pickle file

```
import pandas as pd
hdp = pd.read_csv('/content/Heart_Disease_Prediction.csv')# Dropping null values
hdp = hdp.dropna()
hdp.head()
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import RandomOverSampler
from sklearn.ensemble import RandomForestClassifier0
```

```
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score
X = hdp.drop('Heart Disease',axis=1)
y = hdp['Heart Disease']
import numpy as np
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X ,y)
print(model.feature_importances_) #use inbuilt class feature_importances of tree based
    classifiers
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(13).plot(kind='barh')
plt.show()
```




```
hdp.drop(['FBS over 120', 'EKG results', 'Sex'], axis = 1, inplace = True)
a = hdp.drop('Heart Disease',axis=1)
b = hdp['Heart Disease']
X_train, X_test, y_train, y_test = train_test_split(a,b,test_size=0.2)
oversample = RandomOverSampler(sampling_strategy='minority')
X_over, y_over = oversample.fit_resample(X_train,y_train)
rf = RandomForestClassifier()
rf.fit(X_over,y_over)
preds = rf.predict(X_test)
print(accuracy_score(y_test,preds))
```

```
0.8518518518518519
```

```
import joblib
joblib.dump(rf, 'hdp_model.pkl')
```

```
['hdp_model.pkl']
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import log_loss
import matplotlib.pyplot as plt

sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)

svc = SVC(kernel='linear', C=10.0, random_state=1)
svc.fit(X_train, y_train)

y_pred = svc.predict(X_test)

conf_matrix = confusion_matrix(y_true=y_test, y_pred=y_pred)

fig, ax = plt.subplots(figsize=(5, 5))
ax.matshow(conf_matrix, cmap=plt.cm.Oranges, alpha=0.3)
for i in range(conf_matrix.shape[0]):
    for j in range(conf_matrix.shape[1]):
```

```
        ax.text(x=j, y=i,s=conf_matrix[i, j], va='center', ha='center', size='xx-large')

plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
print ('Accuracy Score is',accuracy_score(y_test, y_pred))
print ('Classification Report : ')
print (classification_report(y_test, y_pred))
```

app.py

```
#imports

import pyrebase
import streamlit as st
from datetime import datetime
import joblib
import pandas as pd
from collections.abc import Mapping
from email_validator import validate_email, EmailNotValidError

#configuration key

firebaseConfig = {
  'apiKey': "AIzaSyBW9IPsQGqk9qrXsgyL8TZpnQ4MacWgc70",
  'authDomain': "test-firestore-streamlit-13160.firebaseapp.com",
  'projectId': "test-firestore-streamlit-13160",
'databaseURL':'https://console.firebase.google.com/u/0/project/test-firestore-streamlit-13160
      /database/test-firestore-streamlit-13160-default-rtdb/data/~2F",
  'storageBucket': "test-firestore-streamlit-13160.appspot.com",
  'messagingSenderId': "703800935011",
  'appId': "1:703800935011:web:60fd49fa5c4d09eb002e56",
  'measurementId': "G-GR57GQVJC3"
```

```
 }

#firebase authentication

firebase = pyrebase.initialize_app(firebaseConfig)
auth = firebase.auth()

#database

db = firebase.database()
storage = firebase.storage()

st.sidebar.title("Heart Disease Prediction")

#authentication
choice = st.sidebar.selectbox('login/Signup', ['Login', 'Sign up'])

email = st.sidebar.text_input("Please enter your email address")
password = st.sidebar.text_input("Please enter your password")



if choice == 'Sign up':
   handle = st.sidebar.text_input("Please input your username", value = 'Default')
   submit = st.sidebar.button("Create my Account")

   if submit:
      try:
         if email=='' or password=='':
            st.error("Please fill the empty email or password")
         else:
            try:
               validation = validate_email(email)
               user = auth.create_user_with_email_and_password(email, password)
```

```
                st.success("Your account is created successfully")
                st.balloons()
                st.title("Welcome "+handle+" !!")
                st.header("Login to view dashboard")
            except:
                st.error("enter valid email")

        except:
            st.error("email already exists")

if choice == 'Login':
    login = st.sidebar.button('Login')
    if login:
        try:
            user = auth.sign_in_with_email_and_password(email,password)



            st.write('<style>div.row-widget.stRadio > div{flex-direction:row;}</style>',
        unsafe_allow_html=True)

            st.title("Heart Disease Prediction")
            col1, col2, col3 = st.columns(3)
            Age = col1.number_input("Enter your age")
            chest_pain_type = col2.selectbox("Enter chest pain type?",["1","2","3","4"])
            BP = col3.number_input("Enter BP level")
            chol = col1.number_input("Enter cholestrol level")
            maxhr = col2.number_input("Enter max heartrate")
            exe_angina = col3.selectbox("Do you have exercise angina?",["Yes", "No"])
            ST_depr = col1.number_input("Enter ST depression value")
            ST_slope = col2.selectbox("Enter ST slope value",["1","2","3"])
            no_of_vess = col3.selectbox("No of vessels of fluro",["0","1","2","3"])
            thall = col2.number_input("Enter thallium level")
```

```
    #st.button('Predict')
    model = joblib.load('hdp_model.pkl')

    df_pred =
pd.DataFrame([[Age,chest_pain_type,BP,chol,maxhr,exe_angina,ST_depr,ST_slope,no_of_vess,
thall]],columns= ['Age','Chest pain type','BP','Cholesterol','Max HR','Exercise angina','ST
depression','Slope of ST','Number of vessels fluro','Thallium'])

    df_pred['Exercise angina'] = df_pred['Exercise angina'].apply(lambda x: 1 if x == 'Yes' else
0)

    model = joblib.load('hdp_model.pkl')
    prediction = model.predict(df_pred)
    if st.button('Predict', key = 0):
        if (df_pred['Age']>=120).bool() | (df_pred['Age']<1).bool()  |
(df_pred['Cholesterol']<20).bool() | (df_pred['Cholesterol']>600).bool() |
(df_pred['BP']>180).bool() | (df_pred['BP']>120).bool() | (df_pred['Max HR']>450).bool() |
(df_pred['Max HR']<=0).bool() | (df_pred['Thallium']>8).bool() | (df_pred['Thallium']<0
).bool() | (df_pred['ST depression']>6).bool() | (df_pred['ST depression']<0).bool():
            if (df_pred['Age']>=120).bool() | (df_pred['Age']<1).bool():
                st.write('<p class="big-font">Invalid AGE.. Please fill in appropriate
data.</p>',unsafe_allow_html=True)
            if (df_pred['Cholesterol']<20).bool() | (df_pred['Cholesterol']>600).bool():
                st.write('<p class="big-font">Extremely low or high cholestrol level.. Please fill in
appropriate data.</p>',unsafe_allow_html=True)
            if (df_pred['BP']>180).bool() or (df_pred['BP']>120).bool():
                st.write('<p class="big-font">Blood pressure extremely high.. Contact medical
professional immediately!</p>',unsafe_allow_html=True)
            if (df_pred['Max HR']>2000).bool() | (df_pred['Max HR']<=0).bool():
                st.write('<p class="big-font">Invalid max heartrate.. Please fill in appropriate
data.</p>',unsafe_allow_html=True)
            if (df_pred['Thallium']>7).bool() | (df_pred['Thallium']<0).bool():
```

```
            st.write('<p class="big-font">Invalid Thallium value.. Please fill in appropriate
data.</p>',unsafe_allow_html=True)
          if (df_pred['ST depression']>6).bool() | (df_pred['ST depression']<0).bool():
            st.write('<p class="big-font">Invalid ST depression value.. Please fill in appropriate
data.</p>',unsafe_allow_html=True)
       elif(prediction[0]=='Absence'):
        st.write('<p class="big-font">Result, Heart Disease
Absent</p>',unsafe_allow_html=True)
       else:
    st.write('<p class="big-font">Result, Heart Disease
Present</p>',unsafe_allow_html=True)

 except:
    st.error("incorrect email/password")
```

GitHub & Project Demo Link

Github link : https://github.com/IBM-EPBL/IBM-Project-33544-1660222473

Project Demo Link : https://www.youtube.com/watch?v=Kz3ZRrSXB5A