# VISUALIZING AND PREDICTION OF HEART DISEASES WITH AN INTERACTIVE DASH BOARD

**NALAIYA THIRAN PROJECT REPORT 2022**

**SUBMITTED BY**


| | |
|---|---|
| **VISHAL RAMAPRABHU** | **193002121** |
| **THAYALAN R** | **193002112** |
| **YASH TATED** | **193002125** |
| **YESWANTHRAJ S P** | **193002312** |


**TEAM ID: PNT2022TMID52997**

# Project Report Format

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases
    2. User Acceptance Testing

9. **RESULTS**
    1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

# 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 risk of having a heart attack or stock.This project aims to create an interactive dashboard using IBM cosign Tool and dataset to predict which patients are likely to suffer from a heart disease soon using the features given.

## 1.2 Purpose

Heart disease (HD) is a major cause of mortality in modern society. Medical diagnosis is a crucial 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 analyse 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.

# 2. LITERATURE SURVEY

## 2.1 Existing Problem

Healthcare industries generate enormous amount of data, so called big data that accommodates hidden knowledge or pattern for decision making. The huge volume of data is used to make decision which is more accurate than intuition. Exploratory Data Analysis (EDA) detects mistakes, finds appropriate data, checks assumptions and determines the correlation among the explanatory variables. In the context, EDA is considered as analysing data that excludes inferences and statistical

modelling. Analytics is an essential technique for any profession as it forecast the future and hidden pattern. Data analytics is considered as a cost effective technology in the recent past and it plays an essential role in healthcare which includes new research findings, emergency situations and outbreaks of disease. The use of analytics in healthcare improves care by facilitating preventive care and EDAis a vital step while analysing data

## 2.2 References

**“Heart Disease Prediction using Exploratory Data Analysis”**
**R. Indrakumari, T.Poongodi, SoumyaRanjan Jena**

In this paper, the risk factors that causes heart disease is considered and predicted using K-means algorithm and the analysis is carried out using a publicly available data for heart disease. The dataset holds 209 records with 8 attributes such as age, chest pain type, blood pressure, blood glucose level,ECG in rest, heart rate and four types of chest pain. To predict the heart disease,K- means clustering algorithm is used along with data analytics and visualization tool. The paper discusses the pre-processing methods,classifier performances and evaluation metrics.In the result section, the visualized data shows that the prediction is accurate.

**Prediction of heart disease at early stage using data mining and big data analytics: A survey N. K. Salma Banu, Suma Swamy**

Several studies have been carried out for developing prediction model using individual technique and also by combining two or more techniques. This paper provides a quick and easy review and understanding of available prediction models using data mining from 2004 to 2016. The comparison shows the accuracy level of each model given by different researchers. Into practice.

## 2.3 Problem Statement definition

### Who does the problem affect?

People with unhealthy lifestyles, stress, depression, age above 40 and when their ancestors got heart disease(since heart disease is hereditary).

### When does the issue occur?

The issue occurs for people with unhealthy lifestyles and age above 40. Where is the issue occurring? The issue is originating from an unhealthy lifestyle. It mostly occurs in the blood valves of the heart.

### What would happen if we didn't solve the problem?

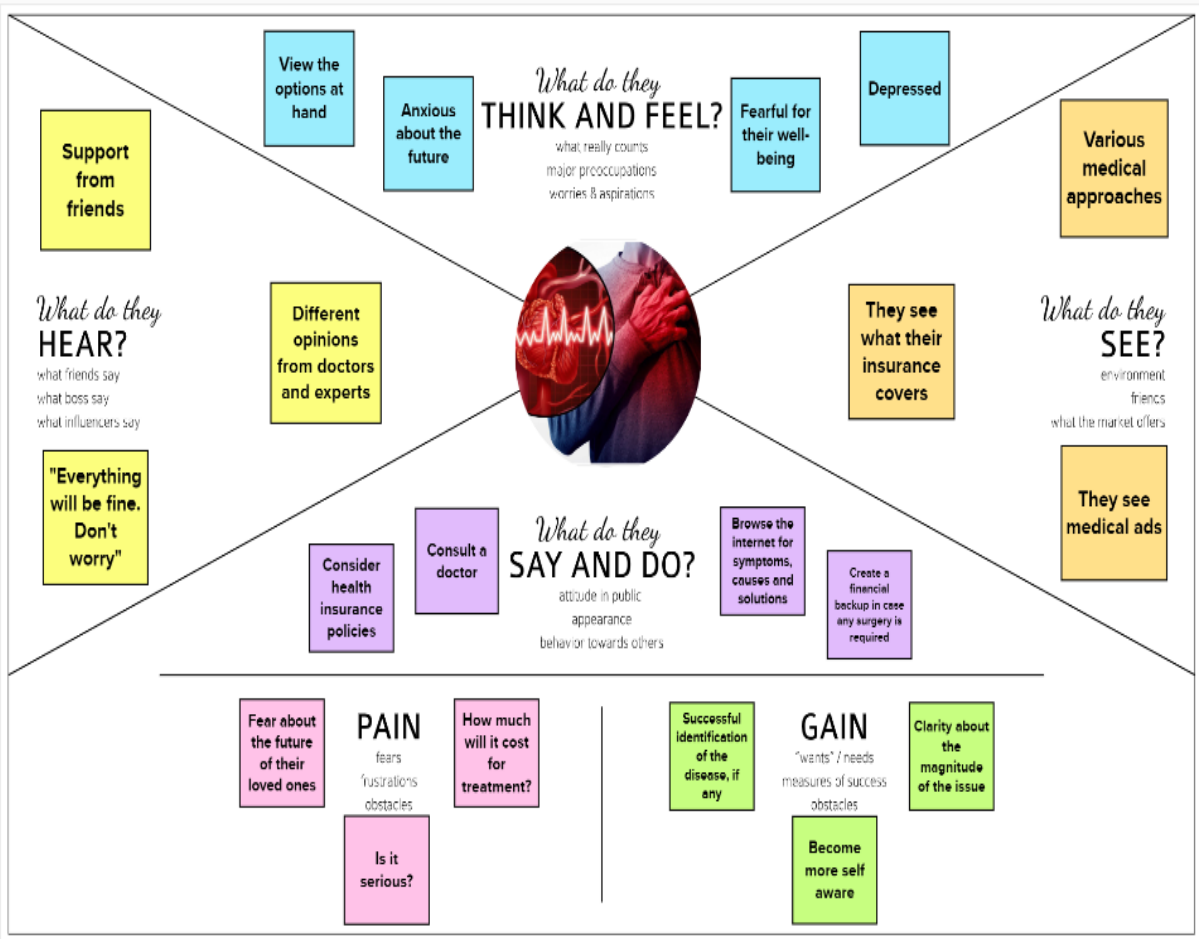
If we don't solve the problem, many people will die at a young age. The death rate due to heart diseasewill explode.

### Why is it important to fix the problem?

We should predict the problem before giving treatment to the patients. As the problem is predicted early, We can solve it easily and early.

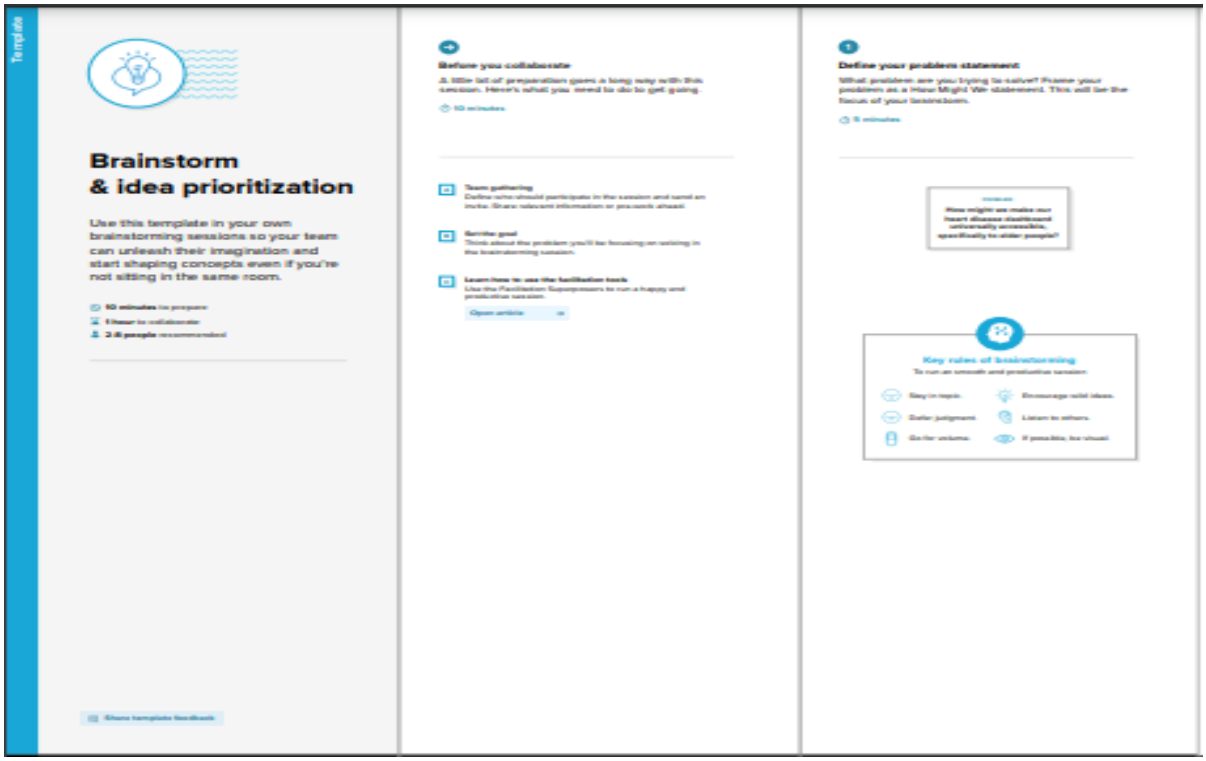
# 3. IDEATION AND PROPOSED SOLUTION

## 3.1 Empathy Map Canvas

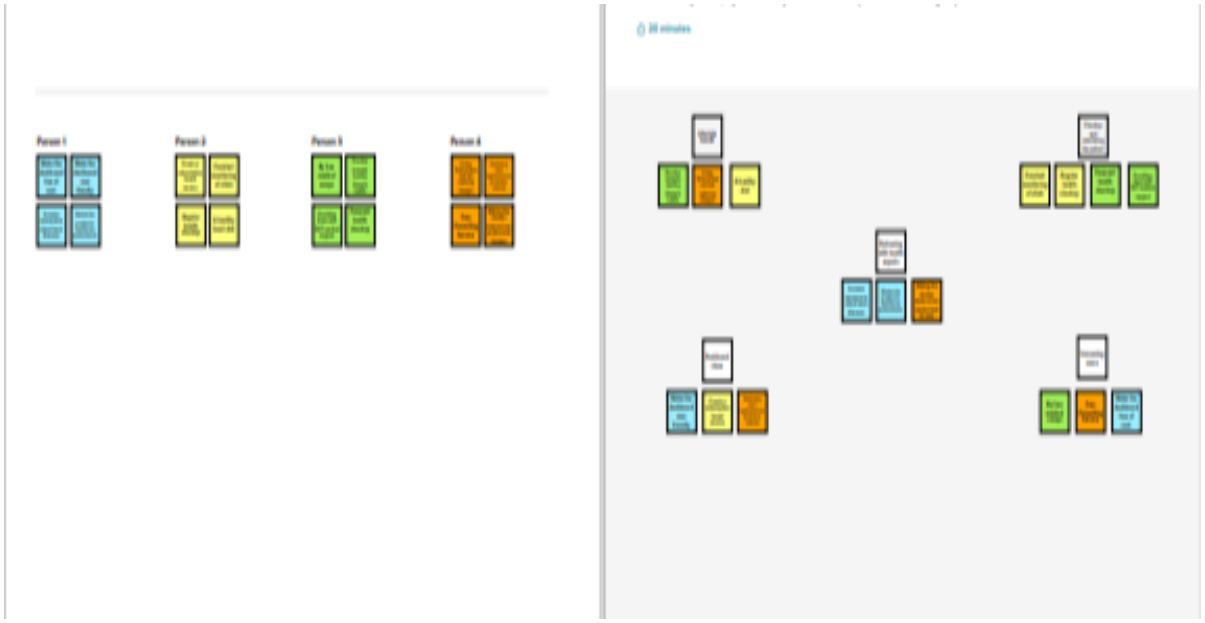


## 3.2 Ideation and Brainstorming

Step-1: Team Gathering, Collaboration and Select the Problem Statement




Step-2: Brainstorm, Idea Listing and Grouping



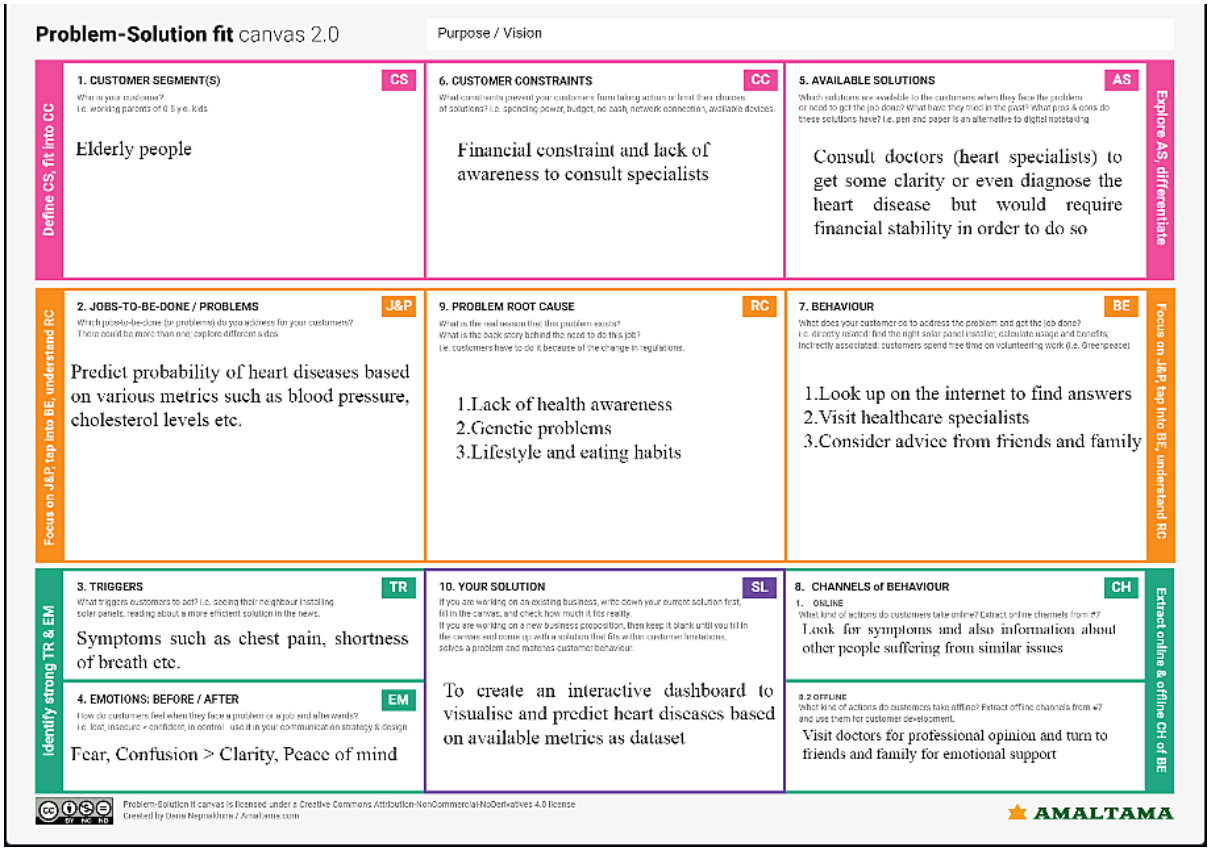
Step-3: Idea Prioritization

## 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 | To use an existing dataset to predict which patients are most likely to suffer from a heart disease in the near future using the details given such as BP,Cholesterol levels, EKG results etc. and visualising the data in the form of graphs and charts |

| | | |
|---|---|---|
| 3. | Novelty / Uniqueness | To find the heart diseases in the early stages by predicting all possible outcomes in such a way by visualising the data obtained in the forms of graphs and charts to educate the user easily and effectively |
| 4. | Social Impact / Customer Satisfaction | 1.Helps users clear the doubt if they actually are suffering from any heart related issues or not<br><br>2.Identifies heart diseases if any, and refers specialists to the user |
| 5. | Business Model (Revenue Model) | Heart specialists and hospitals can use the interactive dashboard to keep track of patient health and receive notifications regarding the same |
| 6. | Scalability of the Solution | To identify and predict other similar diseases involving other organs using respective datasets(E.g Lungs, Kidney, etc) |

## 3.4 Problem Solution Fit



# 4. Requirement Analysis

## 4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

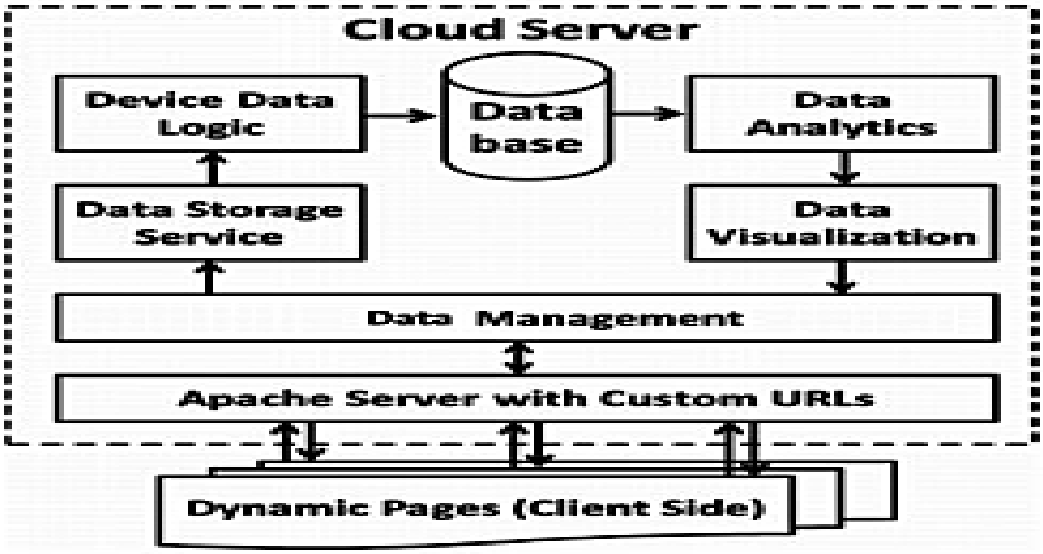
| FR No. | Functional Requirement(Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email |
| FR-3 | Data Analysis | Obtained data is analysed and segregated based on set criteria (blood pressure, cholesterol levels etc.) |
| FR-4 | Data Visualization and dashboard creation | User can visualise the trends on the heart diseases via graphs, charts etc. in the IBM Cognos dashboard that is created<br>Reports are created based on the trends that the user can view |

## 4.2 Non- Functional Requirement

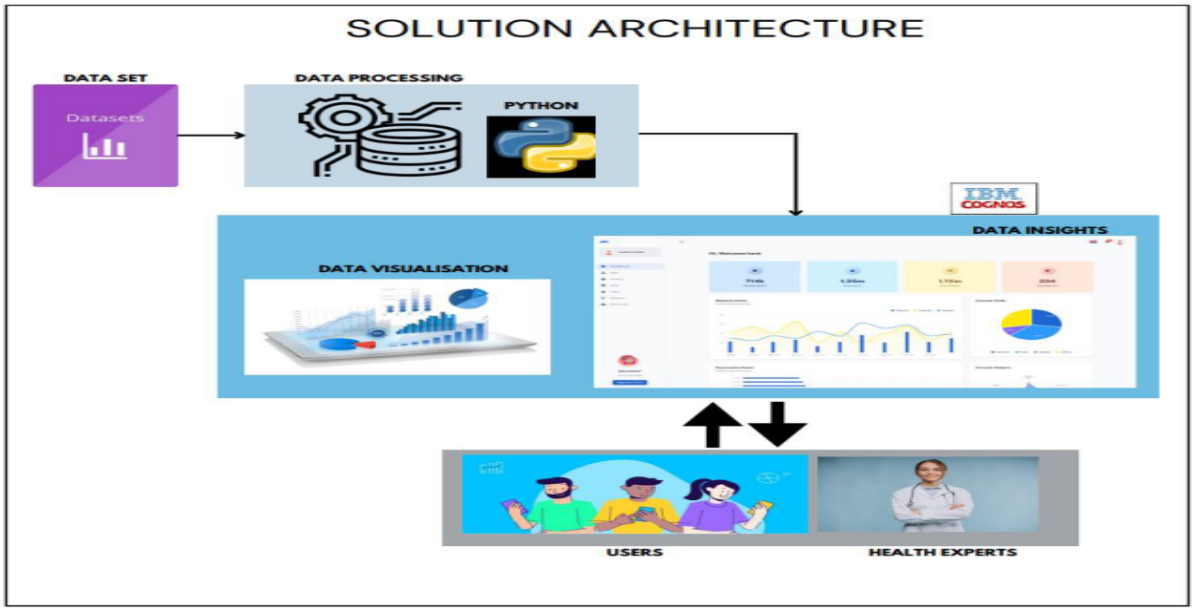
| FR No. | Non-FunctionalRequirement | Description |
|---|---|---|
| NFR-1 | **Usability** | The application will have a simple and user-friendly graphical interface. Users will be able to understand and use all the features of the application easily.Any action has to be performed with just a few clicks |
| NFR-2 | **Security** | For security of the application the technique known as database replication should be used so that all the important data should bekept safe. In case of crash, the system should be able to backup and recover the data |
| NFR-3 | **Reliability** | The application has to be consistent at every scenario and has to work without failure in any environment |
| NFR-4 | **Performance** | Performance of the application depends on the response time andthe speed of the data submission. The response time of the application is direct and faster which depends on the efficiency of implemented algorithm |
| NFR-5 | **Availability** | The application has to be available 24 x 7 for users without anyinterruption |
| NFR-6 | **Scalability** | The application can withstand the increase in the no. of usersand has to be able to develop Higher versions |

# 5. Project Design

## 5.1 Data flow Diagram




## 5.2 Solution & Technical Architecture



# 6.PROJECT PLANNING AND SCHEDULING

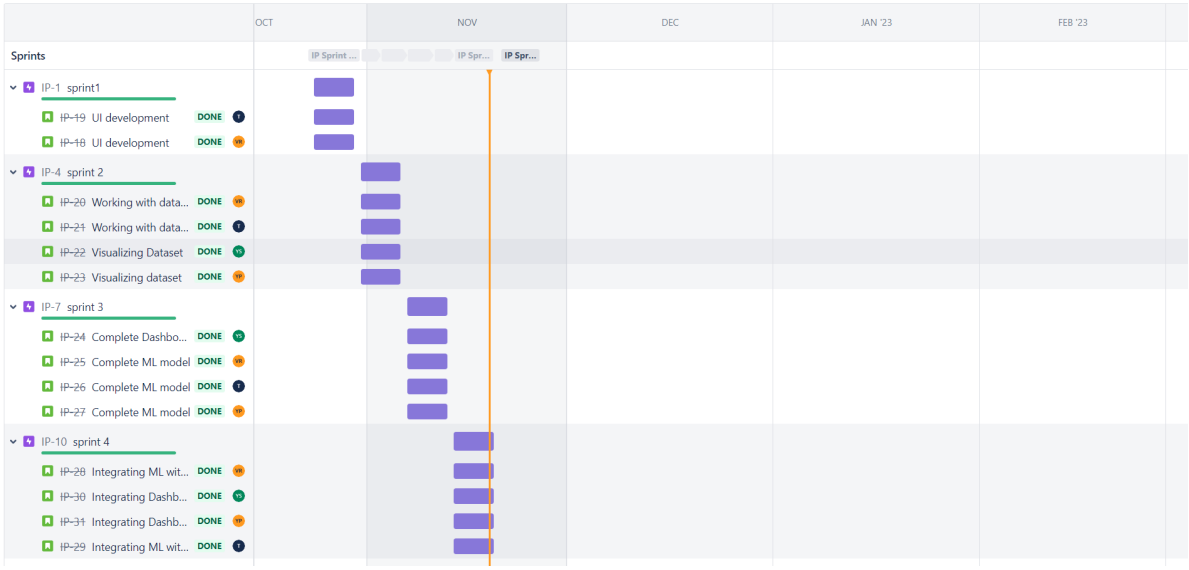
## 6.1 Script Planning and Execution

| 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. | 9 | High | Vishal |
| | | | As a user, I will receive confirmation email once I have registered for the application. | 4 | Low | Yeswanthraj |
| | | USN-2 | As a user, I can register for the application through Gmail. | 7 | Medium | Yash |
| | Login | USN-3 | As a user, I can log into the application by entering email & password. | 9 | High | Thayalan |
| Sprint-2 | Working with the Dataset | USN-4 | Importing the dataset on cognos platform and understand, clean and prepare the dataset. | 9 | High | Vishal, Thayalan |
| | Data Visualization chart | USN-5 | After importing the dataset, we create some visualizations to understand more about the predicting heart diseases. | 7 | Medium | Yeswanthraj, Yash |
| Sprint-3 | Creating the Dashboard | USN-6 | Creating the dashboard to display the visualizations which gives insights of predicting the Heart diseases. | 9 | High | Yash, Vishal, Yeswanthraj |
| Sprint-4 | Export the Analytics | USN-7 | Exporting the created dashboard to showcase the work to others. | 9 | High | Thayalan, Vishal, Yeswanthraj |

## 6.2 Sprint Delivery and 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 | 15 Nov 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 16 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 17 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

# 6.3 Jira Report



# 7.CODING AND SOLUTIONING

## 7.1 Machine Learning

## RF classifier model

Random Forest Classifier

In [15]:
```
max_accuracy = 0

for x in range(500):
    rf_classifier = RandomForestClassifier(random_state=x)
    rf_classifier.fit(X_train,Y_train)
    Y_pred_rf = rf_classifier.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

print(max_accuracy)
print(best_x)
```

```
85.19
135
```

In [16]:
```
rf_classifier = RandomForestClassifier(random_state=best_x)
rf_classifier.fit(X_train,Y_train)
Y_pred_rf = rf_classifier.predict(X_test)
Y_pred_rf.shape
```

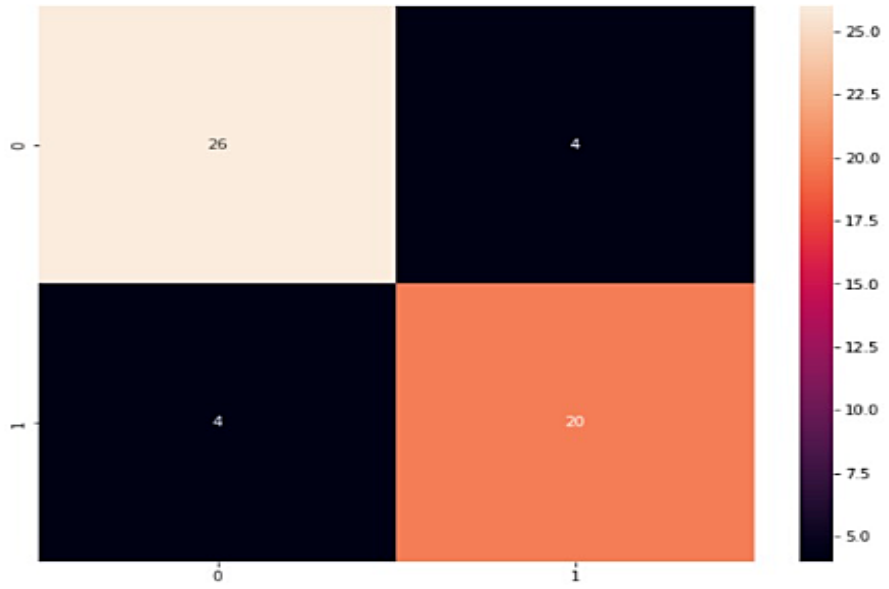
Out[16]: (54,)

In [17]:
```
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
score_rf
```

Out[17]: 85.19

Out[18]:

| | Model | Accuracy | Precision | Sensitivity | Specificity | F1 Score | ROC | Log_Loss | mathew_corrcoef |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Random Forest | 0.851852 | 0.866667 | 0.866667 | 0.866667 | 0.866667 | 0.85 | 19.188653 | 0.7 |



# KNN Model

K-Nearest Neighbours

In [21]:
```
knn_classifier= KNeighborsClassifier(n_neighbors=31,leaf_size=30)
knn_classifier.fit(X_train,Y_train)
Y_pred_knn = knn_classifier.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
score_knn
```

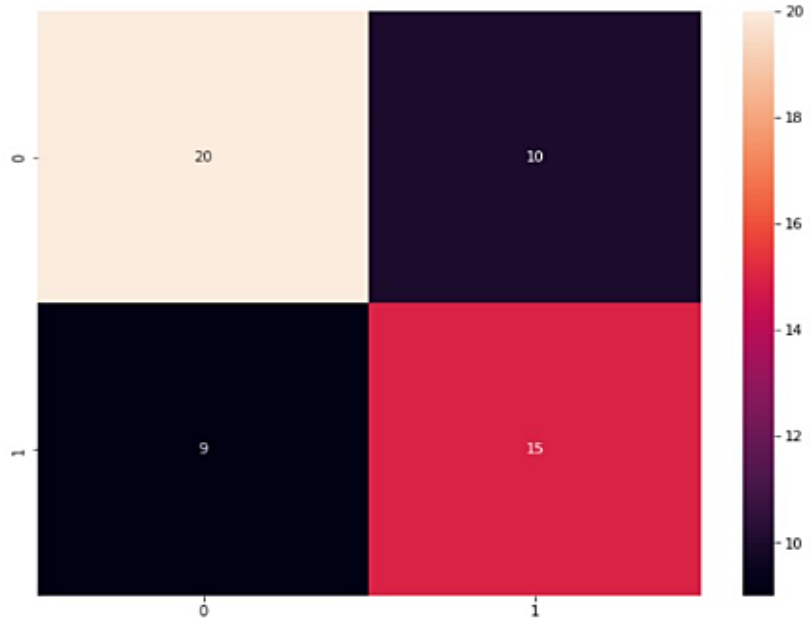
Out[21]: 64.81

Model Evaluation

In [22]:
```
y_pred_knne = knn_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_knne)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_knne)
acc= accuracy_score(Y_test, y_pred_knne)
roc=roc_auc_score(Y_test, y_pred_knne)
prec = precision_score(Y_test, y_pred_knne)
rec = recall_score(Y_test, y_pred_knne)
f1 = f1_score(Y_test, y_pred_knne)

mathew = matthews_corrcoef(Y_test, y_pred_knne)
model_results =pd.DataFrame([['K-Nearest Neighbors ',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
               columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

Out[22]:

| | Model | Accuracy | Precision | Sensitivity | Specificity | F1 Score | ROC | Log_Loss | mathew_corrcoef |
|---|---|---|---|---|---|---|---|---|---|
| 0 | K-Nearest Neighbors | 0.648148 | 0.689655 | 0.666667 | 0.666667 | 0.677966 | 0.645833 | 19.188653 | 0.290659 |



# Decision Tree Classifier model

Decision Tree Classifier

In [25]:
```
dt_classifier = DecisionTreeClassifier(
    max_depth=20,
    min_samples_split=2,
    min_samples_leaf=1,
    min_weight_fraction_leaf=0.00001,
    max_features='auto',
    random_state=46)
dt_classifier.fit(X_train, Y_train)
Y_pred_dtedt_classifier.predict(X_test)
score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
score_dt
```

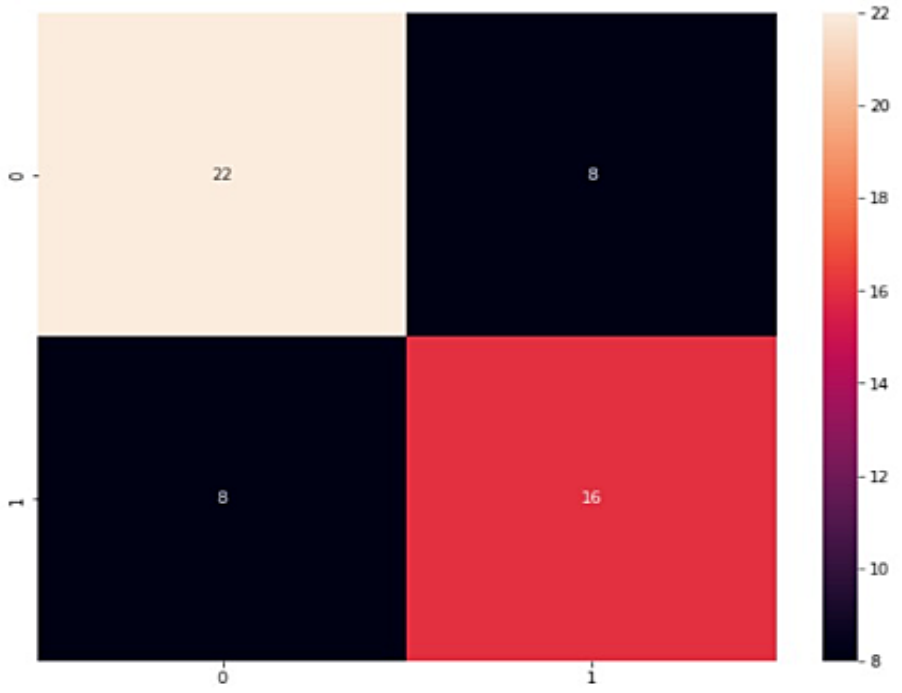
Out[25]: 70.37

Model Evaluation

In [26]:
```
y_pred_dte = dt_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CMsconfusion_matrix(Y_test,y_pred_dte)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_dte)
acc= accuracy_score(Y_test, y_pred_dte)
roc=roc_auc_score(Y_test, y_pred_dte)
prec = precision_score(Y_test, y_pred_dte)
rec = recall_score(Y_test, y_pred_dte)
f1 = f1_score(Y_test, y_pred_dte)

mathew = matthews_corrcoef(Y_test, y_pred_dte)
model_results =pd.DataFrame([['Decision Tree',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
             columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

Out[26]:

| | Model | Accuracy | Precision | Sensitivity | Specificity | F1 Score | ROC | Log_Loss | mathew_corrcoef |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Decision Tree | 0.703704 | 0.733333 | 0.733333 | 0.733333 | 0.733333 | 0.7 | 19.188653 | 0.4 |



# Naive Bayes Classifier Model

Naive Bayes Classifier

In [29]:
```
nb_classifier = GaussianNB( var_smoothing=1e-50)
nb_classifier.fit(X_train,Y_train)
nb_classifier.predict(X_test)
Y_pred_nb = nb_classifier.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
score_nb
```

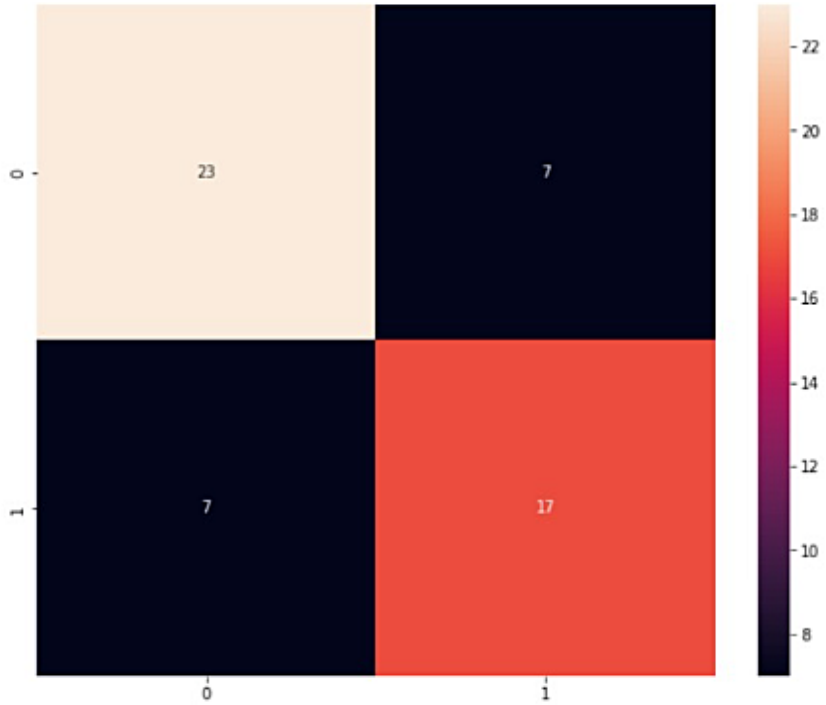
Out[29]: 74.07

Model Evaluation

In [30]:
```
y_pred_nbe = nb_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_nbe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_nbe)
acc= accuracy_score(Y_test, y_pred_nbe)
roc=roc_auc_score(Y_test, y_pred_nbe)
prec = precision_score(Y_test, y_pred_nbe)
rec = recall_score(Y_test, y_pred_nbe)
f1 = f1_score(Y_test, y_pred_nbe)

mathew = matthews_corrcoef(Y_test, y_pred_nbe)
model_results =pd.DataFrame([['Naive Bayes ',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
               columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

Out[30]:

| | Model | Accuracy | Precision | Sensitivity | Specificity | F1 Score | ROC | Log_Loss | mathew_corrcoef |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Naive Bayes | 0.740741 | 0.766667 | 0.766667 | 0.766667 | 0.766667 | 0.7375 | 19.188653 | 0.475 |



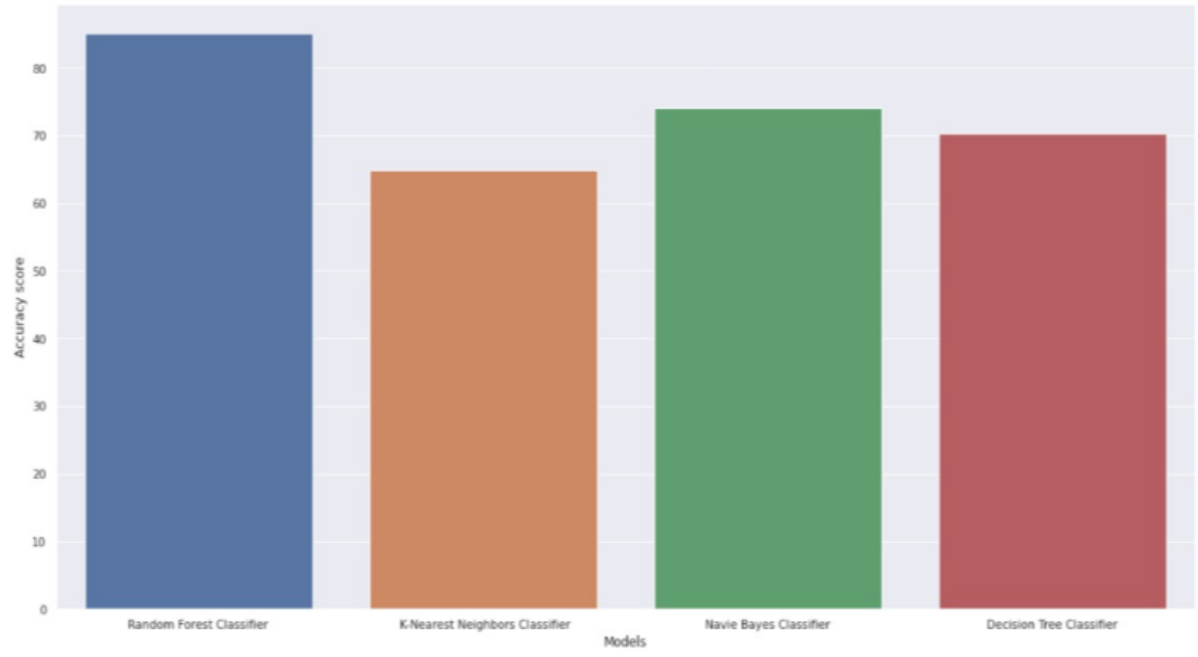
# Thus, selecting The Best Working ML model

Final Selection of the ML Model

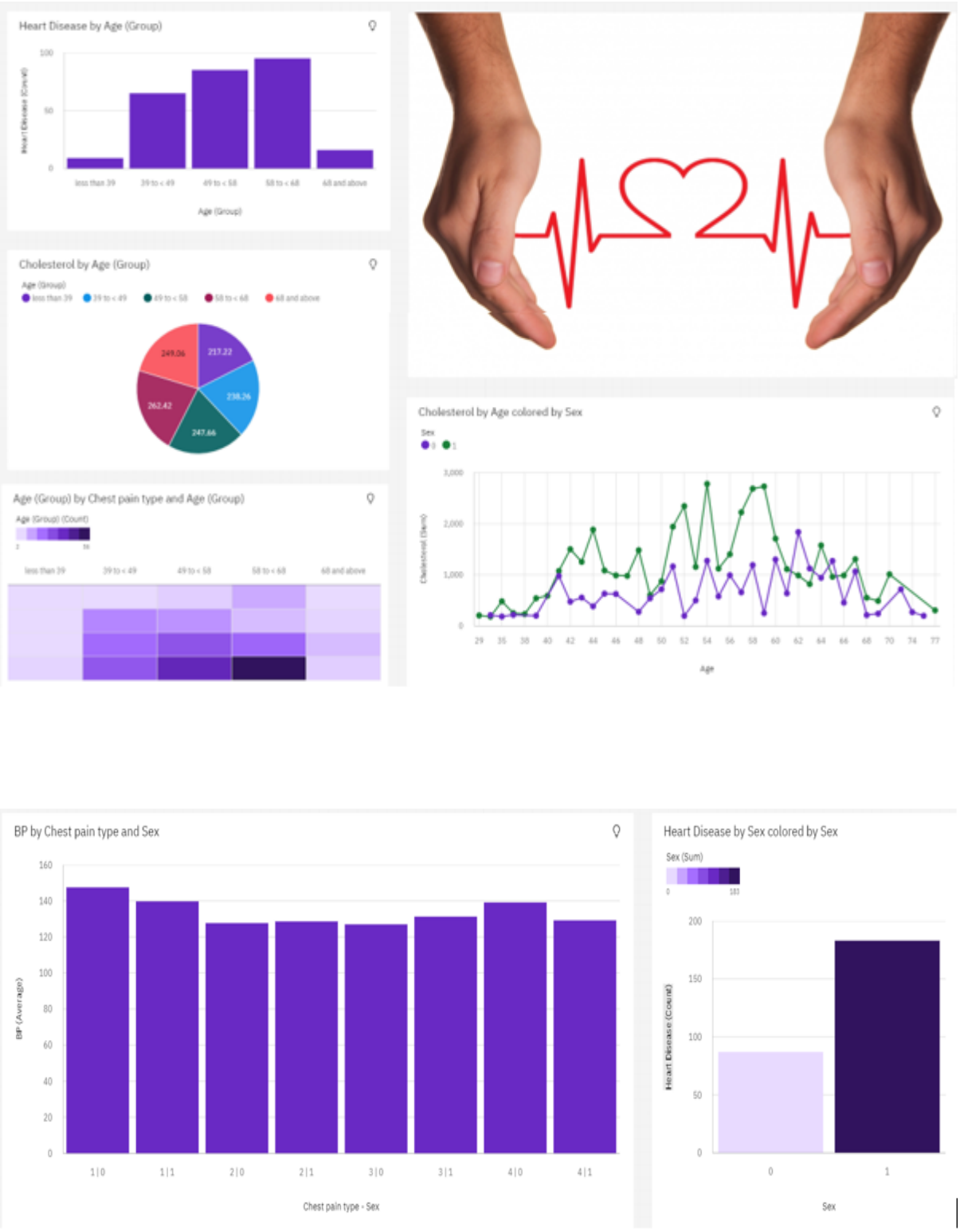
```
scores = [score_rf,score_knn,score_nb,score_dt]
Models = ["Random Forest Classifier"," K-Nearest Neighbors Classifier","Navie Bayes Classifier","Decision Tree Classifier"]

for i in range(len(Models)):
    print("The accuracy score achieved using "+Models[i]+" is: "+str(scores[i])+" %")
```

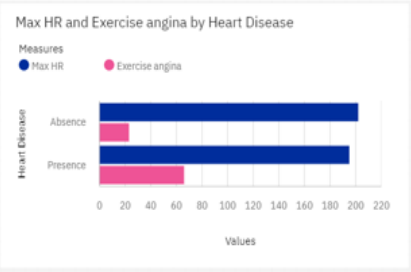
```
The accuracy score achieved using Random Forest Classifier is: 85.19 %
The accuracy score achieved using  K-Nearest Neighbors Classifier is: 64.81 %
The accuracy score achieved using Navie Bayes Classifier is: 74.07 %
The accuracy score achieved using Decision Tree Classifier is: 70.37 %
```

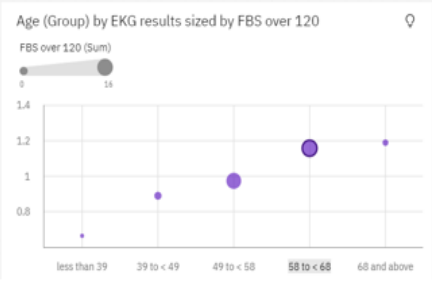
```
sns.set(style="darkgrid",rc={'figure.figsize':(20,10)})
plt.xlabel("Models")
plt.ylabel("Accuracy score")

sns.barplot(Models,scores)
plt.savefig("AccuracyScores.png")
```

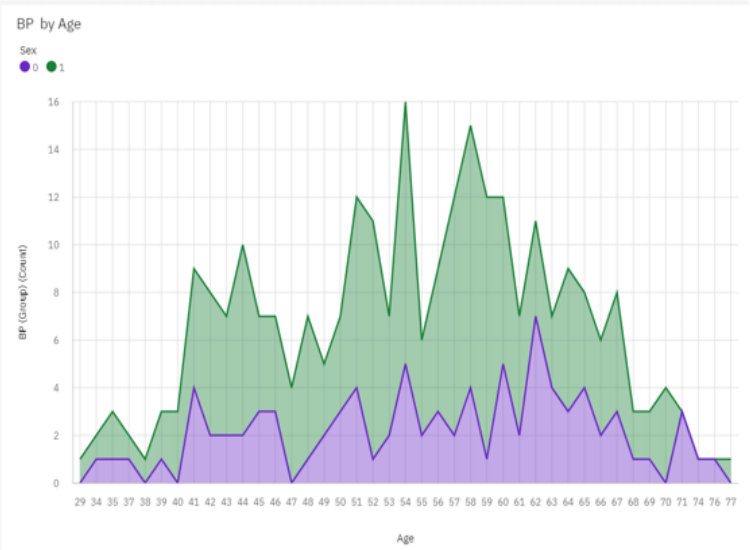


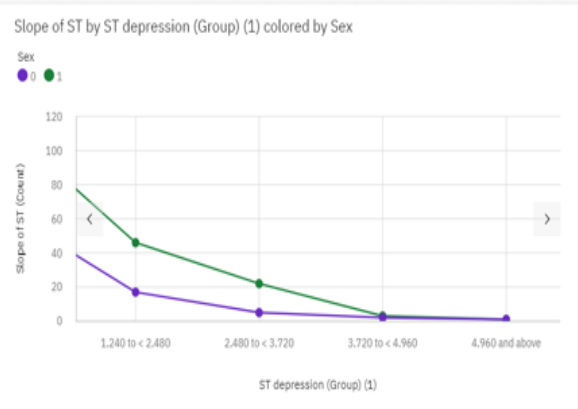
# 7.2 Dashboard

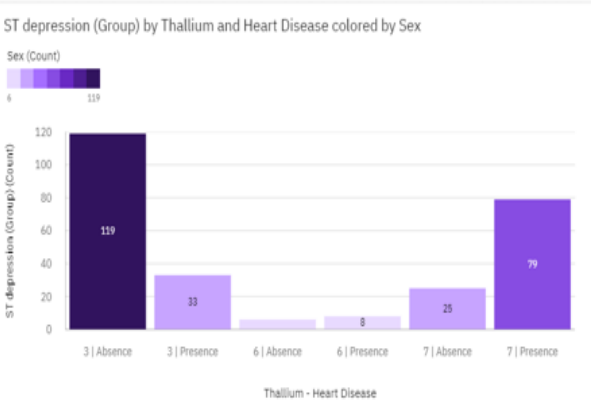



Max HR and Exercise angina by Heart Disease
Measures
Max HR
Exercise angina
Heart Disease
Absence
Presence
0 20 40 60 80 100 120 140 160 180 200 220
Values


Age (Group) by EKG results sized by FBS over 120
FBS over 120 (Sum)
0
16
1.4
1.2
1
0.8
less than 39
39 to < 49
49 to < 58
58 to < 68
68 and above


BP by Age
Sex
0
1
BP (Group) (Count)
16
14
12
10
8
6
4
2
0
29 34 35 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 74 76 77
Age


Slope of ST by ST depression (Group) (1) colored by Sex
Sex
0
1
Slope of ST (Count)
120
100
80
60
40
20
0
1.240 to < 2.480
2.480 to < 3.720
3.720 to < 4.960
4.960 and above
ST depression (Group) (1)


ST depression (Group) by Thallium and Heart Disease colored by Sex
Sex (Count)
6
119
ST depression (Group) (Count)
120
100
80
60
40
20
0
119
33
8
25
79
3 | Absence
3 | Presence
6 | Absence
6 | Presence
7 | Absence
7 | Presence
Thallium - Heart Disease

# 8.TESTING

## 8.1 Test Cases

```
s_pred = rf_classifier.predict([[40,1,2,120,200,1,0,150,0,0.3,1,1,4]])
if(int(s_pred)==2):
  op="Present"
else:
  op="Absent"
print("Predicted Heart condition:",op)
```

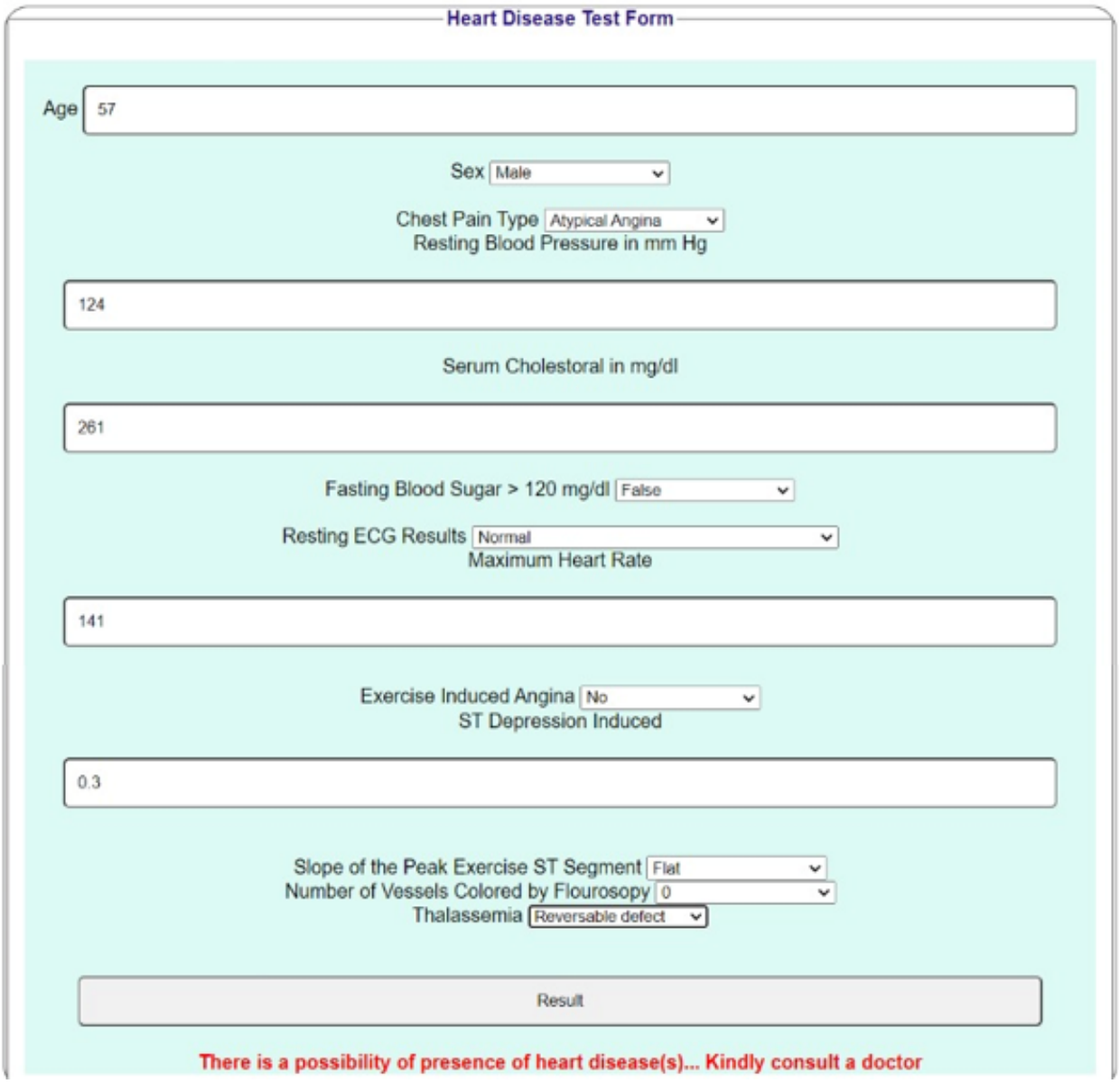
```
Predicted Heart condition: Absent
```

```
s_pred = rf_classifier.predict([[57,1,2,124,261,0,0,141,0,0.3,1,0,7]])
if(int(s_pred)==2):
  op="Present"
else:
  op="Absent"
print("Predicted Heart condition:",op)
```

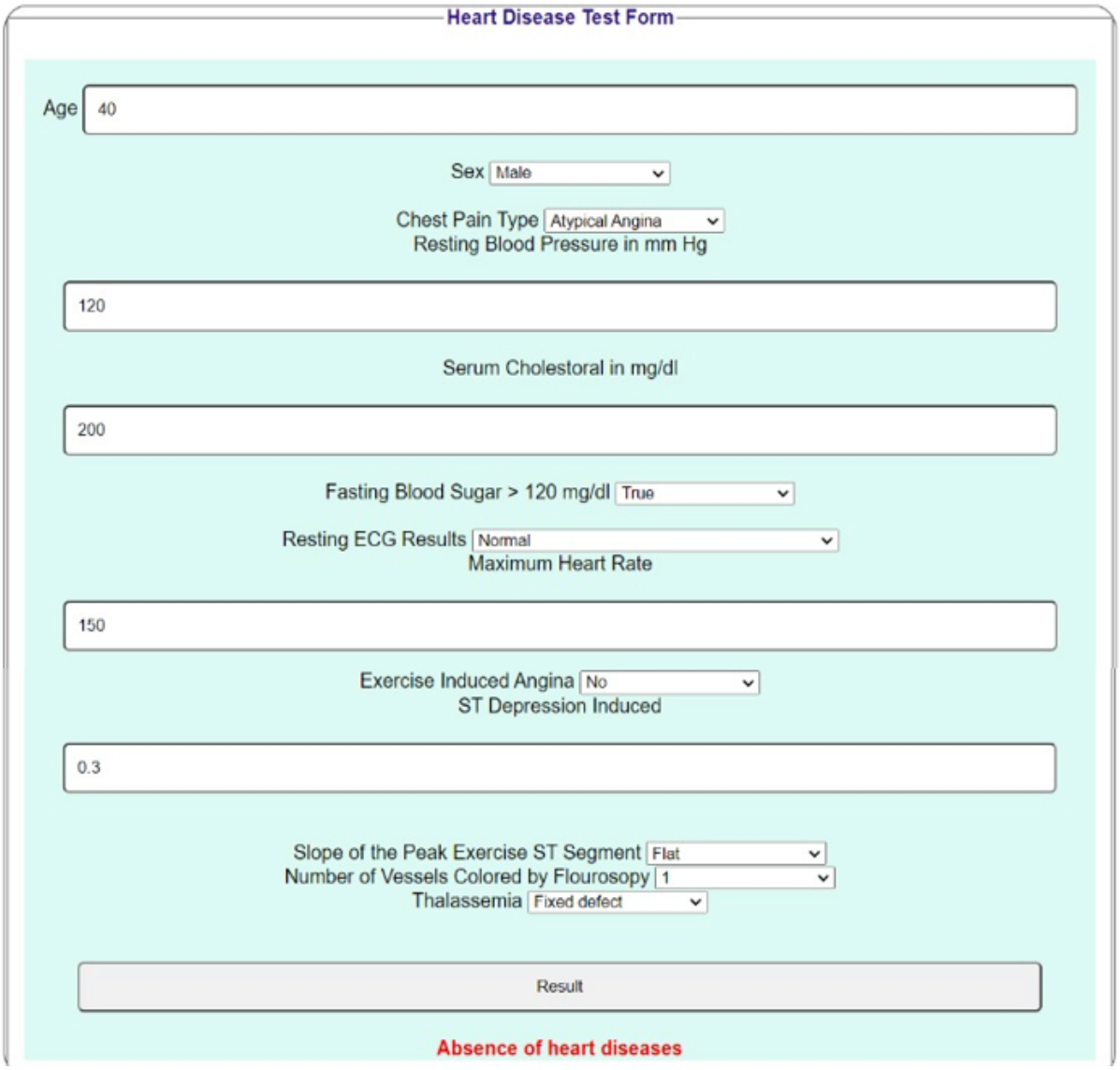
```
Predicted Heart condition: Present
```

## 8.2 User acceptance Testing

a) User having heart disease



b) User not having heart disease



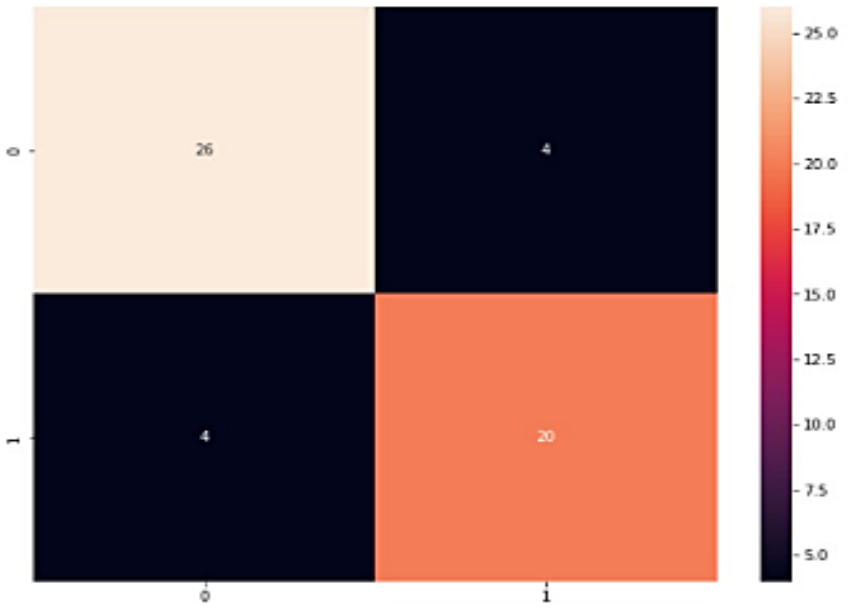
# 9. RESULT

## 9.1 Performance Metrics

In [ ]:
```
y_pred_rfe = rf_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_rfe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_rfe)
acc= accuracy_score(Y_test, y_pred_rfe)
roc=roc_auc_score(Y_test, y_pred_rfe)
prec = precision_score(Y_test, y_pred_rfe)
rec = recall_score(Y_test, y_pred_rfe)
f1 = f1_score(Y_test, y_pred_rfe)

mathew = matthews_corrcoef(Y_test, y_pred_rfe)
model_results =pd.DataFrame([['Random Forest',acc, prec,rec,specificity, f1,roc, loss_log,mathew]],
               columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1 Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
```

Out[ ]:

| | Model | Accuracy | Precision | Sensitivity | Specificity | F1 Score | ROC | Log_Loss | mathew_corrcoef |
|---|---|---|---|---|---|---|---|---|---|
| 0 | Random Forest | 0.851852 | 0.866667 | 0.866667 | 0.866667 | 0.866667 | 0.85 | 19.188653 | 0.7 |

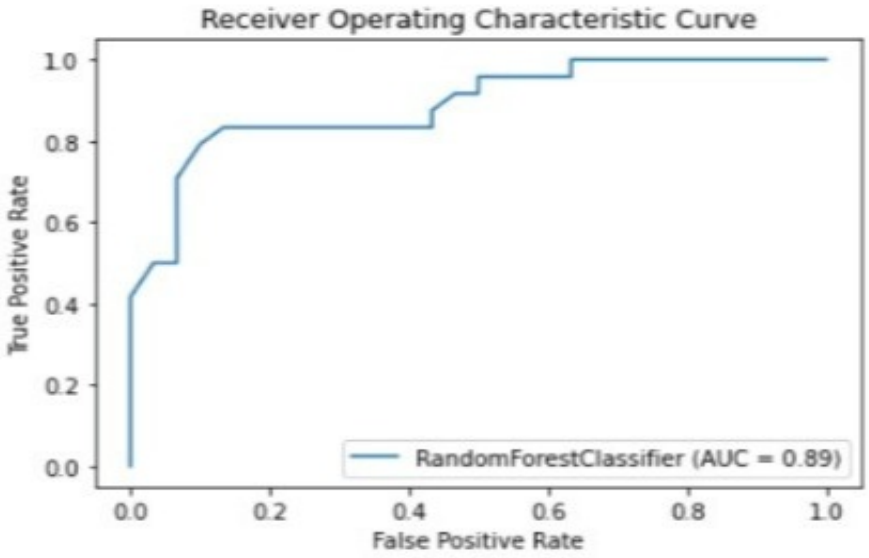



In [ ]:
```
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
```

```
              precision    recall  f1-score   support

           1       0.87      0.87      0.87        30
           2       0.83      0.83      0.83        24

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

In [ ]:

```
plot_roc_curve(rf_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve');
plt.savefig("RF.png")
```



## 10.ADVANTAGES AND DISADVANTAGES

### Advantages:

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

### Disadvantages:

1. Users need to know all the fields
2. Does Not take null value as input
3. Does not provide suggestions to user

## 11. CONCLUSION

Complications of heart disease include heart attack and stroke. You can reduce the risk of complications with early diagnosis and treatment. So the suggestion that we get from the website might help save patients. It is always to get treated in the early stages of heart disease.

# 12. FUTURE SCOPE

Like the saying goes “Prevention is better than cure”. We have to look into methods to prevent heart diseases altogether other than just predicting it in early stages.
To use this website we need to take a substantial amount of tests beforehand. So it would be better if we require less attributes and still give an effective result.

# 13. APPENDIX

**Source Code:**

https://github.com/IBM-EPBL/IBM-Project-23339-1659878869/tree/main/Project%20Development%20Phase/Sprint%204

**Demo Video Link:**

**https://drive.google.com/file/d/1EYh6khHirYeOzUDCuuRVjTn89RfL5YQN/view?usp=drivesdk**