1. INTRODUCTION

1.1. Project Overview

Heart Disease Analysis is a data visualization project designed to analyse and interpret heart disease data using interactive dashboards and storytelling techniques. The application leverages Tableau to transform raw healthcare data into meaningful visual insights that help identify key risk factors such as smoking, obesity, diabetes, physical inactivity, and age-related patterns. By integrating the dashboard with a Flask web application, the system allows users to access and explore heart disease insights through a browser interface in real time.

The project aims to support healthcare professionals, policymakers, and individuals in understanding the major factors contributing to cardiovascular diseases. Instead of manually analysing large datasets, users can rely on interactive visualizations to quickly identify trends, correlations, and high-risk groups. The application is built using Tableau Public for data visualization and Flask for web integration, ensuring accessibility, interactivity, and user-friendly design.

Overall, the Heart Disease Analysis project demonstrates the practical use of business intelligence and data visualization tools in healthcare analytics by providing a structured, interactive, and insightful solution for heart disease data exploration and decision-making.

1.2. Objectives

The main objectives of the Heart Disease Analysis using Tableau and Flask project are:

• To develop an interactive data visualization system to analyze heart disease data using Tableau

• To identify and analyze key risk factors such as smoking, obesity, diabetes, physical inactivity, and age groups contributing to heart disease

• To transform raw healthcare data into meaningful visual insights through dashboards and stories

• To integrate Tableau dashboards with a Flask web application for browser-based accessibility

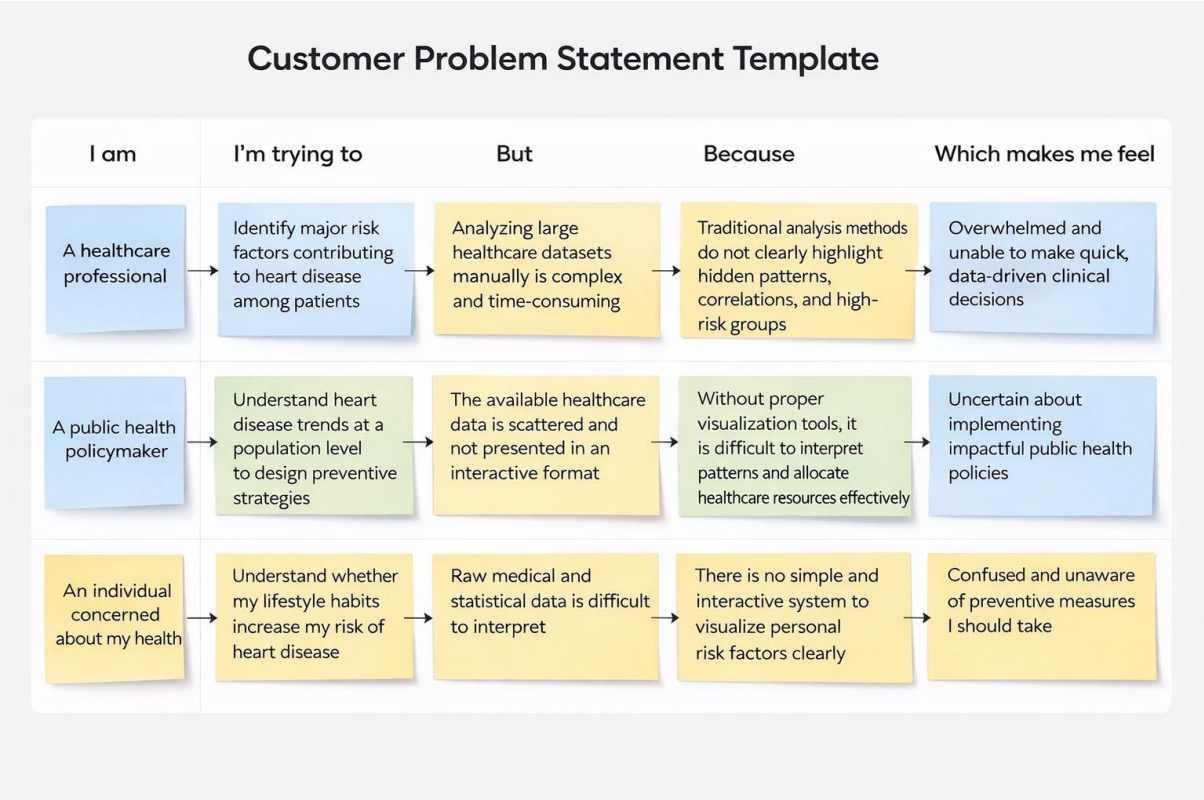
•To provide a simple and user-friendly web interface for exploring heart disease insights

• To enhance understanding of cardiovascular risk patterns through interactive and real-time data visualization

2. Ideation Phase

2.1. Problem Statement

Heart disease is one of the leading causes of death globally, with rising cases attributed to lifestyle factors such as smoking, obesity, diabetes, physical inactivity, and poor dietary habits. Despite the availability of large-scale healthcare datasets, extracting meaningful insights from this data is challenging without proper analytical tools. Traditional data analysis methods are time-consuming and may fail to effectively highlight hidden patterns and correlations. Therefore, there is a need for an interactive data visualization system that can analyze heart disease data and present key risk factors in a clear, structured, and insightful manner to support preventive healthcare decision-making.

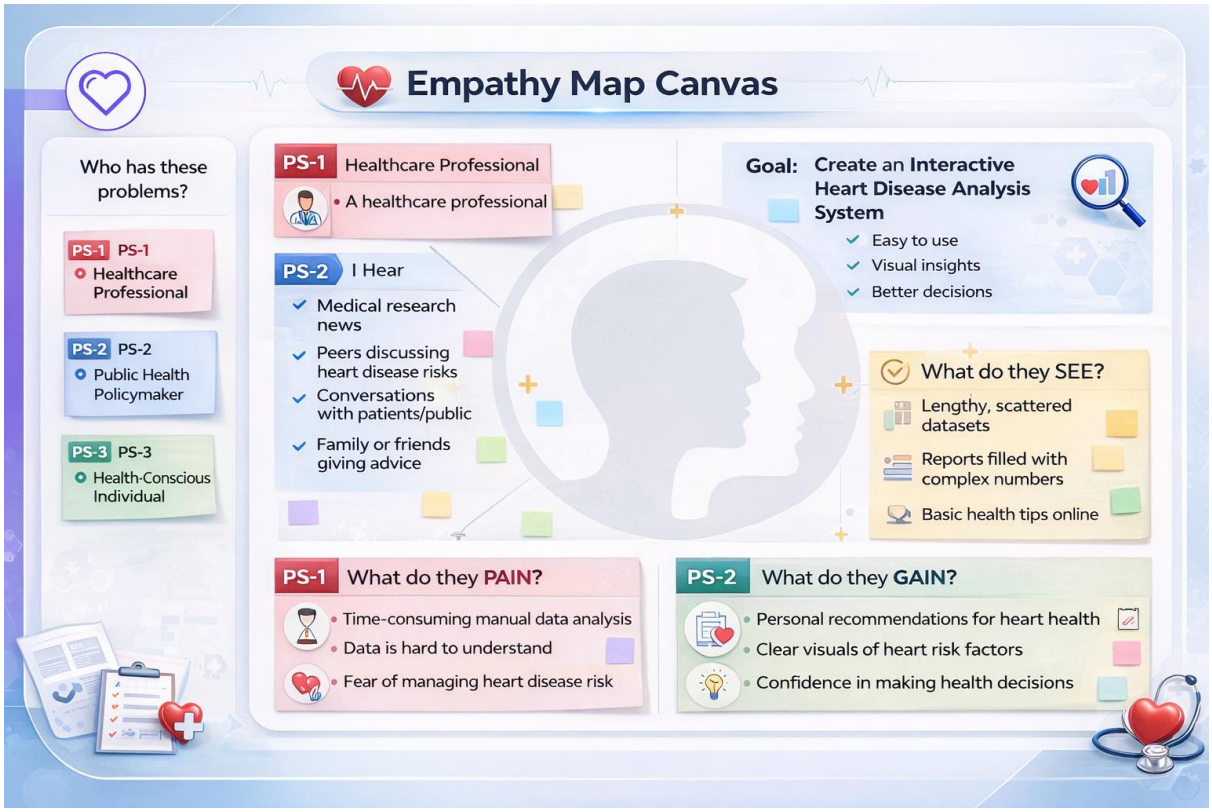


PS	I am	I'm trying to	But	Because	Which makes me feel
PS-1	A healthcare professional	Find heart disease risk factors	Large health data is difficult to analyze	Data is not clearly visualized	Confused and slow in making decisions

PS	I am	I'm trying to	But	Because	Which makes me feel
PS-2	A public health policymaker	Understand heart disease trends	Data is scattered and not interactive	No proper visualization system	Unsure about planning health programs
PS-3	A health-conscious individual	Know my heart disease risk	Medical data is hard to understand	No simple tool to show risk clearly	Worried and unaware of prevention steps

2.2. Empathy Map Canvas

Empathy Map Canvas:



2.3 Brainstorming

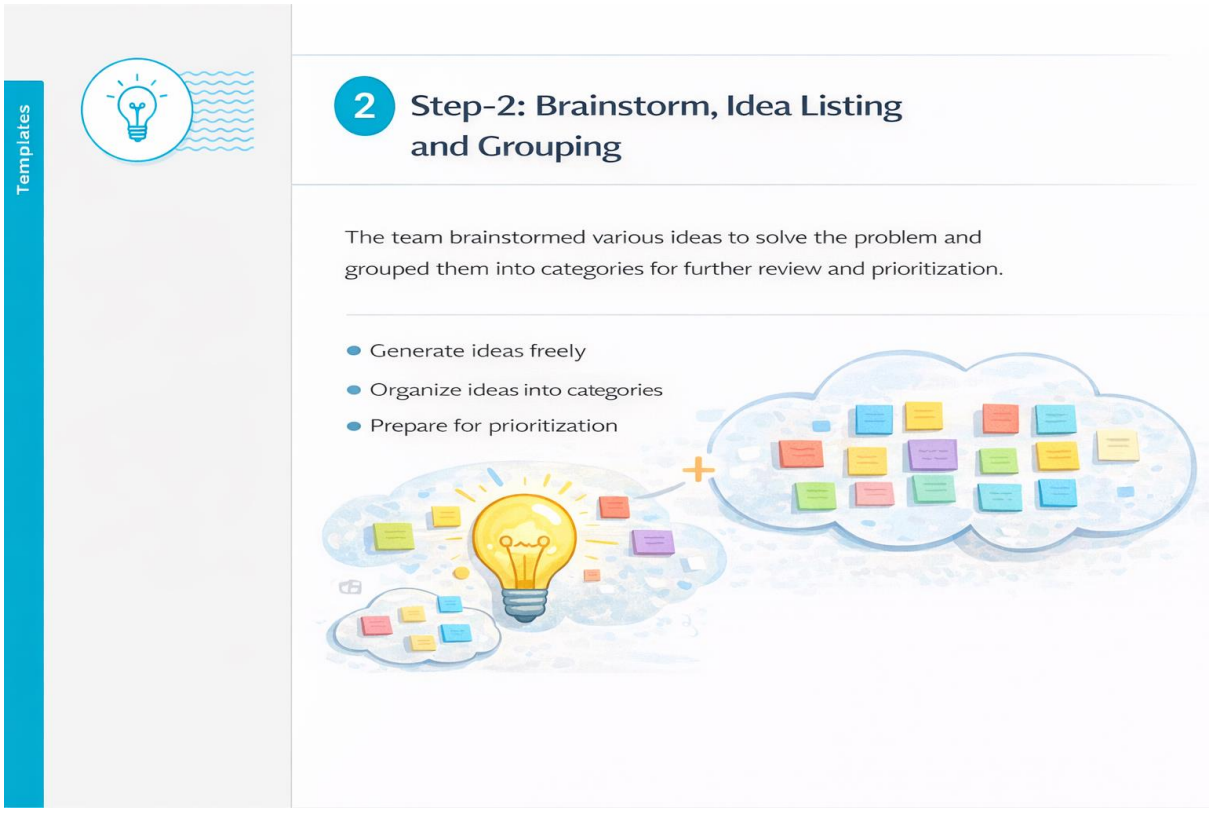
Brainstorm & Idea Prioritization:

Brainstorming provides a collaborative and open environment where team members share ideas to solve the identified problem. In this project, brainstorming was conducted to explore different ways to analyze heart disease data and present meaningful insights through visualization. The main focus was to transform complex healthcare data into simple, interactive, and understandable dashboards.

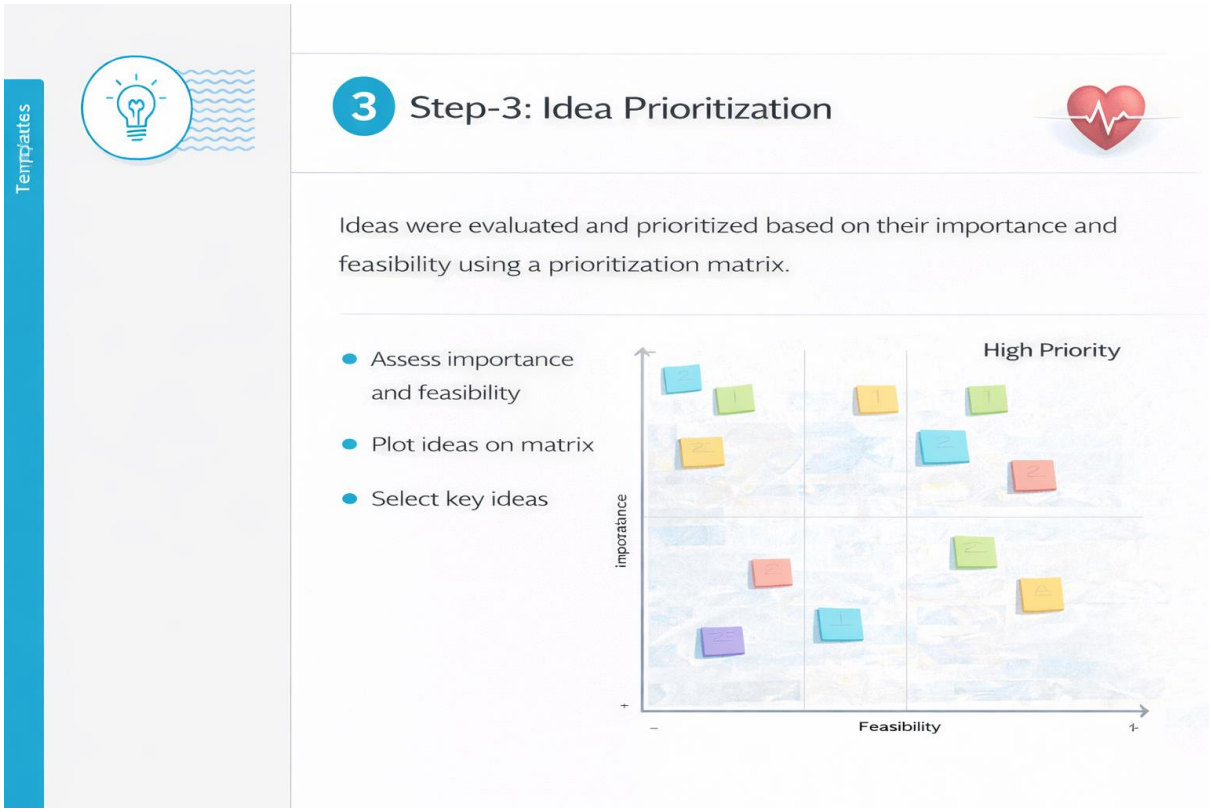
All ideas were discussed freely, and practical solutions were selected based on feasibility, clarity, and impact.

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

Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3. Requirement Analysis

3.1. Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements	Sub Requirements
FR-1	Data Import Module	Import heart disease dataset into Tableau
FR-2	Data Processing	Clean and organize dataset fields
FR-3	Visualization Creation	Create charts for age, BMI, smoking, diabetes, etc.
FR-4	Dashboard Design	Combine multiple visualizations into interactive dashboard
FR-5	Story Development	Create story to explain insights step-by-step
FR-6	Dashboard Publishing	Publish dashboard on Tableau Public
FR-7	Web Integration	Embed Tableau dashboard into Flask application
FR-8	UI Display	Display dashboard properly in web interface

Non-Functional Requirements

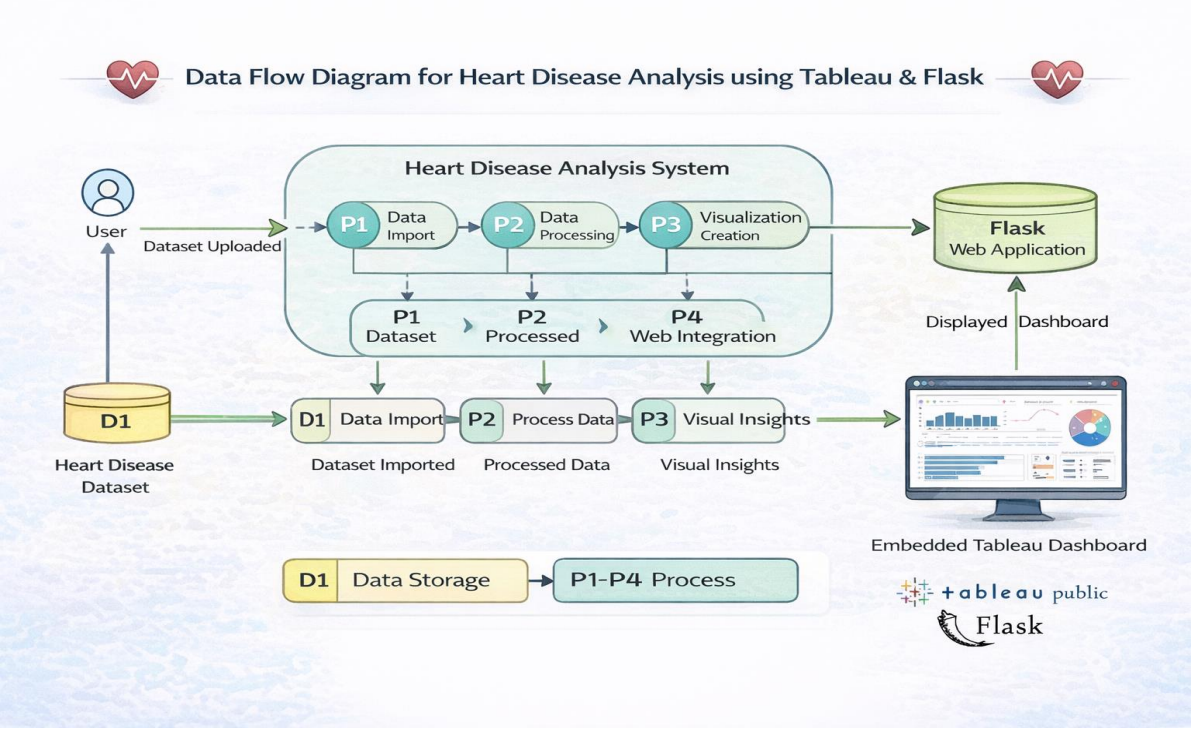
Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The web application must have a simple and user-friendly interface.
NFR-2	Security	The application should not expose sensitive data publicly.
NFR-3	Reliability	The dashboard should load correctly without errors.
NFR-4	Performance	The dashboard should load within a few seconds.
NFR-5	Availability	The published dashboard should be accessible online.
NFR-6	Scalability	The system should support multiple users accessing the dashboard.

3.2. Data Flow Diagram

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 stored.



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Healthcare Professional	Data Visualization	USN-1	As a user, I want to view heart disease data through interactive charts.	I can view multiple visualizations clearly in the dashboard.	High	Sprint-1
Data Analyst	Data Processing	USN-2	As a user, I want the dataset to be cleaned and structured before visualization.	Data is properly processed and organized in Tableau.	High	Sprint-1
Public Health Policymaker	Dashboard Design	USN-3	As a user, I want to explore heart disease trends using filters.	I can apply filters like age, gender, and smoking to analyze trends.	High	Sprint-2
Healthcare Professional	Story Creation	USN-4	As a user, I want to view insights in a story format for better understanding.	I can navigate through story scenes explaining key insights.	Medium	Sprint-2
General User	Web Integration	USN-5	As a user, I want to access the dashboard through a web application.	I can open and view the dashboard in the browser via Flask.	High	Sprint-3
System Administrator	Deployment	USN-6	As a user, I want the dashboard to be published and accessible online.	The dashboard is successfully published on Tableau Public.	Medium	Sprint-3

3.3. Technology Stack

Technical Architecture:

Table-1: Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Web-based interface where users can view the heart disease dashboard	HTML, CSS
2.	Application Logic	Handles routing and rendering of web pages	Flask (Python Web Framework)
3.	Data Source	Heart disease dataset used for analysis	CSV Dataset
4.	Data Processing	Cleaning and organizing dataset for visualization	Tableau
5.	Data Visualization	Creation of charts, dashboards, and stories	Tableau Public
6.	Dashboard Publishing	Publishing dashboard for online access	Tableau Public
7.	Dashboard Integration	Embedding Tableau dashboard into Flask web application	iFrame / Tableau Embed Code
8.	Infrastructure (Server / Local)	Running the web application and hosting dashboard	Flask Local Server / Tableau Public

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Web framework and development tools used	Python, Flask, Tableau Public
2.	Security Implementation	Ensures no sensitive data is exposed publicly	Controlled Data Publishing
3.	Scalable Architecture	Web-based architecture supporting multiple users	Browser-based Access
4.	Availability	Application accessible online after publishing	Tableau Public Hosting
5.	Performance	Fast loading interactive dashboard with filters	Optimized Tableau Visualizations

4. PROJECT DESIGN

4.1. Problem Solution Fit

Problem – Solution Fit Template:

Problem–Solution Fit refers to identifying a real problem faced by the target users and ensuring that the proposed solution effectively addresses that problem. In this project, healthcare professionals and policymakers face challenges in analyzing large heart disease datasets and extracting meaningful insights. The proposed solution — an interactive Heart Disease Analysis system using Tableau and Flask — directly solves this problem by converting complex data into clear, interactive visual dashboards.

This approach helps identify behavioral patterns, understand user needs, and implement a system that improves decision-making efficiency.

Purpose:

❑ Solve complex healthcare data analysis problems using interactive visualization techniques.

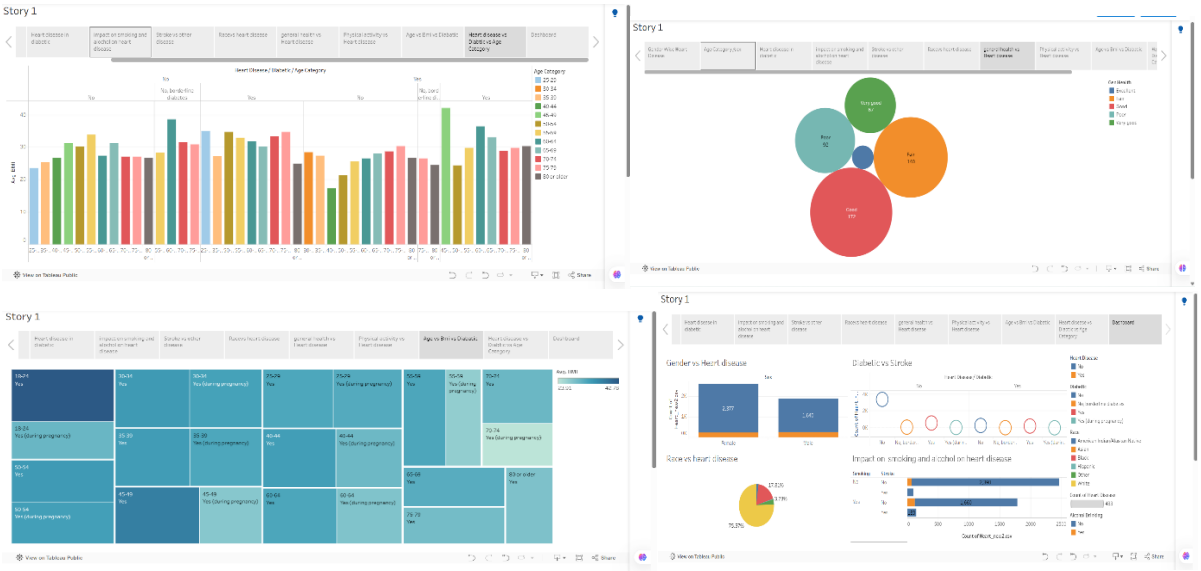
❑ Improve decision-making speed by presenting insights in a structured and user-friendly dashboard.

❑ Enhance communication of healthcare insights through clear visual representation and storytelling.

❑ Increase accessibility to heart disease data through web-based dashboard integration.

❑ Understand current data challenges and transform raw datasets into actionable insights for healthcare stakeholders.

Template:



4.2. Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

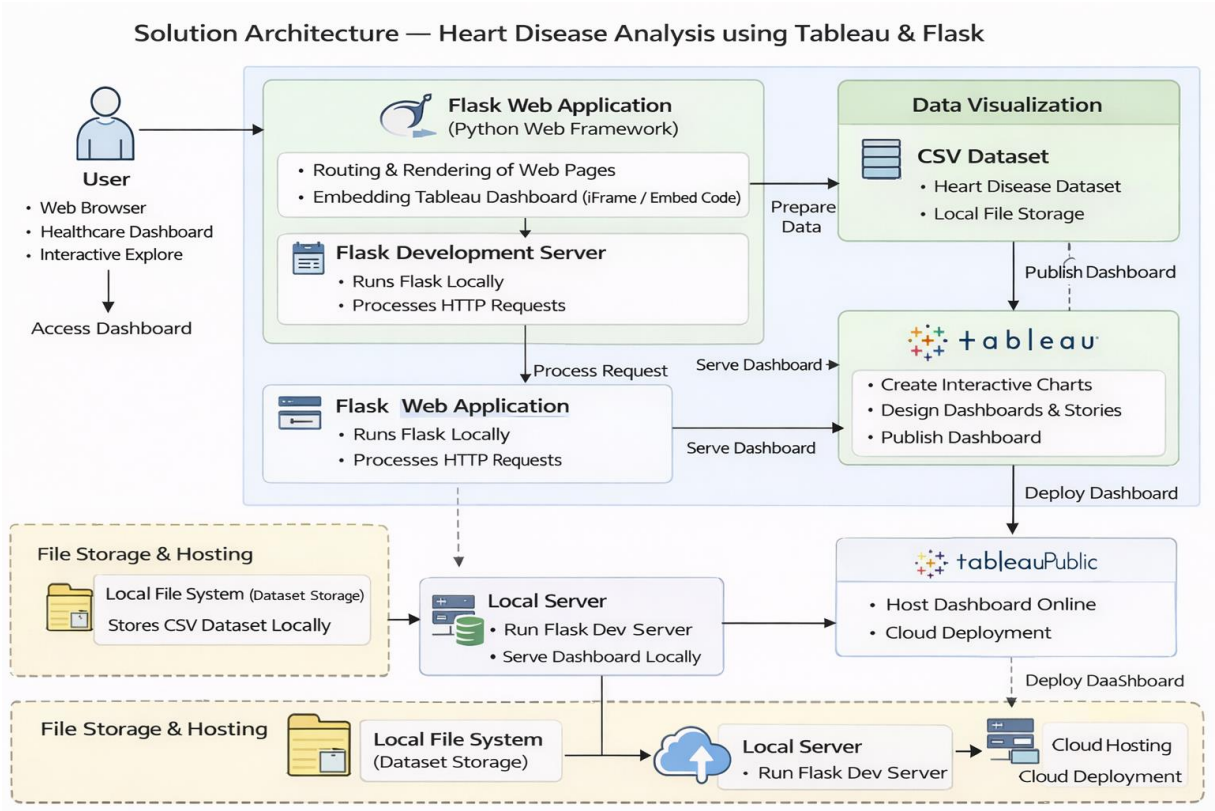
S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	Healthcare professionals and policymakers face difficulty in analyzing large heart disease datasets and identifying key risk factors. Manual data analysis is time-consuming and does not clearly highlight patterns, trends, and correlations. There is a need for an interactive system that transforms raw healthcare data into meaningful visual insights for better decision-making.
2.	**Idea / Solution Description**	The proposed solution is an interactive Heart Disease Analysis system developed using Tableau for data visualization and Flask for web integration. The system converts raw healthcare data into structured dashboards and stories that highlight important risk factors such as age, BMI, smoking, diabetes, and physical activity. The dashboard is published on Tableau Public and embedded into a web application for easy browser-based access.
3.	**Novelty / Uniqueness**	Unlike traditional static healthcare reports, this system provides interactive visual dashboards with dynamic filters. Users can explore data in real-time, apply filters, and analyze trends easily. The integration of Tableau with Flask enables web-based accessibility, making the solution more flexible and user-friendly than static reports.
4.	**Social Impact / Customer Satisfaction**	The solution helps healthcare professionals and policymakers make informed decisions based on data insights. It improves understanding of heart disease risk factors and supports preventive healthcare planning. Users benefit from clear visual representation of complex data, reducing confusion and improving productivity.
5.	**Business Model (Revenue Model)**	The system can be extended for enterprise healthcare analytics services. Revenue can be generated through dashboard customization services, healthcare analytics consulting, subscription-based advanced analytics features, or integration with hospital management systems.
6.	**Scalability of the Solution**	The dashboard is cloud-accessible through Tableau Public and can be deployed on scalable web infrastructure. The system can support multiple users accessing the dashboard simultaneously. Future enhancements such as real-time database integration, predictive analytics, and cloud hosting can further scale the solution.

4.3 Solution Architecture

Solution Architecture:

The solution architecture of the Heart Disease Analysis system consists of a Tableau-based data visualization platform integrated with a Flask web application. The heart disease dataset is first imported into Tableau, where it is cleaned and structured for analysis. Multiple interactive visualizations are created to highlight important risk factors such as age, BMI, smoking, diabetes, and physical activity. These visualizations are combined into an interactive dashboard and story format to provide clear and meaningful insights.

The completed dashboard is published on Tableau Public for online accessibility and embedded into a Flask web application using an iframe or Tableau embed code. The Flask application acts as the interface layer, allowing users to access the dashboard through a web browser. This architecture ensures a simple, scalable, and user-friendly system that transforms raw healthcare data into interactive and actionable insights.



5. PROJECT PLANNING & SCHEDULING

5.1. Project Planning

Use the below template to create product backlog and sprint schedule

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Database Setup	USN-1	As a developer, I want to store the heart disease dataset in MySQL so that it can be queried efficiently.	2	High	All Team Members
Sprint-2	SQL Data Processing	USN-2	As a developer, I want to perform SQL queries to clean and filter the dataset for analysis.	2	High	All Team Members
Sprint-3	Tableau Integration	USN-3	As a user, I want Tableau to connect with MySQL to create interactive visualizations.	3	High	All Team Members
Sprint-4	Visualization Development	USN-4	As a user, I want to analyze heart disease risk factors using charts and graphs.	3	High	All Team Members
Sprint-5	Dashboard Creation	USN-5	As a user, I want to view all visual insights in a single interactive dashboard.	2	High	All Team Members
Sprint-6	Flask Web Integration	USN-6	As a user, I want the Tableau dashboard to be embedded in a Flask web application.	2	Medium	All Team Members
Sprint-7	Deployment	USN-7	As a user, I want the application to be deployed online for public access.	2	Medium	All Team Members

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 1	20	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 3	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 4	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 5	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026
Sprint 6	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Database Connection	Connect MySQL database to Tableau	Successful database connection established	Connected Successfully	Pass
FT-02	SQL Query Execution	Run SELECT queries for filtering heart disease data	Correct filtered dataset returned	As Expected	Pass
FT-03	Visualization Rendering	Load charts in Tableau dashboard	Charts display correct data without errors	As Expected	Pass
FT-04	Dashboard Filter Functionality	Apply filters (Age, Gender, BMI, Smoking)	Dashboard updates dynamically based on filter	Working Properly	Pass
FT-05	Flask Web Integration	Open Flask app and load embedded Tableau dashboard	Dashboard loads inside web page	As Expected	Pass

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
PT-01	Dashboard Load Time	Open dashboard and measure load time	Dashboard loads within 3–5 seconds	Within Limit	Pass
PT-02	Multiple Filter Usage	Apply multiple filters simultaneously	System handles filtering without lag	Stable	Pass
PT-03	Deployment Test	Access deployed application through browser	Application loads and works correctly online	Working	Pass

7. RESULTS

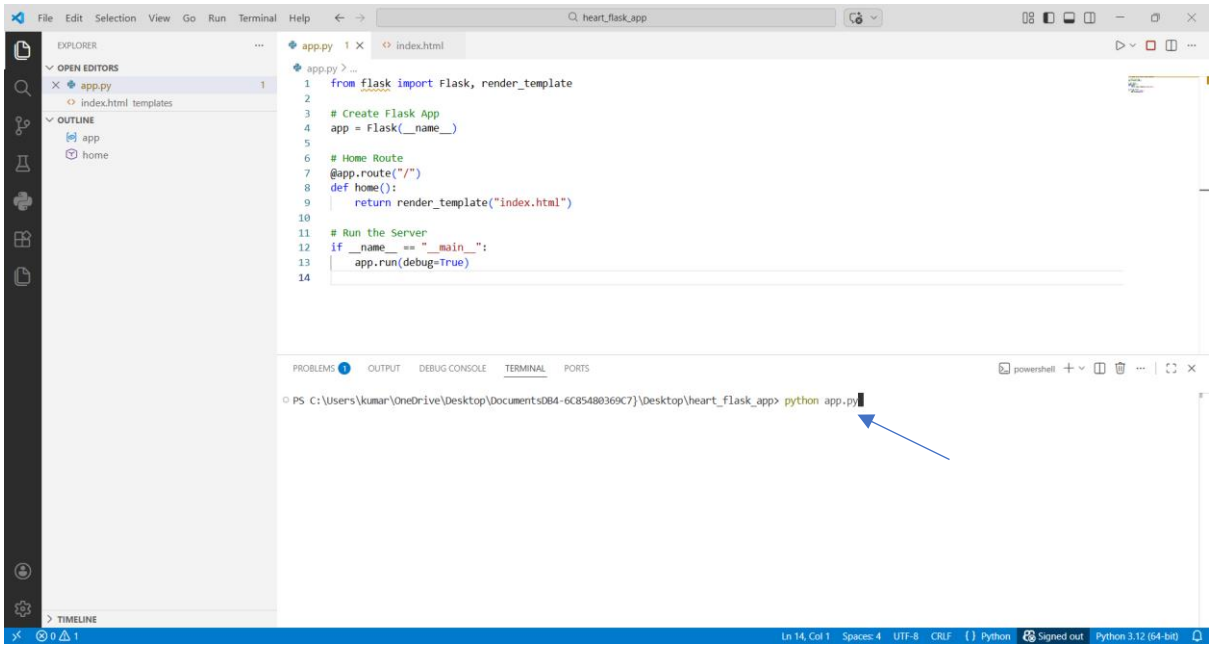
7.1. Output Screenshots

Step 1: Running the Flask Application

To execute the application, open the terminal in the project directory where the app.py file is located and run the following command:

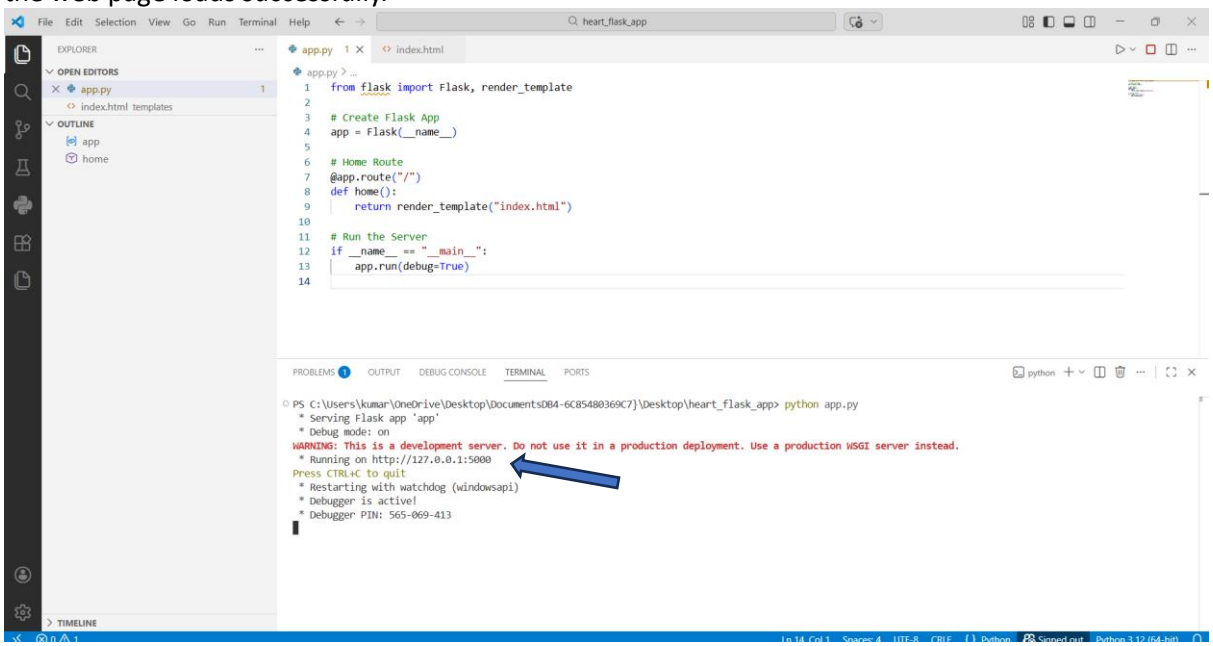
python app.py

This command starts the Flask development server.



Step 2: Accessing the Application

After running the command, the terminal generates a local host URL such as:
http://127.0.0.1:5000
The user can access the application by opening this URL in a web browser. Upon accessing the link, the web page loads successfully.



Step 3: Application Homepage Display

The Flask web page opens and displays the homepage of the Heart Disease Analysis application. The homepage contains:

- Project Title
- Brief description of the system
- Embedded Tableau Dashboard

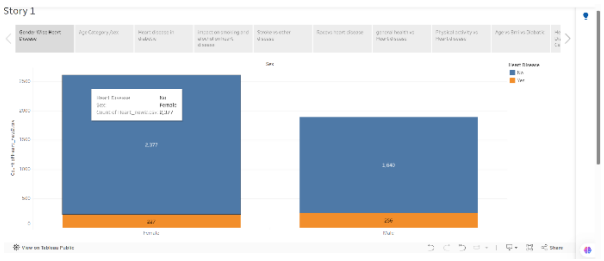
The interface is developed using Flask templates, and the dashboard is integrated using Tableau's embed functionality.

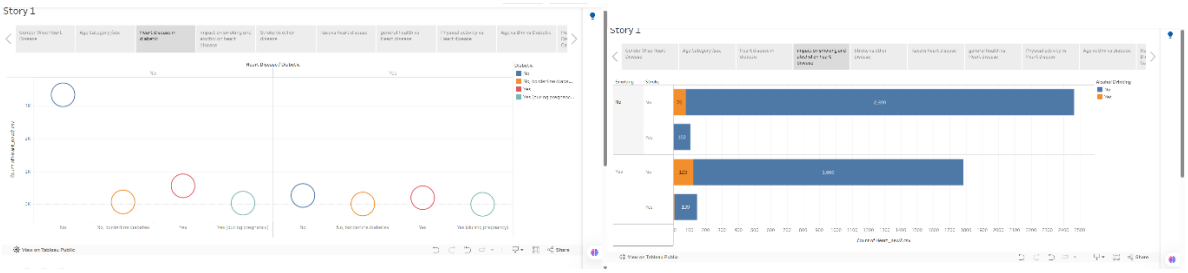
Step 4: Dashboard Interaction

Once the dashboard loads, users can interact with various filters such as:

- Age
- Gender
- BMI
- Smoking Status
- Diabetes Status
- Physical Activity

When filters are applied, the dashboard dynamically updates the visualizations in real time.



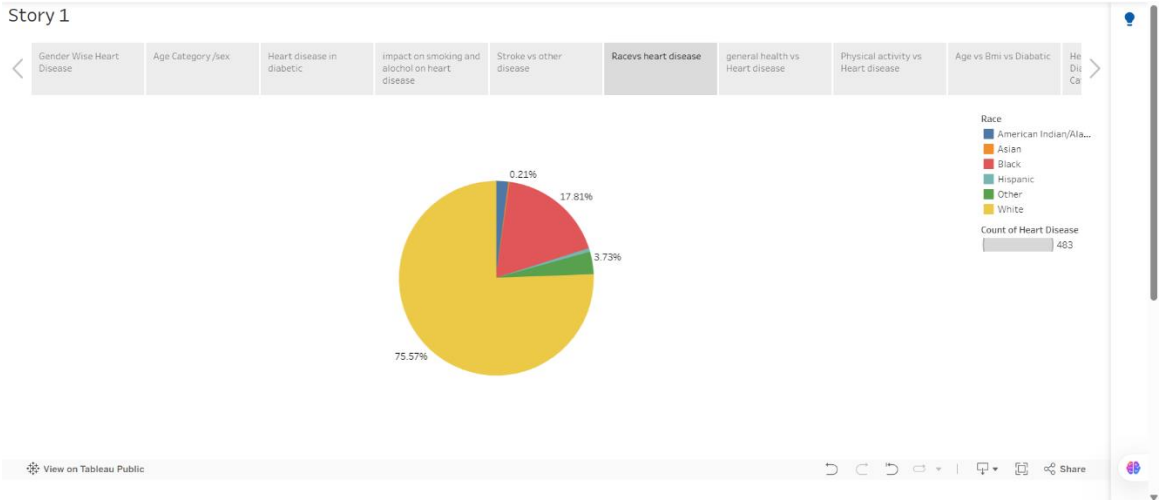


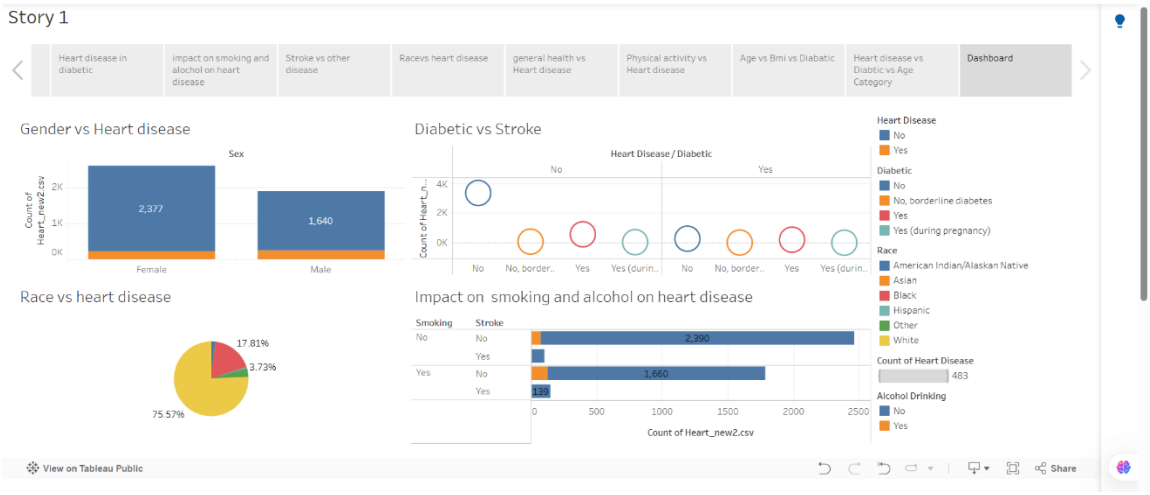
Step 5: Visualization Output

The dashboard displays multiple visualizations including:

- Gender-wise Heart Disease Analysis
- Age vs Heart Disease Trends
- Impact of Smoking and Alcohol
- Diabetes and Stroke Correlation
- Physical Activity vs Heart Disease
- Combined Age and BMI Analysis

These visualizations provide meaningful insights into heart disease risk factor





Step 6: Deployment and Accessibility

After deployment, the application becomes accessible through a public URL. The system functions correctly online, and all interactive components operate without errors.

8. ADVANTAGES AND DISADVANTAGES

Advantages

- Helps identify key risk factors contributing to heart disease
- Enables data-driven decision making for healthcare professionals
- Provides interactive and dynamic visualizations using Tableau
- Reduces manual data analysis effort
- Real-time dashboard filtering for better insights
- Integrates database, visualization, and web deployment in a single system
- User-friendly interface through Flask web integration

Disadvantages

- Requires internet connection for accessing Tableau Public dashboards
- Performance depends on database query optimization
- Limited to available dataset (no real-time hospital data integration)
- Requires basic technical knowledge to maintain or update system
- Data accuracy depends on dataset quality

9. CONCLUSION

The **Heart Disease Analysis using Tableau and Flask** project successfully demonstrates how data visualization and business intelligence tools can be used to analyze healthcare data effectively. The system transforms raw heart disease data into meaningful visual insights through interactive dashboards.

By integrating MySQL for data storage, SQL for processing, Tableau for visualization, and Flask for web deployment, the project provides a complete end-to-end analytics solution. The dashboard enables healthcare professionals, policymakers, and individuals to understand risk factors such as age, BMI, smoking, diabetes, and physical activity, thereby supporting preventive healthcare decisions.

10. FUTURE SCOPE

The project can be enhanced further by integrating real-time hospital data for live monitoring of heart disease trends. Future improvements may include adding predictive analytics using Machine Learning models to estimate heart disease risk levels.

The system can also be extended with:

- AI-based risk prediction models
- Real-time health monitoring integration
- Role-based user login system
- Advanced analytics with trend forecasting
- Mobile application deployment

These enhancements can transform the system into a comprehensive healthcare decision-support platform.

11. APPENDIX

11.1 Source Code

The source code for the **Heart Disease Analysis using Tableau and Flask** project includes:

- MySQL database setup and SQL queries for data storage and processing
- Tableau dashboard creation and visualization logic
- Flask web application code for embedding the Tableau dashboard
- HTML templates used for rendering the web interface

The application is developed using **Python (Flask framework)** and follows a structured and modular approach to ensure readability, maintainability, and scalability.

11.2. Github & Project Demo Link

Github Repository Link: - https://github.com/DurgalakshmiPeketi/Heart-Disease-Analysis

Demo Link:
https://drive.google.com/file/d/1kJAkLYCZZoUqpopoY0Mk4rSr9GPWN8Cf/view?usp=sharing