DATA ANALYTICS

VISUALIZING AND PREDICTING HEART DISEASES WITH AN INTERACTIVE DASHBOARD

TEAM ID : PNT2022TMID28916

TEAM MEMBERS:

1. SRISHTI.R
2. JENEFA REGINA MARY
3. JESSICA TIFFANY.D
4. HARINI.M

INDEX

1. INTRODUCTION

1.1 Project Overview

In many nations, cardiovascular disease is the main cause of death. Cardiovascular illness is frequently identified by doctors based on the results of recent clinical testing and their prior experience treating patients who presented with comparable symptoms. Analytics is an essential technique for any profession because it predicts the future and uncovers hidden patterns. In recent years, data analytics has been regarded as a cost-effective technology, and it plays an important role in healthcare, including new research findings, emergency situations, and disease outbreaks. The use of analytics in healthcare improves care by facilitating preventive care, and EDA is a critical step when analysing data. The risk factors that cause heart disease are considered and predicted using the K-means algorithm, and the analysis is carried out using publicly available heart disease data. The K-means clustering algorithm, in conjunction with data analytics and visualization tools, is used to predict heart disease. Pre-processing methods, classifier performance, and evaluation metrics are all covered. The visualized data in the result section shows that the prediction is correct.

1.2 Purpose

The diagnosis of heart illness is a difficult undertaking, but it can provide an automated prognosis of the patient's heart status to help with subsequent treatment. Typically, the patient's physical examination, signs, and symptoms of heart disease serve as the foundation for the diagnosis. Due to their lifestyle choices and the state of the environment today, individuals are susceptible to many diseases. To prevent the severity of these disorders, early detection and prediction of their occurrence are crucial. Predictive analytics in healthcare can raise the standard of care, gather more clinical data for individualized treatment, and correctly identify each patient's medical condition. For the purpose of making wise decisions, healthcare businesses gather enormous amounts of data that may contain some hidden information. Some sophisticated data mining techniques are utilized to deliver accurate results and make data-driven judgments. In this study, we established a project for estimating the degree of heart disease risk using a neural network. For prediction, the algorithm makes use of 12 medical variables, including age, sex, blood pressure, cholesterol, and obesity.

2. LITERATURE SURVEY

2.1 Existing problem

"Predicting the Risk of Heart Failure With EHR Sequential Data Modelling," suggested a neural network-based approach. This essay utilized real-world datasets derived from electronic health record (EHR) data connected to congestive heart failure to complete the experiment and foretell cardiac disease in advance. We tend to utilize word vectors and one-hot encryption when modeling the cardiac failure events predicted by the diagnosis using a long memory's fundamental tenets against its network theory. Analyzing the outcomes, we frequently find the significance of honoring clinical procedures' logical sequence documents. In the literature, methods for identifying cardiac disease include waveform analysis, time-frequency analysis, Neuro-Fuzzy RBF ANN, and Total Least Square-based Prony modelling algorithms. However, Marshall et al investigation's found (Marshall et al 1991), This method's classification accuracy (up to 79%) was poor, and choosing the best model required considering a range of modifications was still adequate.

Heart Attack Prediction and Visualization of Contributing Factors Using Machine Learning" It associates many risk factors in heart disease and a need for time to get accurate, reliable, and sensible approaches to make an early diagnosis to achieve prompt management of the disease. Data analysis and machine learning are the most commonly used techniques for processing enormous data in the healthcare domain. Researchers apply several data mining and machine learning techniques to analyze huge complex medical data, helping healthcare professionals to predict heart disease. This research paper presents various attributes like age, gender, chest pain, cholesterol, etc which are used to predict heart attack, and the model is trained using 4 machine learning algorithms namely- Logistic Regression, Gaussian Naïve Bayes, Decision tree, and Random Forest algorithm. It uses the existing dataset from the UCI Heart Disease Data set of heart disease patients. The dataset comprises 303 instances and 76 attributes. This research paper aims to envision the probability of developing heart attacks in patients. The results portray that the highest accuracy score is achieved with Logistic Regression.

This research paper "A novel approach for heart disease prediction using strength scores with significant predictors" talks about CVDs are disorders of the heart and blood vessels and include coronary heart disease, cerebrovascular disease, and other conditions. Heart attacks and strokes are the main causes of mortality in cardiovascular disease with a rate near one out of three. With the high rate of mortality, diagnosis and prevention measures need to be performed effectively and efficiently. Many data mining techniques have been used to help address these issues. Most of the past Heart Attack Prediction and Visualization of Contributing Factors Using Machine Learning research looked into identifying features that contribute to better heart prediction accuracy. However, very little research looked into the relationships that exist between these features. The association between each feature that contributes to heart disease prediction can be obtained using the Associative Rule Mining (ARM) technique.

The ARM technique is popular in transactional and relational datasets. The hidden knowledge in large datasets such as business transactions developed the interest of many

business owners to understand the patterns that can help them to improve their business decisions (Agarwal and Mithal). For instance, discovering the frequently bought items by customers in market basket analysis.

2.2 References

[1]" Predicting the Risk of Heart Failure With EHR Sequential Data Modeling," Bo Jin, Chao Che.

[2]“Heart Attack Prediction and Visualization of Contributing Factors Using Machine Learning” by Megha Banerjee, Reetodeep Hazra, Suvranil Saha, Megha Bhushan, Subhankar Bhattacharjee.

[3]“A novel approach for heart disease prediction using strength scores with significant Predictors” by Armin Yazdani, Kasturi Dewi Varathan, Yin Kia Chiam, Asad Waqar Malik, and Wan Azman Wan Ahmad.

[4]“Heart Disease Risk Prediction Using Machine Learning Classifiers with Attribute Evaluators” by Karna Vishnu Vardhana Reddy, Irraivan Elamvazuthi, Azrina Abd Aziz, Sivajothi Paramasivam, Hui Na Chua, and S. Pranavanand

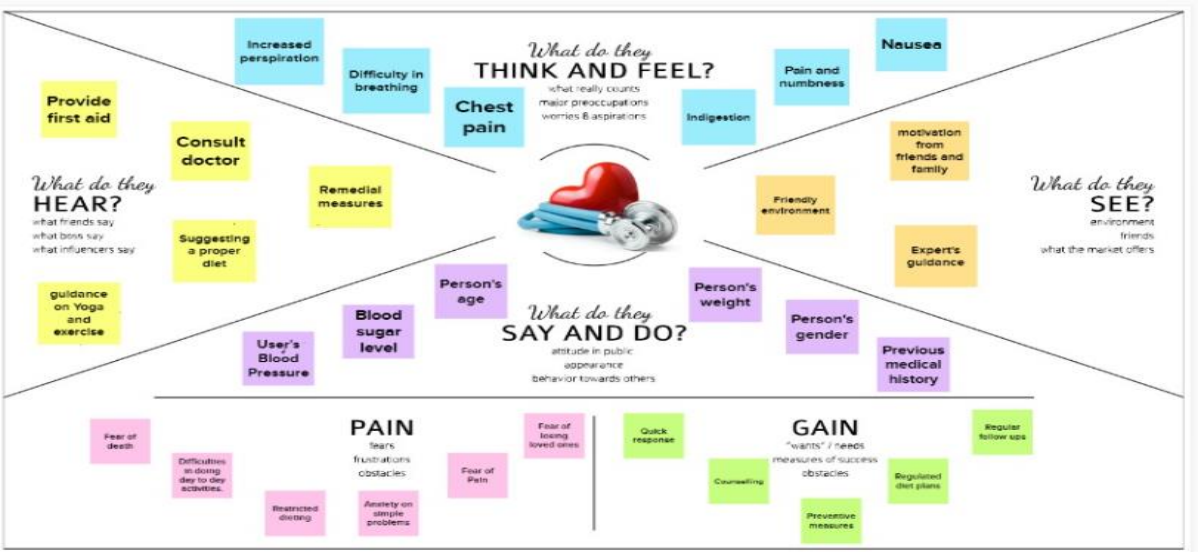
2.3 Problem Statement Definition

Heart acts a major role in the corporeal organisms. The diseases of the heart want more perfection and exactness for diagnosis and analyses. Heart disease is a dangerous disease. This disease occurs due to various problems such as overpressure, blood sugar, high blood pressure, Cholesterol, etc. in the human body By using Python and machine learning, this paper is analyzed and predicted heart disease. We can predict this disease by using various attributes in the data set. We have collected a data set consisting of 13 elements and 383 individual values to analyze the patient's performance. The main aim of the paper is to get better accuracy to detect heart disease using the ML algorithm.

Problem Statement(PS)	**I am (Customer)**	**I’m trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Cust1	Check if I have any heart disease	I have Diabetes	Age factor	Stressed
PS-2	Cust2	Check if I have any heart disease	I am completely healthy	Healthy Diet	Relieved

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



A psychological aid to the general practitioner creates an empathy map for a patient with a lifestyle issue during a consultation.

3.2 Ideation & Brainstorming

In order to make the dataset we are working with easier to understand, we will first take a look at data Wrangling A timely diagnosis of heart disease (HD), one of the most prevalent diseases today, is essential for many healthcare professionals to protect their patients from the condition and save lives. To accurately classify and/or predict HD cases with a few variables, a comparison analysis of various classifiers can be done for the classification of the heart disease dataset.

Accurate decision-making and ideal therapy are needed to address cardiac risk. Five machine learning models were utilized in a Canadian study to examine 1-month mortality among hospitalized patients with congestive heart failure. Several tests, including auscultation, blood pressure, cholesterol, ECG, and blood sugar, are carried out prior to the diagnosis of a condition. These tests assist in identifying the patient's medication requirements. In this work, the predictive accuracy of various machine learning methods is investigated to calculate cardiovascular risk. The performance comparison of the most recent REP Tree and Random Tree machine learning algorithms in terms of cardiovascular disease prediction is innovative.

Pre-processing of Data:

You can eliminate the missing numbers or use the mean value in their place. Therefore, employing a filtering strategy, the data obtained must be slightly adjusted in order to conduct a good study. Here, the multifilter method is applied.

Extraction of Features:

Reduce the number of input attributes prior to data processing. Not all characteristics affect prediction success in the same way. Multiple attributes lead to increased complexity and worse performance. It is necessary to carefully extract features without sacrificing system performance as a result.

Data Collection and Processing Data Collection:

The dataset was compiled from multiple patient records that included 14 attributes such as restagc, fbs, thal, ca, cp, sex, age, thalach, chol, trestbps, slope, oldpeak, exangand target. The table below contains a description of the attributes used for analysis. The visualization shows that age does not play a significant role in predicting heart disease because the same age groups have an equal number of people with and without heart disease. Outlier detection and data preprocessing. For each attribute or feature, we calculate the Z-score of each individual value of that attribute in relation to the column mean and standard deviation. After that, take the magnitude or absolute value of the obtained z score. If the z score is less than a certain threshold, a particular row or record is an outlier and is removed.

DM Techniques	Accuracy	Model Construction Time	Mean Absolute Error
Naive Bayes	96.5%	0.02s	0.044
Decision Tree	99.2%	0.09s	0.00016
Classification via Clustering	88.3%	0.06s	0.117

3.3 Proposed Solution

3.3.1. 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.

3.3.2. Idea / Solution description

An interactive dashboard that visualizes and predicts using Machine Learning algorithms.

3.3.3. Novelty / Uniqueness

We are employing the Naive Bayes algorithm in this system to develop an efficient heart attack prediction system. The system can receive input from a CSV file or manually. After receiving input, the Nave Bayes algorithm is used on that input. After gaining access to the data set, the process is carried out, and a useful heart attack level is generated. The suggested approach will include additional heart attack risk factors such as weight, age, and priority levels after speaking with doctors and other medical professionals.

3.3.4. Social Impact / Customer Satisfaction

It helps in time of emergencies. If any heart problem is predicted in advance, we can provide valuable insights to clinicians, allowing them to tailor their diagnosis and treatment to each individual patient.

3.3.5. Business Model (Revenue Model)

For prediction, the algorithm makes use of 15 medical variables, including age, sex, blood pressure, cholesterol, and obesity. The likelihood that a patient may develop heart disease is predicted by the EHDPS. It makes it possible to build linkages between important knowledge, including patterns between medical parameters associated with heart disease.

3.3.6. Scalability of the Solution

Our key contribution to this work is to predict heart disease diagnosis using a modest number of parameters. Our prediction system employs random forest on Apache Spark, allowing healthcare analysts to deploy this solution on a constantly changing, scalable big data landscape for informed decision-making. We demonstrate that this method achieves up to 98% accuracy. We also compare our classifier to the Nave-Bayes classifier.

ALGORITHMS USED:

1. Logistic Regression Algorithm:

Logistic regression is a sort of statistical regression analysis that predicts the outcome of a categorical dependent variable from a set of predictor or independent variables. The dependant variable in logistic regression is always binary. Logistic regression is mostly used for prediction and calculating the likelihood of success.

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

2. Naïve Bayes Algorithm:

Naive Bayes algorithms are frequently used in sentiment analysis, spam filtering, recommendation systems, and other applications. They are quick and simple to implement, but their main limitation is that predictors must be independent.

- It is straightforward and simple to apply.
- It does not required as much training data.
- It can work with both continuous and discrete data.
- The number of predictors and data points is extremely scalable.
- It is quick and can provide real-time forecasts.

```
for z in range(5):
    print("\nTest Train Split no. ",z+1,"\n")
    nx_train,nx_test,ny_train,ny_test = train_test_split(nx,ny,test_size=0.25,random_state=
    # Gaussian function of sklearn
    gnb = GaussianNB()
    gnb.fit(nx_train, ny_train.ravel())
    ny_pred = gnb.predict(nx_test)
```

Training the Naïve Bayes algorithm

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

```
Naive Bayes model accuracy(in %): 0.7794117647058824
```

3. KNN Algorithm:

One of the simplest machine learning algorithms, based on the supervised learning method, is K-Nearest Neighbour.

The K-NN algorithm makes the assumption that the new case and the existing cases are comparable, and it places the new instance in the category that is most like the existing categories.

A new data point is classified using the K-NN algorithm based on similarity after all the existing data has been stored. This means that utilising the K-NN method, fresh data can be quickly and accurately sorted into a suitable category.

Although the K-NN approach is most frequently employed for classification problems, it can also be utilised for regression.

Since K-NN is a non-parametric technique, it makes no assumptions about the underlying data.

```
knn = KNeighborsClassifier(n_neighbors = 20)
knn.fit(KX_train, KY_train)
print(knn.score(KX_test, KY_test))
```

```
0.6111111111111112
```

Training the data set using KNN algorithm

4. Decision Tree Algorithm:

The Decision Tree algorithm is a member of the supervised learning algorithm family. The decision tree approach, unlike other supervised learning algorithms, may also be utilized to solve regression and classification issues.

The purpose of employing a Choice Tree is to build a training model that can predict the class or value of a target variable by learning basic decision rules from prior data (training data).

In Decision Trees, we begin at the root of the tree to forecast a class label for a record. We compare the values of the root attribute with the attribute of the record. Based on the comparison, we follow the branch corresponding to that value and proceed to the next node

```
for x in range(200):
    dt = DecisionTreeClassifier(random_state=x)
    dt.fit(DX_train,dy_train)
    dy_pred_dt = dt.predict(DX_test)
    current_accuracy = round(accuracy_score(dy_pred_dt,dy_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
```

.

```
print("The accuracy score achieved using Decision Tree is: "+str(score_dt))
```

```
The accuracy score achieved using Decision Tree is: 0.7962962962962963
```

Accuracy of Decision tree algorithm

Overall Comparison of all 4 Algorithm:

```
print('Logistic Regression :',l)
print('KNN :',k)
print('Naive Bayes :',n)
print('Decision Tree :' ,d)
```

```
Logistic Regression : 0.8271604938271605
KNN : 0.6111111111111112
Naive Bayes : 0.7794117647058824
Decision Tree : 0.7962962962962963
```

3.4 Problem Solution fit

3.4.1. Customer Segment(s)

All adults. Especially people who are elder than 40 years and those who are on the verge of getting heart disease due to various factors such as age, obesity, diabetes, stress, etc.

3.4.2. Jobs-To-Be-Done / Problems J&P

To predict and identify the heart disease patient. It is a very useful strategy that was used to control how the model can be utilized to increase the accuracy of the prediction of Heart Attack in each.

3.4.3. Triggers TR

The generation currently living now leads an extremely unhealthy lifestyle. People worry about the sharp rise in mortality from heat -related illnesses. They, therefore, desire to adopt a better lifestyle.

3.4.4. Emotions: Before / After EM

People frequently worry that their health will decline. They suffer unneeded tension and emotional breakdowns as a result of this. Our prediction system would enable them to keep track of their health independently and assist them in overcoming their erroneous concerns.

3.4.5. Available Solutions

EDA: Exploratory data analysis is the key step for getting meaningful results.

Pros: Improve understanding of variables by extracting averages, mean, minimum, and maximum values, etc. Discover errors, outliers, and missing values in the data. Identify patterns by visualizing data in graphs such as box plots, scatter plots, and histograms.

Cons: Exploratory research comes with disadvantages that include offering inconclusive results, lack of standardized analysis, a small sample population, and outdated information that can adversely affect the authenticity of the information.

3.4.6. Customer Constraints

The patient's medical status must be continuously monitored. Unpredictability could lead to inaccurate results. The patient must be genuine about the periodic readings they record. The process could consume the internet and could be slightly expensive.

3.4.7. Behaviour

To solve their problem, a suitable application must be available. To effectively diagnose the situation for their present health status, appropriate information such as age, weight, current symptoms, and cholesterol should be provided.

3.4.8. Channels of Behaviour

3.4.8.1 ONLINE • Data Collected from offline devices is used in this application in order to visualize and predict heart diseases

3.4.8.2 OFFLINE • ECG • Blood Sugar Level • Blood Pressure • Cholesterol

3.4.9. Problem Root Cause

The risk of heart disease is influenced by a number of variables, including smoking, body cholesterol, family history of the disease, obesity, high blood pressure, and inactivity.

3.4.10. Your Solution

We classified heart disease using python and pandas operations for the data taken from the repository after analyzing the outcomes from the existing approaches. It offers a simple-to-understand visual depiction of the dataset, the working environment, and the process of developing predictive analytics.

The machine learning (ML) process begins with a data pre processing phase, which is followed by feature selection based on data cleaning, classification, and modeling performance evaluation. The accuracy of the outcome is increased using the Naive Bayes approach. Unsupervised machine learning is a branch of data analytics that is used for the prediction of heart disease. Unsupervised machine learning techniques include K-means clustering. Unsupervised algorithms often forecast the intended result without reference to any values. The K-means clustering technique clusters the data so that there is a high intra-class similarity and a low inter-class similarity. The sum of squares' distance from the cluster centroid is reduced by this algorithm. The program creates k clusters with a centroid out of the data. K-means iteratively locates the center which minimizes the separation between the cluster's individual points and its cecenterThe k-means clustering technique is demonstrated in the following flow chart.

This section presents the findings of the data analysis conducted to find the necessary hidden patterns for forecasting cardiac illnesses. Age, the type of chest pain, blood pressure, blood sugar level, resting ECG, heart rate, the four different types of chest pain, and exercise-induced angina are the variables here taken into account to predict heart disease. Pre-processing the heart disease dataset efficiently involves removing irrelevant records and assigning values to tuples that are missing. The K-means technique is then used to put together the pre-processed heart disease data set. In this article, four different types of heart diseases-asymptomatic pain, atypical angina pain, non-anginal pain, and non-anginal pain-are explored. Visualization of data Data visualization is a crucial phase in the data science process that enables teams and individuals to communicate data to coworkers and decision-makers more effectively. Teams that oversee reporting systems frequently use predefined template views to keep an eye on efficiency. However, performance dashboards aren't the only applications for data visualization. For instance, while text mining unstructured data, an analyst might employ a word cloud to identify important ideas, patterns, and undiscovered connections. As an alternative, they can show the connections between things in a knowledge graph using a graph structure. It's crucial to keep in mind that there are numerous methods to represent various sorts of data, and that this is a set of abilities that should go beyond your core analytics team. In order to make the dataset we are working with easier to understand, we will first take a look at data Wrangling.It would enable us to make better use of the data. We have to import pandas, matplotlib and seaborn. We can now conduct exploratory data analysis after finishing data wrangling. A timely diagnosis of heart disease (HD), one of the most prevalent diseases today, is essential for many healthcare professionals to protect their patients from the condition and save lives. To accurately classify and/or predict HD cases with few variables, a comparison analysis of various classifiers can be done for the classification of the heart disease dataset. Accurate decision-making and ideal therapy are needed to address cardiac risk. Five machine learning models were utilized in a Canadian study to examine 1-month mortality among hospitalised patients with congestive heart failure. Several tests, including auscultation, blood pressure, cholesterol, ECG, and blood sugar, are carried out prior to the diagnosis of a condition. These tests assist in identifying the patient's medication requirements. In this work, the predictive accuracy of various machine learning methods is investigated to calculate cardiovascular risk.

4. REQUIREMENT ANALYSIS:

4.1 Functional Requirements:

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

User Registration Registration through Form Registration through Gmail Registration through Google

FR NO.	Functional Requireme (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gm Registration through Google
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Health Condition	• Cholesterol • Diabetes • Blood Pressure • Heart Rate • Oxygen Level • Obesity • Medical history
FR-4	User Symptoms	• Increased Heart rate • Difficulty in breathing • Chest, arm and Shoulder Pain • Precipitation

4.2 Non-Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to another.

They are also called non-behavioral requirements. The system should be described by non-functional requirements. behave in a practical manner and impose limitations. This sort is known as the system's quality of requirements attributes. Features like performance, security, and usability, compatibility is not system feature; rather, they an essential quality.

We are unable to separate ourselves from the entire setup. To execute them, write a specific line of code. Any of the customer's desired characteristics are listed in the specification. We can only mention the requirements that are suitable for our project. They basically deal with issues like

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

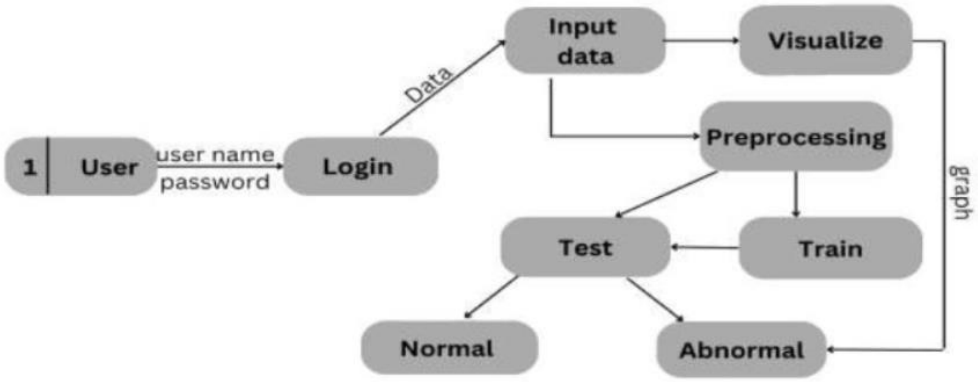
FR NO	Non-Functional Requirement (Epic)	Description
NFR-1	Usability	People can predict their health conditions at any time even when the doctor is not available. It is user friendly
NFR-2	Security	The entire health report of the patient is completely authenticated.
NFR-3	Reliability	Since, we use the Naïve Bayes algorithm we can get of accurate results

NFR-4	Performance	It provides us with 98% Accuracy and also give us faster results.
NFR-5	Availability	It is available at any place and at any time where there is a decent bandwidth.
NFR-6	Scalability	Our system could perform well and maintain its performance even if it is tested by large operational demands

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**.**



FLOW:

In the application,

1) the user creates an account.
2) The user enters the medical information in the dashboard.
3) The user can explore trend visualizations for his or her medical records using the learned dataset as graphs and charts.
4) In the dashboard, the user can see the accuracy of the probability of developing heart disease.

5.2 Solution & Technical Architecture

Solution Architecture:

The Software Architecture is followed by the following steps:

1. The patient first registers by providing certain parameters.
2. That registered data is collected in a database using machine learning techniques similar to data collection techniques, and when he goes to check on his health, the collected values or data that has been stored in the database is extracted.
3. Classification is done using some feature extraction methods when data is extracted, it goes through certain processes, and as a result, a disease is created.
4. The predicted data and a report are produced.

This is an overview of the machine-learning-based heart disease prediction system.

Solution Architecture Diagram:

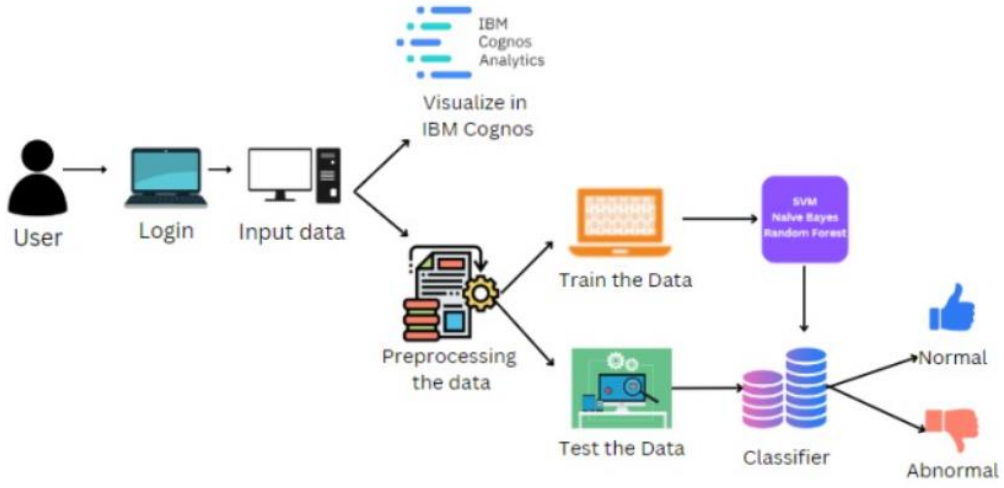


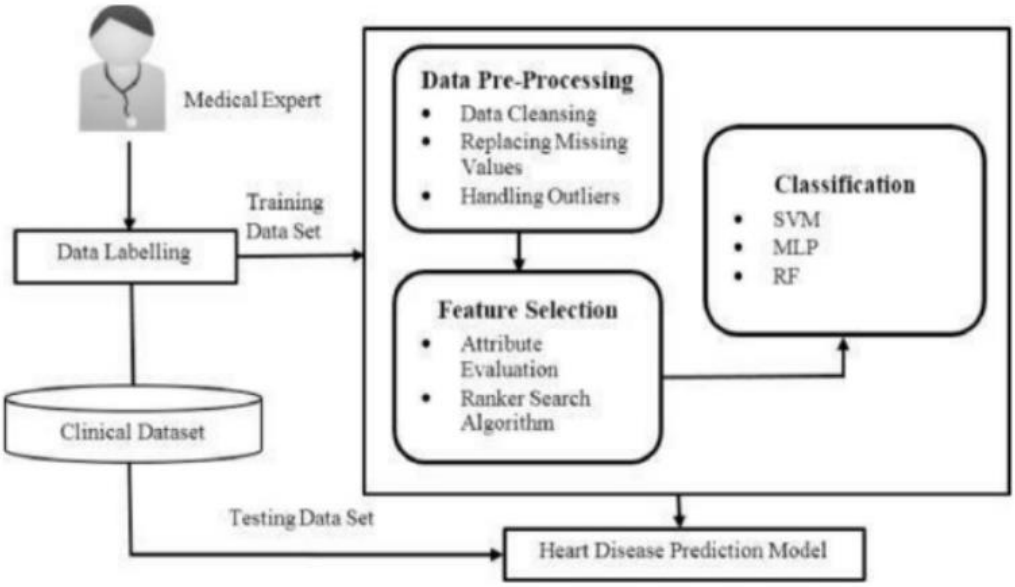
Figure 1: Architecture and data flow of Visualizing and Predicting Heart Diseases with an Interactive Dash Board

Technical Architecture:

1. Patient database: The following characteristics make up the dataset: age, chest pain, blood pressure, cholesterol, diabetes, ECG, heart rate, physical activity, slope, thalassemia, ca, and sex. 'num' will be used to specify the last anticipated attribute.

2. Preprocessing: The knowledge discovery method includes a crucial stage called preprocessing. Real-world data typically exhibit noise and inconsistency. These difficulties are overcome with the use of data processing techniques like data cleaning, etc. Normalization of the dataset aids in data classification, which further improves the data's ability to flow smoothly and produce effective results for algorithms. The normalize function is used to do normalizing. this aids in dividing the data into different classes. Following that, a "num" variable will be created and used to store the predicted characteristic.

3. Testing the model: Testing will be done to ascertain whether the trained model is producing the required results. The '.csv' file will be retrieved when the data is being input for testing so that it may be cross-checked and then compared before the results of the newly entered data are generated. The user will input values of his choosing to the required attributes by how the model is trained using the dataset, and the results will be generated as to whether there is a risk of heart disease or not.



5.3 User Stories:

Story 1:

As an aging individual, I want an application so that I could predict my cardiac health. Let's break this down one step further, As the user is an aging individual, we are building a heart disease-predicting application that enables the user to predict their cardiac health immediately within a few seconds. The app has the user login and signup for the authentication of information, and it uses the Logistic Regression algorithm to predict the result. We have visualized the user's query for their requirement only concerning the Age and Cholesterol of the user. The prediction gives a result if the disease could be present or not. As a working person, I want an application so that I can predict my own cardiac health even while I am at work.

Story 2:

When we address this user issue, As the user is a working person, our heart disease predicting application would enable the user to predict his health just by simply typing the values into the boxes given. The process is very simple, it takes only a few seconds to view the results of the process. The app has the user login and signup for the authentication of information, and it uses the Logistic Regression algorithm to predict the result. The users can customize their range of values and can visualize the entire result in a graphical representation. It allows the user to have a clear knowledge of what must be done and what to be neglected.

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

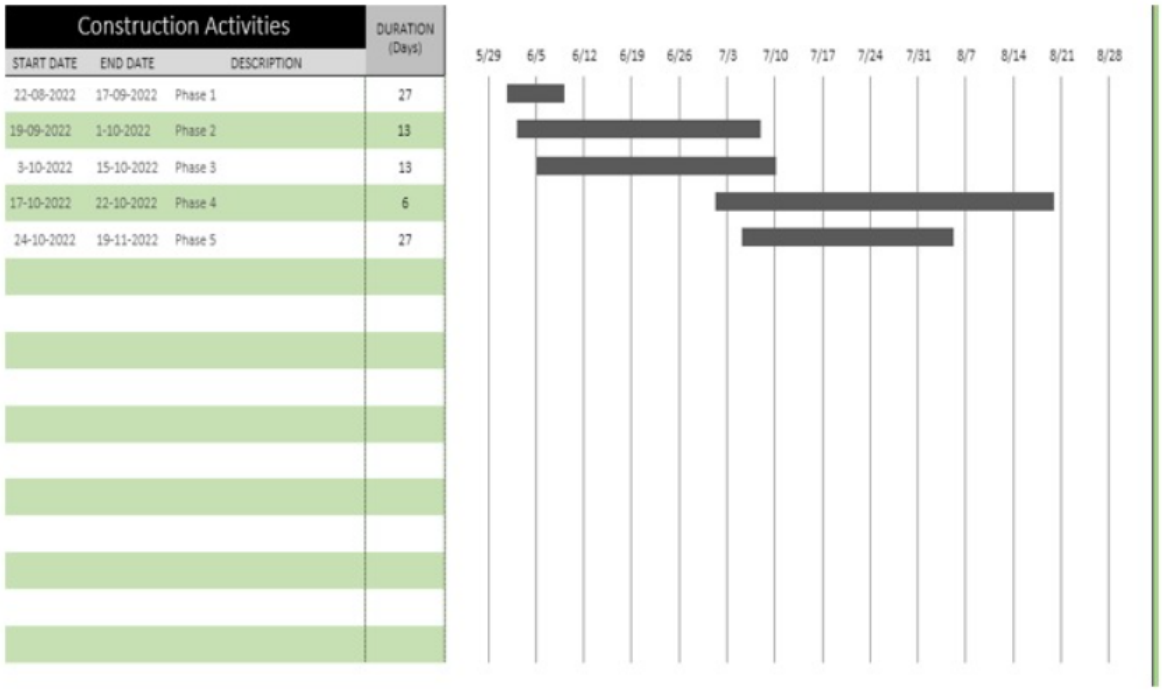
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	2	High	Srishti. R, Jenefa Regina Mary. J
Sprint-1	Mail Confirmation	USN-2	As a user, I will receive a confirmation email once I have registered for application	1	High	Jenefa Regina Mary. J, Harini. M
Sprint-2	Cloud Login	USN-3	As a user, I can register for the application through Facebook	2	Low	Harini. M, Jessica Tiffany. D
Sprint-1	Mail Registration	USN-4	As a user, I can register for the application through Gmail	2	Medium	Jessica Tiffany. D, Jenefa Regina Mary.J
Sprint-1	Login to Dashboard	USN-5	As a user, I can log into the application by entering email & password	1	High	Jenefa Regina Mary.J, Srishti. R

6.2 Sprint Delivery Schedule

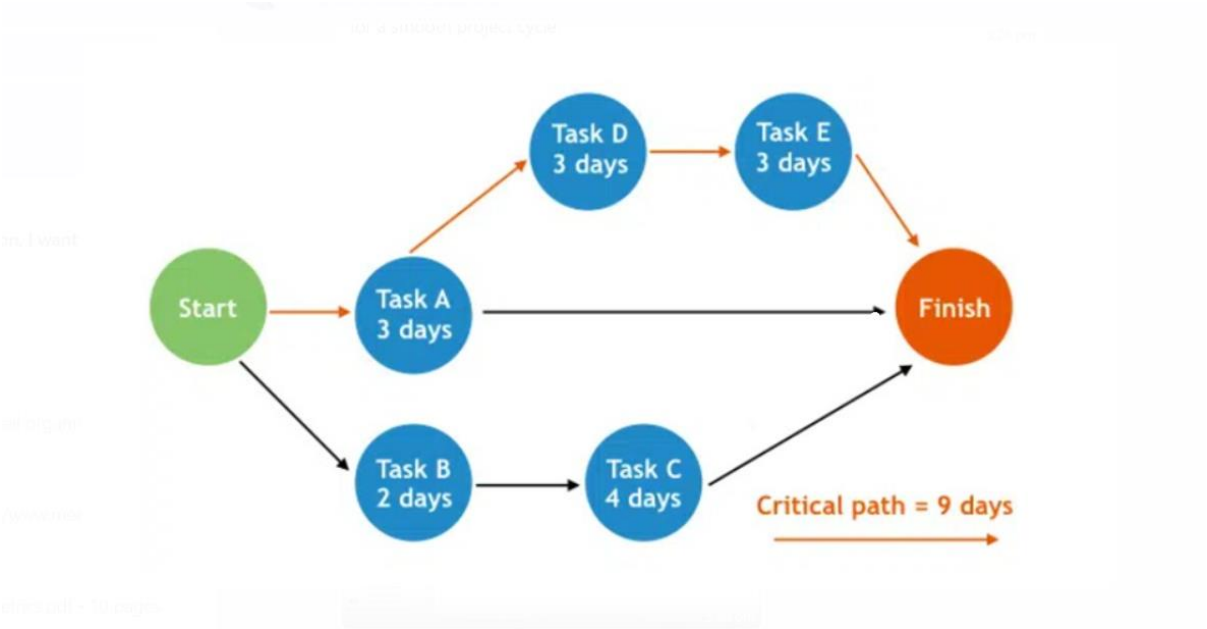
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	6 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Gantt Charts:

A Gantt chart is a diagram that shows all the jobs that are planned to be completed late for your project. Plans for projects of all sizes and kinds are made using them.



Project management is not a straightforward science, as we all know. It is an intricate synthesis of many different ideas, from strategy to people management, and IT communications to figure crunching. This article will teach us about the most crucial resources available to managers. They are used to plan or schedule tasks for projects, identify critical paths, keep track of progress, and do all other crucial duties required for a smooth project cycle.



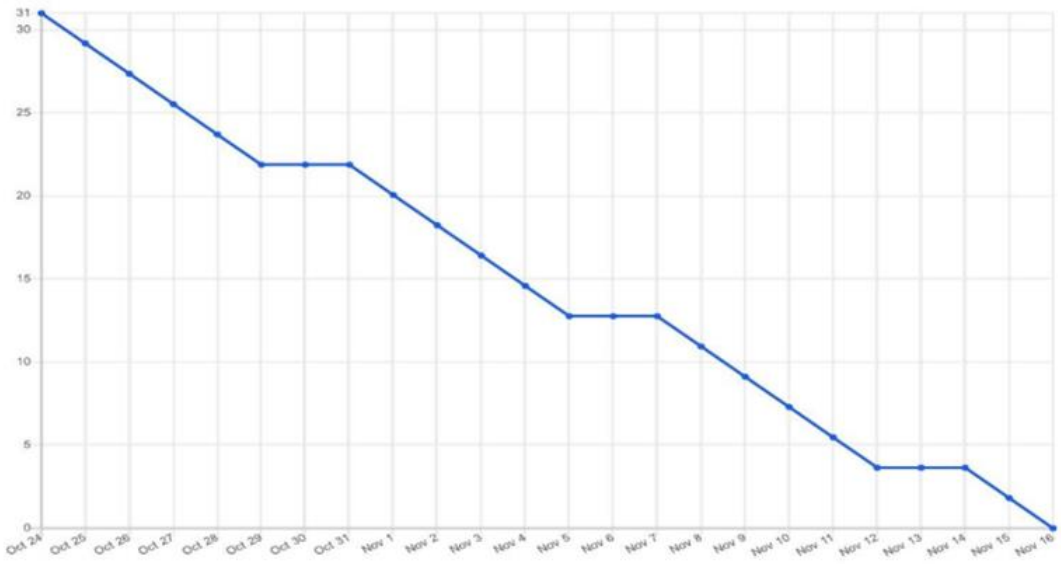
Velocity:

We had a 10-day sprint duration, and the velocity of the team was 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn-down charts can be applied to any project containing measurable progress over time.



6.3. Reports from JIRA:

Jira Software is a member of a family of tools for teams of all sizes to manage their work. Jira was first intended to monitor issues and bugs. But today, Jira has developed into a potent task management platform for a variety of use cases, from agile software creation to requirements and test case management.

Jira software can be used for the following purposes:

- Requirements and Test case management
- In Agile Methodology
- Project Management
- Software Development
- Product Management
- Task Management
- Bug Tracking

Here is a step by step process on how to use Jira software:

Step 1) Open Jira software and navigate to the Jira Home icon

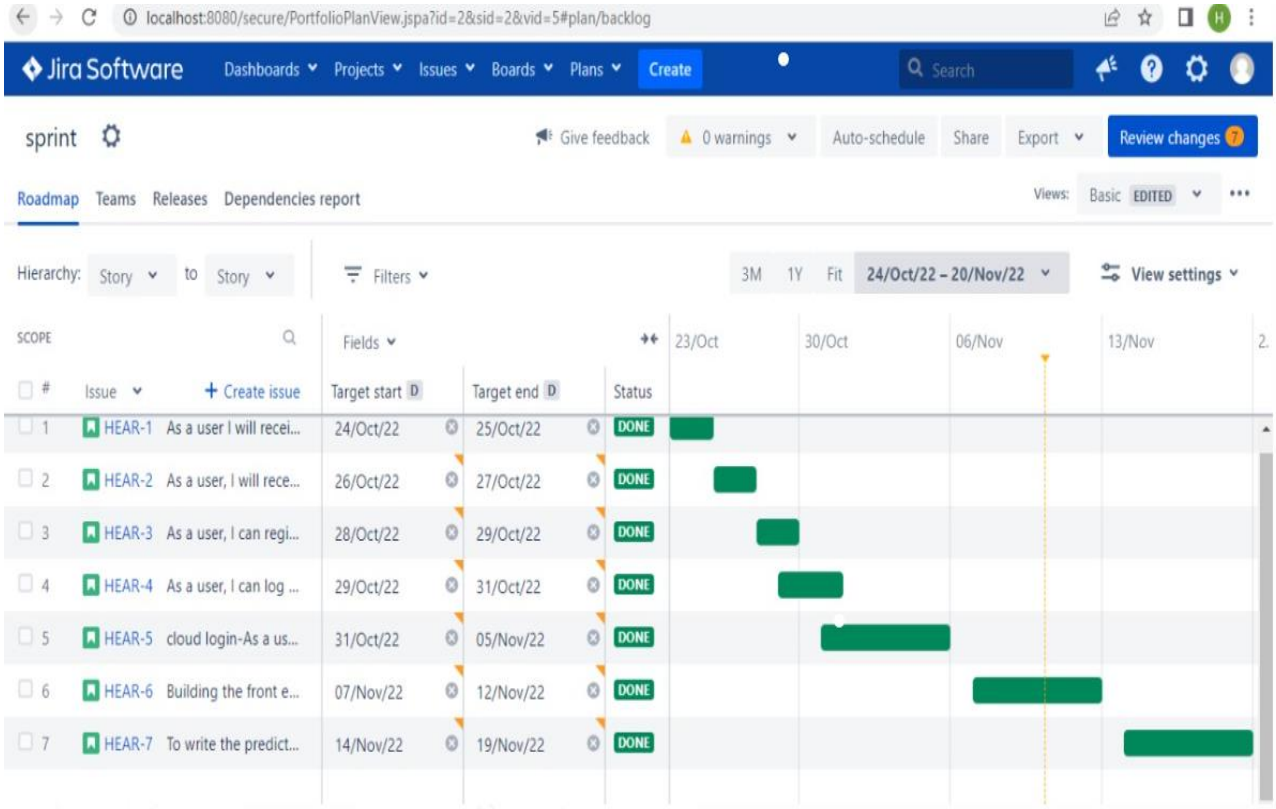
Step 2) Select Create project option

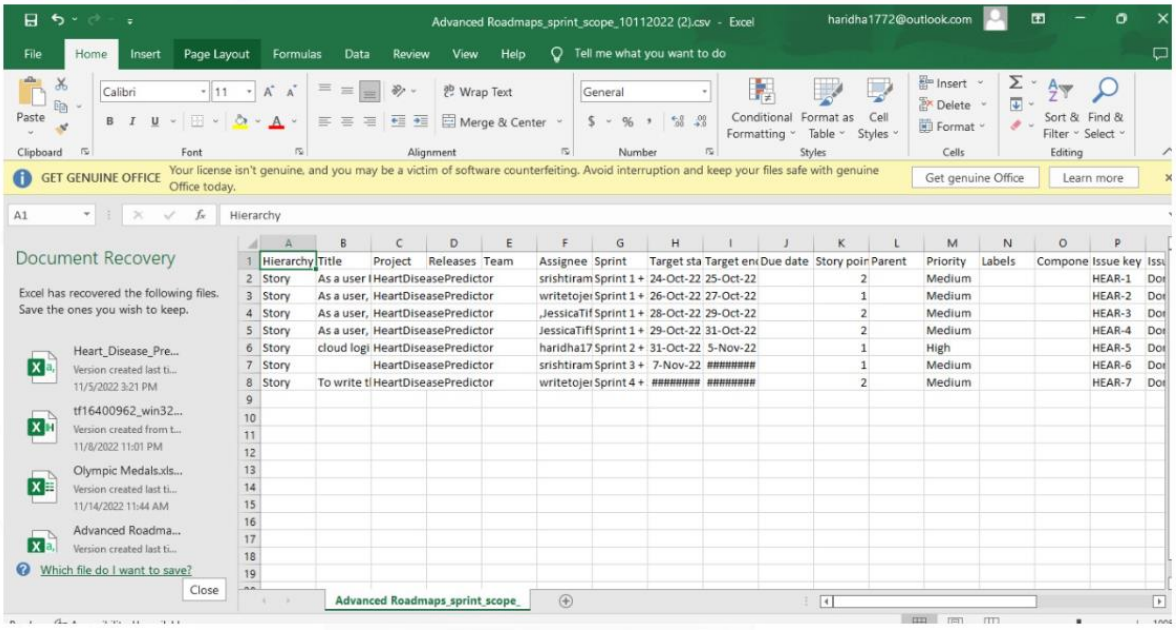
Step 3) Choose a template from the library

Step 4) Set up the columns as per your need from Board settings

Step 5) Create an issue

Step 6) Invite your Team members and start workingg


localhost:8080/secure/PortfolioPlanView.jspa?id=2&sid=2&vid=5#plan/backlog
Jira Software
Dashboards
Projects
Issues
Boards
Plans
Create
Search
sprint
Give feedback
0 warnings
Auto-schedule
Share
Export
Review changes
Roadmap
Teams
Releases
Dependencies report
Views:
Basic
EDITED
Hierarchy:
Story
to
Story
Filters
3M
1Y
Fit
24/Oct/22 – 20/Nov/22
View settings
SCOPE
Fields
23/Oct
30/Oct
06/Nov
13/Nov
Issue
Create issue
Target start
Target end
Status
1 HEAR-1 As a user I will recei... 24/Oct/22 25/Oct/22 DONE
2 HEAR-2 As a user, I will rece... 26/Oct/22 27/Oct/22 DONE
3 HEAR-3 As a user, I can regi... 28/Oct/22 29/Oct/22 DONE
4 HEAR-4 As a user, I can log ... 29/Oct/22 31/Oct/22 DONE
5 HEAR-5 cloud login-As a us... 31/Oct/22 05/Nov/22 DONE
6 HEAR-6 Building the front e... 07/Nov/22 12/Nov/22 DONE
7 HEAR-7 To write the predict... 14/Nov/22 19/Nov/22 DONE


Advanced Roadmaps_sprint_scope_10112022 (2).csv - Excel
haridha1772@outlook.com
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Help
Tell me what you want to do
GET GENUINE OFFICE
Your license isn't genuine, and you may be a victim of software counterfeiting. Avoid interruption and keep your files safe with genuine Office today.
Get genuine Office
Learn more
Document Recovery
Excel has recovered the following files. Save the ones you wish to keep.
Heart_Disease_Pre...
tf16400962_win32...
Olympic Medals.xls...
Advanced Roadma...
Which file do I want to save?
Close
Hierarchy Title Project Releases Team Assignee Sprint Target sta Target en Due date Story poir Parent Priority Labels Compone Issue key
Story As a user I HeartDiseasePredictor srishtiram Sprint 1 + 24-Oct-22 25-Oct-22 2 Medium HEAR-1
Story As a user, HeartDiseasePredictor writetojei Sprint 1 + 26-Oct-22 27-Oct-22 1 Medium HEAR-2
Story As a user, HeartDiseasePredictor ,JessicaTif Sprint 1 + 28-Oct-22 29-Oct-22 2 Medium HEAR-3
Story As a user, HeartDiseasePredictor JessicaTif Sprint 1 + 29-Oct-22 31-Oct-22 2 Medium HEAR-4
Story cloud logi HeartDiseasePredictor haridha17 Sprint 2 + 31-Oct-22 5-Nov-22 1 High HEAR-5
Story HeartDiseasePredictor srishtiram Sprint 3 + 7-Nov-22 ######## 1 Medium HEAR-6
Story To write tl HeartDiseasePredictor writetojei Sprint 4 + ######## ######## 2 Medium HEAR-7
Advanced Roadmaps_sprint_scope_

7. CODING & SOLUTIONING

7.1. Feature 1

7.1.1. Login

The login page is a staple of all secure applications and is frequently used online to verify user identity before displaying the secured sections of web apps. Our sign-in and login pages have been developed. Our application is incredibly secure as a result. In order to determine if you are an OS authorised user or not, the login page will verify your identity. If not, you won't be able to access the operating system. A very old procedure that has been in use since operating systems were created is the security and authentication of any programme. Windows is regarded as the operating system with the best security, and it effectively manages security.

Code

```
def login():
        global root2
        root2 = Toplevel(root)
        root2.title("Account Login")
        root2.geometry("450x300")
        root2.config(bg="white")

        global username_verification
        global password_verification
        Label(root2, text='Login Credentials', bd=5,font=('arial', 12,'bold'),relief="groove",
fg="white",
        bg="purple",width=300).pack()
        username_verification = StringVar()
        password_verification = StringVar()
        Label(root2, text="").pack()
        Label(root2, text="Username :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=username_verification).pack()
        Label(root2, text="").pack()
        Label(root2, text="Password :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=password_verification, show="*").pack()
        Label(root2, text="").pack()
        Button(root2, text="Login", bg="blue", fg='white', relief="groove", font=('arial', 12,
'bold'),command=login_verification).pack()
        Label(root2, text="")
```

7.1.2. Signup

You may directly demonstrate to your visitor the advantages of signing up for your list on the Signup page. Any effective landing page almost always includes benefits, and for good reason.Users and organizations can independently register and receive access to your system via the signup page (sometimes referred to as a registration page). Different

signup pages are frequently used depending on the persons and organizations you want to register.

Code

```
def signup():
        global root2
        root2 = Toplevel(root)
        root2.title("Account signup")
        root2.geometry("450x300")
        root2.config(bg="white")
        global username_verification
        global password_verification
        Label(root2, text='Signup Credentials', bd=5,font=('arial', 12, 'bold'),
       relief="groove", fg="white",
        bg="blue",width=300).pack()
        username_verification = StringVar()
        password_verification = StringVar()
        Label(root2, text="").pack()
        Label(root2, text="Username :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=username_verification).pack()
        Label(root2, text="").pack()
        Label(root2, text="Password :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=password_verification, show="*").pack()
        Label(root2, text="").pack()
        Button(root2, text="Login", bg="blue", fg='white', relief="groove", font=('arial',
       12, 'bold'),command=login).pack()
        Label(root2, text="")
        sql = "INSERT INTO proj (user_name, password) VALUES (%s, %s)"
        val = ('?','?')
        cursordb.execute(sql, val)
```

7.2. Feature 2

7.2.2. Prediction page

Predictive analytics in healthcare can raise the standard of care, gather more clinical data for individualized treatment, and correctly identify each patient's medical condition.Based on historical data, machine learning model predictions enable businesses to make extremely precise assumptions about the most likely outcomes of a question. These assumptions can be made about a variety of topics, including the likelihood of customer churn, potential fraudulent activity, and more.

```
def pred():
   Age1=(Age.get())
   Sex1=(Sex.get())
```

```
Chestpaintype1=(Chestpaintype.get())
FBSover1=(FBSover.get())
MaxHR1=(MaxHR.get())
Exerciseangina1=(Exerciseangina.get())
Cholesterol1=(Cholesterol.get())
BP1=(BP.get())
EKGresults1=(EKGresults.get())
Thallium1=(Thallium.get())
STdepression1=(STdepression.get())
Numberofvesselsfluro1=(Numberofvesselsfluro.get())
                SlopeofST1=(SlopeofST.get())
                 print(Age1,Sex1,Chestpaintype1,FBSover1,MaxHR1,Exerciseangina
1,Cholesterol1 , BP1 , EKGresults1 , Thallium1, STdepression1,
Numberofvesselsfluro1, SlopeofST1)
                if Age1 or Sex1 or Chestpaintype1 or FBSover or MaxHR1 or
Exerciseangina1 or Cholesterol1 or BP1 or EKGresults1 or Thallium1 or
STdepression1 or Numberofvesselsfluro1 or SlopeofST1 :
                  p=model.predict([[Age1,Sex1,Chestpaintype1,FBSover1,MaxHR1,
Exerciseangina1,Cholesterol1,BP1,EKGresults1,Thallium1,STdepression1,Numberofv
esselsfluro1,SlopeofST1]])
                  p2=model.predict_proba([[Age1,Sex1,Chestpaintype1,FBSover1,M
axHR1,Exerciseangina1,Cholesterol1,BP1,EKGresults1,Thallium1,STdepression1,Nu
mberofvesselsfluro1,SlopeofST1]])
      print(p[0])
      p3=round(max(p2[0])*100,2)
      p4=str(p3)+" % "+str(p[0])
      lb6.configure(text=p4)

       txt1.delete(first=0,last=100)
       txt2.delete(first=0,last=100)
       txt3.delete(first=0,last=100)
       txt4.delete(first=0,last=100)
       txt5.delete(first=0,last=100)
       txt6.delete(first=0,last=100)
       txt7.delete(first=0,last=100)
       txt8.delete(first=0,last=100)
       txt9.delete(first=0,last=100)
       txt10.delete(first=0,last=100)
       txt11.delete(first=0,last=100)
       txt12.delete(first=0,last=100)
       txt13.delete(first=0,last=100)
```

8. TESTING

Software testing can take many distinct forms, each with its own goals and techniques:

Acceptance testing: Checking that the entire system operates as planned. Verifying that various software elements or functionalities work together is known as integration testing.

Unit testing: Verifying that each piece of software works as it should. The smallest piece of an application that can be tested is called a unit.
Functional testing: is the process of verifying that programmes work as intended by simulating real-world situations. Functions are frequently checked via black-box testing.

Performance testing: Analysing how well the software works under various workloads. For instance, performance under actual load circumstances is assessed via load testing.

Testing for functionality degradation or breakage as a result of new features. When there isn't enough time for a comprehensive regression test, sanity testing can be performed to quickly examine menus, functions, and instructions.

Stress testing: Determining the amount of strain the system can withstand before failing. recognised as a subset of non-functional testing.

Usability testing: Validating how well a customer can use a system or web application to complete a task.

8.1. Test Cases

Verify users are able to see the Login/Signup popup when the user clicks on the My account button.

1. Click login
2. Check if the user is registered or not in the database
3. If the user is registered, allow login.

Verify the user is able to log into the application with Invalid credentials

1. Click login
2. Check if the user is registered or not in the database
3. If the user is registered allow login.
4. If the user is not registered it shows invalid

Verify user if the user is able to sign up

1. Click the signup button
2. Type the user name and password you want to set.
3. Sign-up
4. Login using the signed username

Login verification page

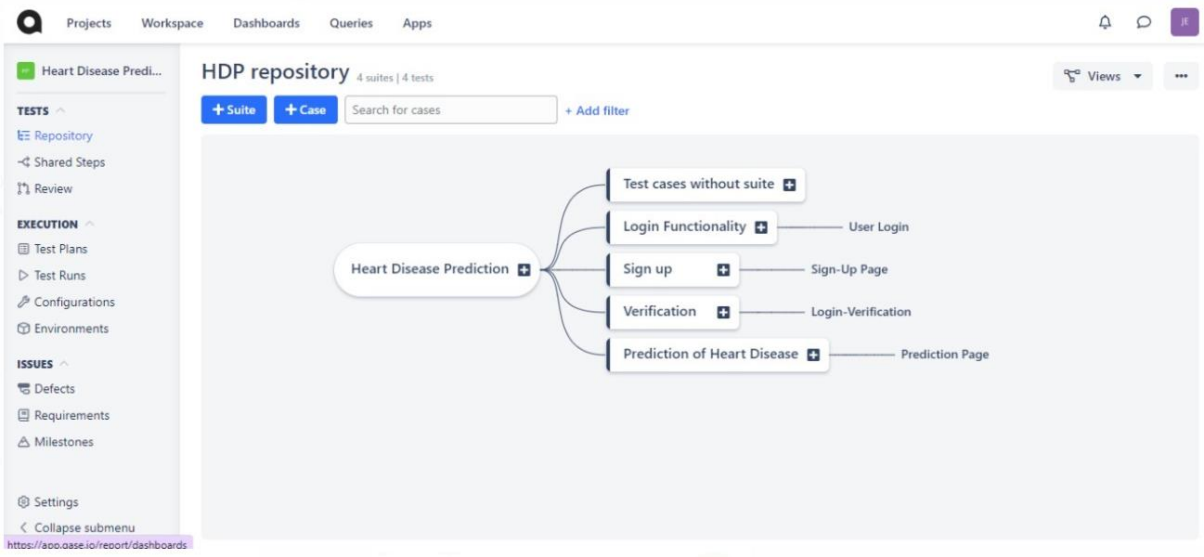
1. Click login
2. Check if the user is registered or not in the database.
3. If the user is registered allows login.
4. If the user is not registered it does not allow login.

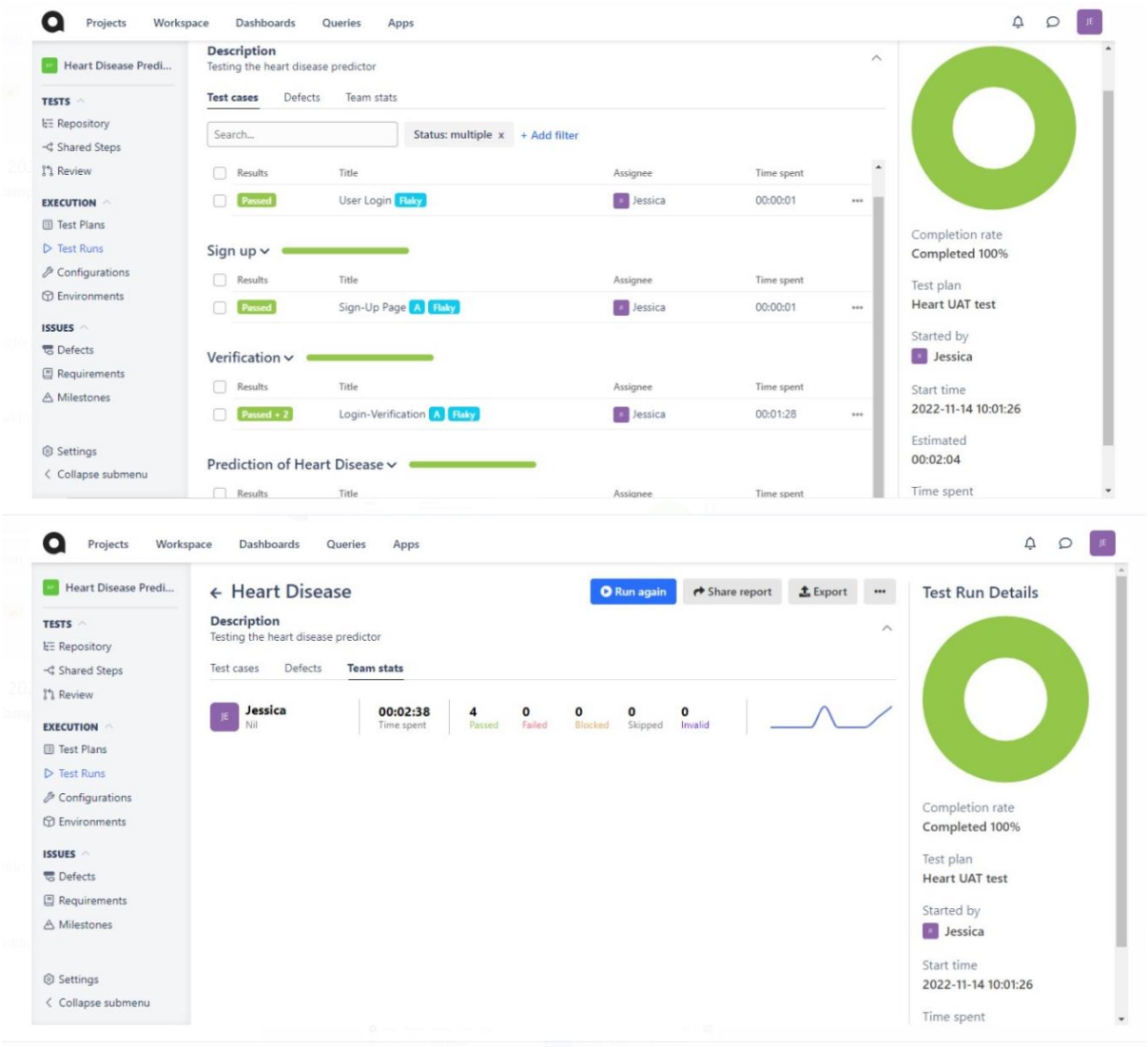
Verify if the user is able to predict the heart disease

1. Type the values in the blank boxes
2. click predict button
3. Check if the result is valid

8.2 User Acceptance Testing

Before deploying the software programme to a production environment, the end user or client does a sort of testing known as user acceptance testing, or UAT. After functional, integration, and system testing are complete, UAT is carried out as the last stage of testing. UAT's primary goal is to verify the entire business process. It doesn't concentrate on minor typos, misspellings, or system testing. User Acceptance Testing is performed using production-like data setup in a separate testing environment. It will resemble black box testing and involve two or more end users.





UAT is performed by –

1. Client
2. End users

9. RESULTS

9.1. Performance Metrics

Metrics are measurements and parameters obtained throughout the quality assurance procedure. They may make reference to several test types. As you might have guessed, performance testing data gives you the ability to evaluate the efficiency of performance testing. Alternatively said, these measurements demonstrate how well software reacts to user scenarios and manages user flow in real time.
The following two categories of data are appropriate:

1. Measurements are data that are kept track of when testing, such as how long it takes to react to a request.
2. Metrics, which include various types of percentages, average indicators, and other metrics, are computations performed with the aid of certain formulas.

Accuracy Percentages:

The number of wrong predictions on the test set as a whole divided by all of the test set predictions yields the error rate. Since accuracy and error rate are complementary quantities, we can always compute one from the other.

Accuracy = 1 – Error Rate
Error rate=1-Accuracy

Logistic Regression: 0.82
KNN: 0.61
Naïve Bayes: 0.77
Decision Tree: 0.79

Response Time:

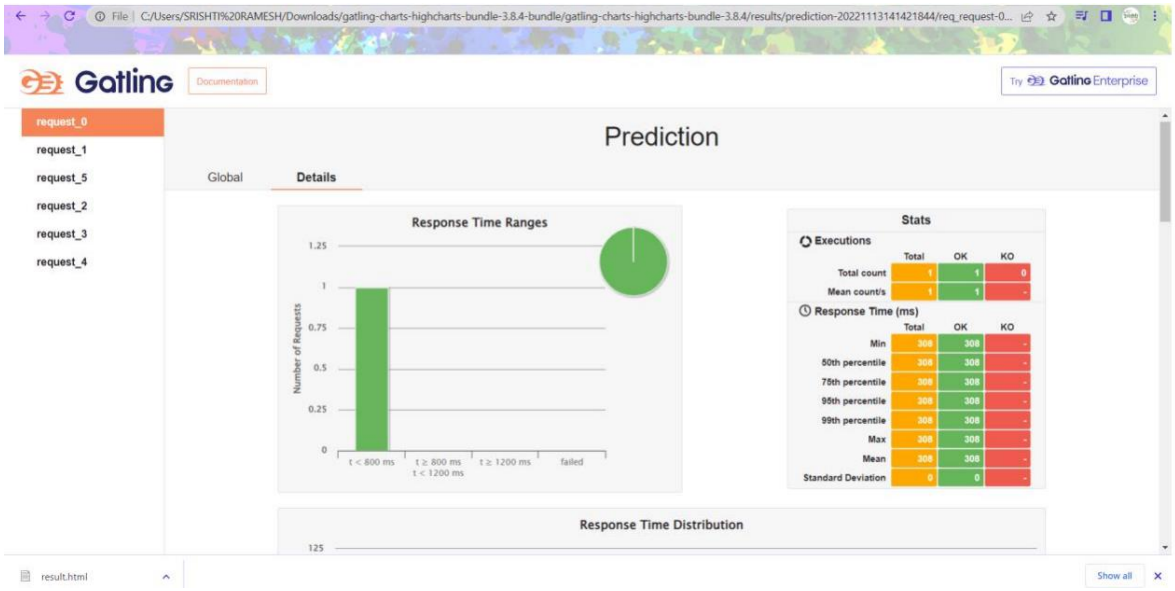
The response time is not too long as our project as we have used real time data analysis. So, once the user enters his/her data in the Heart Disease prediction phase then the data will immediately be displayed so the response time is very less.

Requests per second:

An HTTP request is created and sent to a server by a client application. The client receives a request from the server, which processes it, produces a response, and sends it back. The measure we are interested in is requests per second, which is the total number of consistent requests per second (RPS). These can be requests for any type of data source, including XML documents, HTML files, multimedia files, and JavaScript libraries.

User transactions:

A user transaction is a series of user operations performed through a software interface. You may determine how well the system survived the load testing by comparing the number of transactions per second or actual transaction time with the projected number of transactions per second.



Virtual users per unit of time:

This statistic also aids in determining whether the performance of a software product satisfies the stated requirements. A QA team finds it useful to predict average load as well as programme behavior under various load levels.

Error rate:

The ratio of incorrect to correct replies over time is used to calculate this measure. Percentages are used to calculate the findings. When software load surpasses its capacity, mistakes frequently happen.

10. ADVANTAGES & DISADVANTAGES

10.1. Advantages

a. The proposed work predicts the chances of Heart Disease and classifies patient's risk level
b. It is implementing different data mining techniques such as Naive Bayes, Decision Tree, Logistic Regression and Random Forest.
c. User friendly

10.2. Disadvantages

a. Data analytics techniques do not help to. provide effective decision making.
b. Cannot handle enormous datasets
c. Prediction of cardiovascular disease results is not accurate

11. CONCLUSION

The long-term preservation of human life and the early identification of irregularities in heart problems will benefit from the identification of the processing of raw healthcare data related to the heart. In this study, raw data was processed using machine learning techniques to produce a brand-new understanding of cardiac disease. In the medical field, heart disease prediction is difficult and crucial. The death rate, however, can be significantly reduced if the disease is identified in its early stages and preventative measures are put in place as soon as feasible. To move the investigations from simply theoretical frameworks and simulations to actual datasets, further elaboration of this study is extremely desirable. The model's ability to be employed to increase the precision of heart attack prediction in any individual was regulated using a very helpful technique. When compared to the previously employed classifiers, such as naive bayes, etc., the proposed model's strength was quite satisfying. It was able to predict signs of having a heart illness in a specific individual by applying KNN and Logistic Regression, which demonstrated good accuracy. Therefore, by utilizing the provided model to determine the likelihood that the classifier will correctly and reliably detect the heart illness, a large amount of pressure has been reduced. The Given heart disease prediction system improves and lowers the cost of medical care. This project gives us significant knowledge that can help us predict the patients with heart disease It is implemented on the .pynb format.

12. FUTURE SCOPE

This study discusses the issue of constricting and summarizing various data mining strategies utilized in the field of medical forecasting. For intelligent and successful heart attack prediction via data mining, the emphasis is on combining various methods and combinations of numerous target attributes. Significantly, 15 attributes are specified for predicting heart attacks, and using simple data mining techniques, other approaches, including ANN, time series, clustering and association rules, soft computing approaches, etc., can also be included. The results of predictive data mining on the same dataset show that Decision Tree outperforms and, occasionally, Bayesian classification has accuracy levels comparable to those of decision tree, but other predictive methods, such as KNN, Neural Networks, and Classification based on clustering, are not performing well. The second finding is that using a genetic algorithm to lower the actual data quantity and obtain the ideal subset of attributes suitable for heart disease prediction increases the decision tree and Bayesian classification's accuracy. For the automation of heart disease prediction, the proposed work can be expanded and improved. Real data from healthcare institutions and agencies must be gathered, and all methods must be compared for the highest level of accuracy**.**

13. APPENDIX

Source Code:

```
def login():
        global root2
        root2 = Toplevel(root)
        root2.title("Account Login")
        root2.geometry("450x300")
        root2.config(bg="white")

        global username_verification
        global password_verification
        Label(root2, text='Login Credentials', bd=5,font=('arial', 12, 'bold'), relief="groove",
       fg="white",
        bg="blue",width=300).pack()
        username_verification = StringVar()
        password_verification = StringVar()
        Label(root2, text="").pack()
```

```
        Label(root2, text="Username :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=username_verification).pack()
        Label(root2, text="").pack()
        Label(root2, text="Password :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=password_verification, show="*").pack()
        Label(root2, text="").pack()
        Button(root2, text="Login", bg="blue", fg='white', relief="groove", font=('arial', 12,
      'bold'),command=login_verification).pack()
        Label(root2, text="")

def signup():
        global root2
        root2 = Toplevel(root)
        root2.title("Account signup")
        root2.geometry("450x300")
        root2.config(bg="white")
        global username_verification
        global password_verification
        Label(root2, text='Signup Credentials', bd=5,font=('arial', 12, 'bold'),
      relief="groove", fg="white",
        bg="blue",width=300).pack()
        username_verification = StringVar()
        password_verification = StringVar()
        Label(root2, text="").pack()
        Label(root2, text="Username :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=username_verification).pack()
        Label(root2, text="").pack()
        Label(root2, text="Password :", fg="black", font=('arial', 12, 'bold')).pack()
        Entry(root2, textvariable=password_verification, show="*").pack()
        Label(root2, text="").pack()
        Button(root2, text="Login", bg="blue", fg='white', relief="groove",
      font=('arial', 12, 'bold'),command=login).pack()
        Label(root2, text="")
        sql = "INSERT INTO heart1 (user_name, password) VALUES (%s, %s)"
        val = ('?','?')
        cursordb.execute(sql, val)
```

```
def logged_destroy():
        logged_message.destroy()
        root2.destroy()

def failed_destroy():
        failed_message.destroy()

def logged():
        global logged_message
        logged_message = Toplevel(root2)
        logged_message.title("Welcome")
        logged_message.geometry("500x100")
        Label(logged_message, text="Login Successfully!... Welcome {}
       ".format(username_verification.get()), fg="green", font="bold").pack()
        Label(logged_message, text="").pack()
        Button(logged_message, text="Prediction page", bg="blue", fg='white',
      relief="groove", font=('arial', 12, 'bold'), command=pred).pack()


def failed():
        global failed_message
        failed_message = Toplevel(root2)
        failed_message.title("Invalid Message")
        failed_message.geometry("500x100")
        Label(failed_message, text="Invalid Username or Password", fg="red",
      font="bold").pack()
        Label(failed_message, text="").pack()
        Button(failed_message,text="Ok", bg="blue", fg='white', relief="groove",
      font=('arial', 12, 'bold'), command=failed_destroy).pack()


def login_verification():
        user_verification = username_verification.get()
        pass_verification = password_verification.get()
        sql = "select * from heart1 where user_name = %s and password = %s"
        cursordb.execute(sql,[(user_verification),(pass_verification)])
        results = cursordb.fetchall()
```

```
        if results:
         for i in results:
          logged()
          break
        else:
         failed()

def Exit():
        wayOut = tkinter.messagebox.askyesno("Login System", "Do you want to
exit the system")
        if wayOut > 0:
          root.destroy()
        return

def main_display():
          global root
          root = Tk()
          root.config(bg="white")
          root.title("Login System")
          root.geometry("500x500")
          Label(root,text='Welcome to Log In System', bd=20, font=('arial', 20,
'bold'), relief="groove", fg="white",
          bg="blue",width=300).pack()
          Label(root,text="").pack()
          Button(root,text='Log In', height="1",width="20", bd=8, font=('arial', 12,
'bold'), relief="groove", fg="white",
          bg="blue",command=login).pack()
          Label(root,text="").pack()
          Button(root,text='Sign up', height="1",width="20", bd=8, font=('arial', 12,
'bold'), relief="groove", fg="white",
          bg="blue",command=signup).pack()
          Label(root,text="").pack()
          Button(root,text='Exit', height="1",width="20", bd=8, font=('arial', 12,
'bold'), relief="groove", fg="white",
          bg="blue",command=Exit).pack()
          Label(root,text="").pack()
```

```
        Label(root,text='SrishJeneJessHari', bd=20, font=('arial', 15, 'bold'),
relief="groove", fg="white",
        bg="blue",width=300).pack()
        Label(root,text="").pack()
```

GitHub & Project Demo Link:

https://github.com/IBM-EPBL/IBM-Project-12009-1659366288