

# **PATIENT CASE SIMILARITY**

**A PROJECT REPORT**

*Submitted by,*

**Mr. Pranav Ganesh- 20211CAI0062**

**Ms. Prerna Kakade – 20211CAI0063**

**Ms. Bhuvana V – 20211CAI0069**

**Ms. Nida Aiyman – 20211CAI0085**

*Under the guidance of,*

**Dr. Mohammadi Akheela Khanum**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING – ARTIFICIAL  
INTELLIGENCE AND MACHINE LEARNING**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**DECEMBER 2024**

# **PRESIDENCY UNIVERSITY**

## **SCHOOL OF COMPUTER SCIENCE ENGINEERING**

### **CERTIFICATE**

This is to certify that the Project report “**PATIENT CASE SIMILARITY**” being submitted by “PRANAV GANESH, PRERNA KAKADE, BHUVANA V, NIDA AIYMAN” bearing roll numbers “20211CAI0062, 20211CAI0063, 20211CAI0069, 20211CAI0085” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

**Dr. Mohammadi Akheela Khanum**  
Professor  
School of CSE&IS  
Presidency University

**Dr. Zafar Ali Khan**  
N-Professor & HoD  
School of CSE&IS  
Presidency University

**Dr. L. SHAKKEERA**  
Associate Dean  
School of CSE  
Presidency University

**Dr. MYDHILI NAIR**  
Associate Dean  
School of CSE  
Presidency University

**Dr. SAMEERUDDIN KHAN**  
Pro-Vc School of Engineering  
Dean -School of CSE&IS  
Presidency University

# **PRESIDENCY UNIVERSITY**

## **SCHOOL OF COMPUTER SCIENCE ENGINEERING**

### **DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **PATIENT CASE SIMILARITY** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Engineering – Artificial Intelligence and Machine Learning**, is a record of our own investigations carried under the guidance of **MOHAMMADI AKHEELA KHANUM, PROFESSOR, School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

**Pranav Ganesh – 20211CAI0062**

**Prerna Kakade – 20211CAI0063**

**Bhuvana V – 20211CAI0069**

**Nida Aiyman – 20211CAI0085**

## **ABSTRACT**

**The Patient Case Similarity web app helps doctors and researchers by comparing new patient data with past cases.**

**It uses electronic health records (EHRs) and medical research to find patterns, and predict diseases.**

**The web app groups patients with similar conditions, like heart diseases, to spot trends and improve diagnosis accuracy. It helps build better prediction tools to improve patient care.**

**First, we load the data into a program. Then, we look at the data to see patterns or connections. If there's missing information, we fix it by either filling it in or removing it. Next, we use charts to make the data easier to understand. We split the data into two parts: one to train the model and one to test it. We train the model with the first part and check how well it works with the second part. Lastly, we group similar data together to find patterns.**

**Our innovative approach enhances the modern day medical decision which leads to better patient outcomes.**

## ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair**, School of Computer Science Engineering & Information Science, Presidency University, and Dr. Zafar Ali Khan, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr. Mohammadi Akheela Khanum, Professor**, and Reviewer **Dr. Afroz Pasha, Associate Professor**, School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Pranav Ganesh – 20211CAI0062**

**Prerna Kakade – 20211CAI0063**

**Bhuvana V – 20211CAI0069**

**Nida Aiyman – 20211CAI0085**

## LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1	Table 2.1	Literature Survey	2-7

## LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1	Figure 3.1	Bar Plot of Age VS. Heart Rate	12
2	Figure 3.2	Histogram of Heart Rate	12
3	Figure 3.3	Pairplot of patient data	13
4	Figure 3.4	Scatter plot for K-Means Clustering	15
5	Figure 6.1	System Design	18
6	Figure 7.1	Timeline using Gantt Chart	19

# **TABLE OF CONTENTS**

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
	<b>Abstract acknowledgment</b>	<b>v</b>
	<b>List of tables</b>	<b>vi</b>
	<b>List of figures</b>	<b>vii</b>
<b>1.</b>	<b>Introduction</b>	<b>1</b>
	1.1 general	1
<b>2.</b>	<b>Literature review</b>	<b>2</b>
<b>3.</b>	<b>Research gaps of existing methods</b>	<b>8</b>
<b>4.</b>	<b>Proposed methodology</b>	<b>10</b>
	4.1 Loading data	10
	4.2 Exploring data	10
	4.3 Missing values	10
	4.4 Preprocessing	11
	4.5 Data visualiation	11
	4.6 Training and testing	13
	4.7 Models	14
	4.8 Clustering	15
	4.9 Silhouette score	16
<b>5.</b>	<b>Objectives</b>	<b>17</b>
<b>6.</b>	<b>System design &amp; implementation</b>	<b>18</b>
<b>7.</b>	<b>Timeline for execution of project</b>	<b>19</b>
<b>8.</b>	<b>Outcomes</b>	<b>20</b>
<b>9.</b>	<b>Results and discussions</b>	<b>21</b>
<b>10.</b>	<b>Conclusion</b>	<b>22</b>
<b>11.</b>	<b>References</b>	<b>23</b>
<b>12.</b>	<b>Appendix-a</b>	<b>25</b>
	Front end	25
	Back end	47
<b>13.</b>	<b>Appendix-b</b>	<b>52</b>



<b>14.</b>	<b>Appendix-c</b>	<b>57</b>
	14.1 Journal publication	57
	14.2 Certifications and awards	58
	14.3 Plagiarism check	59
	14.4 SDG mapping	60

# **CHAPTER-1**

## **INTRODUCTION**

Patient Case Similarity is used in healthcare systems, mainly in clinical decision support systems, to give similarity scores between the new and old patients. In this project, we are developing a web application designed for doctors and researchers to enhance patient care and medical research by comparing a new patient's data with a historical patient. The data is gathered from electronic health records (EHRs) and various research papers. This is done to identify similar patterns and predict the disease. The main goal is to cluster patients based on heart diseases. After which we will improve the diagnostic accuracy and predictive models to give the similarity score between the patients.

## CHAPTER-2

### LITERATURE SURVEY

Research Paper	Year	Advantages	Disadvantages
i) Use abstracted patient-specific features to assist an information-theoretic measurement to assess similarity between medical cases	2008	<ul style="list-style-type: none"> <li>• Improve case similarity measurement</li> <li>• Integrate natural language processing (NLP) for feature abstraction</li> </ul>	<ul style="list-style-type: none"> <li>• The study admits limitations and focuses on only four feature types and the lack of contextual information in weighing features.</li> <li>• Future research should examine the impact of additional features and contextual factors on similarity measures to upgrade their suitability.</li> </ul>
ii) Patient Similarity: Emerging Concepts in Systems and Precision Medicine	2016	<ul style="list-style-type: none"> <li>• The goal is to set the foundation for the integration of computational tools and data analytics to enhance personalized healthcare.</li> <li>• The main focus is how patient similarity algorithms can improve medical decisions by grouping similar</li> </ul>	<ul style="list-style-type: none"> <li>• Data Difficulties</li> <li>• Growth Challenges</li> </ul>

		patients.	
iii) Patient similarity for precision medicine: A systematic review	<b>2018</b>	<ul style="list-style-type: none"> <li>The primary aim is to improve clinical outcomes for individual patients through more precise treatment targeting by leveraging on genetic, biomarker, phenotypic, or psychosocial characteristics.</li> <li>That distinguish a given patient from others with similar clinical presentations</li> </ul>	<ul style="list-style-type: none"> <li>Limited Database Scope</li> <li>Lack of Real-World Application</li> </ul>
iv) A patient-similarity-based model for diagnostic prediction	2019	<ul style="list-style-type: none"> <li>To simulate the clinical reasoning of doctors, retrieve analogous patients of an index patient automatically and predict diagnoses by the similar/dissimilar patients.</li> <li>The main goal is to predict patient diagnoses by comparing the similarities between the clinical features</li> </ul>	<ul style="list-style-type: none"> <li>Limited Dataset Size</li> <li>High Computational Costs</li> </ul>

		of current patients and historical patient data.	
v) Measurement and application of patient similarity in personalized predictive, modeling based on electronic medical records	2019	<ul style="list-style-type: none"> <li>The main goal of this research was to create a method to measure how similar patients are using data from electronic medical records(EMRs).</li> <li>By measuring similarity, the study aimed to improve predictions of health outcomes, particularly for diabetes.</li> </ul>	<ul style="list-style-type: none"> <li>The study didn't fully use all the available data when calculating patient similarity</li> <li>The models didn't include specific exclusion criteria when choosing patients for the study.</li> </ul>
vi)Measuring patient similarities using a deep learning model with medical concept embedding	2019	<ul style="list-style-type: none"> <li>Created a framework to measure clinical similarities between patients using EHRs.</li> <li>Kept track of time-related information in patient data, which is often missed in other</li> </ul>	<ul style="list-style-type: none"> <li>Loss of temporal information in existing methods:</li> <li>High dimensionality and sparsity</li> </ul>

		model.	
vii) Patient-Case Similarity	2020	<ul style="list-style-type: none"> <li>• Build a system to identify patients with similar medical histories.</li> <li>• Improve decision-making processes in clinical environments using patient data.</li> </ul>	<ul style="list-style-type: none"> <li>• Data Quality Dependency</li> <li>• Complex Medical Cases</li> </ul>
viii) Patient similarity: methods and applications	2020	<ul style="list-style-type: none"> <li>• Study and compute similarities between patients using electronic health records (EHRs), genetic, and other data.</li> <li>• Improve predictive models in healthcare by integrating patient-specific data from various sources.</li> </ul>	<ul style="list-style-type: none"> <li>• Information Loss</li> <li>• Complexity in Implementation</li> </ul>
ix) Patient similarity analytics for explainable clinical risk prediction	2021	<ul style="list-style-type: none"> <li>• To develop an explainable and interpretable Clinical Risk Prediction Model (CRPM) by</li> </ul>	<ul style="list-style-type: none"> <li>• Incomplete Variable Set</li> <li>• Static Data Usage</li> </ul>

		<p>leveraging patient similarity analytics, specifically to improve explainability and interpretability.</p> <ul style="list-style-type: none"> <li>To use real-world data from electronic medical records of patients with type-2 diabetes, hypertension, and dyslipidaemia in Singapore to develop and validate the patient similarity model</li> </ul>	
x) A Novel Patient Similarity Network (PSN) Framework Based on Multi-Model Deep Learning for Precision Medicine	2022	<ul style="list-style-type: none"> <li>Uses multi-model deep learning to identify similarities among patients.</li> <li>The patient Similarity Network (PSN) approach aims to improve precision medicine by using different types of data, like clinical records, genetic data, and imaging</li> </ul>	<ul style="list-style-type: none"> <li>Different types and sizes of data</li> <li>Limited access to public datasets</li> </ul>
xi) Deep Dynamic	2022	<ul style="list-style-type: none"> <li>Develop a Novel</li> </ul>	<ul style="list-style-type: none"> <li>Limited Clinical</li> </ul>

Patient Similarity Analysis: Model Development and Validation in ICU		<p>Dynamic Patient Similarity Model</p> <ul style="list-style-type: none"> <li>• Validate Model Using Clinical Tasks</li> </ul>	<p>Application</p> <ul style="list-style-type: none"> <li>• Computational Complexity</li> </ul>
xii) Patient Case Similarity	2024	<ul style="list-style-type: none"> <li>• Improve healthcare analytics by utilizing data science techniques to enhance diagnostics, treatment recommendations, and patient care outcomes</li> <li>• The system uses machine learning (ML) and natural language processing (NLP) to identify similarities between patient cases, enabling more personalized, data-driven medical action.</li> </ul>	<ul style="list-style-type: none"> <li>• Prescription Recommendation Accuracy</li> <li>• Scalability Concerns</li> </ul>



## **CHAPTER-3**

### **RESEARCH GAPS OF EXISTING METHODS**

i. Use abstracted patient-specific features to assist an information-theoretic measurement to assess similarity between medical cases – 2008:-

The study only looks at four types of features and doesn't consider the context when weighing them. Future research should look at how adding more features and context can improve similarity measures

ii. Patient Similarity: Emerging Concepts in Systems and Precision Medicine – 2016:-

Complexity of algorithms – many of the proposed algorithms are not yet optimized for real-world clinical use due to their complexity and reliance on high-end computing infrastructure.

iii. Patient similarity for precision medicine: A systematic review – 2018:-

Lack of Deep Learning Exploration - The paper talks very little about deep learning, which is now important for analyzing complex medical data and finding patient similarities. This might be a missed chance to use better methods.

iv. A patient-similarity-based model for diagnostic prediction – 2019:-

Low Success Percentage - The model's success percentage (the percentage of patients for whom diagnoses were correctly predicted) is low (19%).

v. Measurement and application of patient similarity in personalized predictive, modeling based on electronic medical records – 2019 :-

The models didn't include specific exclusion criteria when choosing patients for the study. This could affect the accuracy of the predictions because not all patients may be equally relevant for the predictive task

vi. Measuring Patient Similarities via a Deep Architecture with Medical Concept Embedding – 2019:

Electronic Health Records are complex, and patient records contain sparse and high-dimensional data.

vii. Patient-Case Similarity – 2020:

The use of machine learning models leads to over fitting, where the model performs well on training data but fails to generalize new or unseen data.

viii. Patient similarity: methods and applications – 2020

During data transformation and integration, particularly in early integration strategies, there is a risk of losing valuable patient information.

ix. Patient Similarity Analytics for Explainable Clinical Risk Prediction – 2021

The model doesn't include important factors like gender, race, diet, and lifestyle, which are linked to complications of diabetes, hypertension, and high cholesterol. Missing this data makes the model less complete.

x. A Novel Patient Similarity Network (PSN) Framework Based on Multi-Model Deep Learning for Precision Medicine – 2022

Data Heterogeneity and Dimensionality - A Novel Patient Similarity Network (PSN) Framework Based on Multi-Model Deep Learning for Precision Medicine – 2022

The mix of different types of clinical data, both structured and unstructured, makes it hard to create accurate models. Handling this complex data that can cause information to be lost during processes like auto encoders, which reduce the data's size. Using auto encoders for this can lead to a loss in accuracy.

xi. Deep Dynamic Patient Similarity Analysis: Model Development and Validation in ICU – 2022

Need for Clinical Protocol - The model requires a clinical protocol for practical implementation, which hasn't been discussed in this research

xii. Patient Case Similarity – 2024: Prescription Recommendation Accuracy:

The system shows lower accuracy in prescription recommendations (61.27%), indicating room for improvement in this area.

## **CHAPTER-4**

### **PROPOSED METHODOLOGY**

#### 1. Loading data :

- We have first imported the pandas library in Python which is used for data manipulation and analysis.
- The `pd.read_csv()` method has been used in order to load data from CSV files into pandas DataFrames.
- The DataFrame is a fundamental data structure in pandas so it becomes simple to read CSV data directly into the DataFrame.
- We have initially loaded two datasets : test data and train data. These two DataFrames are concatenated into a single DataFrame using the `concat()` function.

#### 2. Exploring

- `Data.info()` is used in the exploration step of the algorithm. This step gives us information about all the columns present in the dataframe.
- It confirms the data structure by displaying the class of the object.
- It lists down all the columns name in order for easy reference.
- It also displays the count of the number of non-null values in each column which helps us to easily identify missing data.
- It shows the memory that the data frame has consumed.
- `Data.describe()` provides summary of statistics of all the numerical data in the dataframe such as count, mean, std, min, max and the percentiles.

#### 3. Missing values

- This method lets us know how many missing values are present in each column of the dataframe.

- The `isnull()` function generates the missing values in the dataframe with `True` indicating missing values and `False` indicating no missing values.
- The `.sum()` method calculates the total number of missing values in each column by considering the number of `True` values.
- Columns that show 0 count are the columns that have no missing values.

#### 4. Preprocessing

- This is a crucial step in data analysis which prepares raw data for further analysis and modeling.
- An effective way to handle these missing values is by filling them by finding out the median value of each column.
- Certain columns like `TenYearCHD`, `is_smoking` and `education` are dropped because they provide unnecessary information.

#### 5. Data visualization

- In our project, we have used libraries such as `matplotlib` and `seaborn`.
- `Matplotlib` and `seaborn` are popular python libraries.
- `Matplotlib` is used for generating charts and plots.
- It allows customization of the charts and plots by adding titles, changing color of graph and giving labels on X axis and Y axis.
- `Seaborn` is built on top of `matplotlib` and is a library that is used to make complicated statistical visuals easier to create and understand.
- A bar plot has been created in order to compare the Age vs. Heart Rate of the patient and give us a detailed visual view of the same.

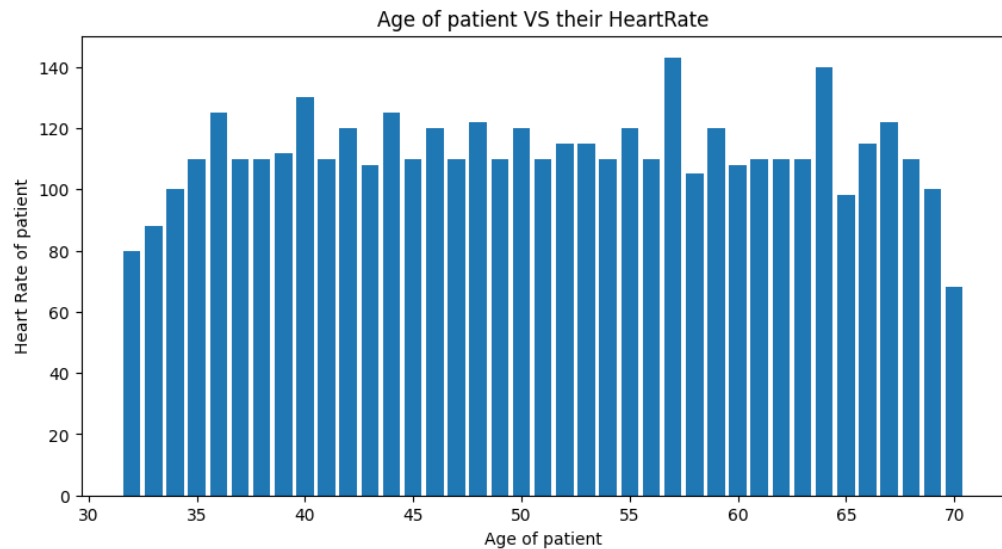


Fig 3.1: Bar Plot of Age VS. Heart Rate

- A Histogram has been created for checking the distribution of Heart Rates across various Frequencies. By looking at the Histogram it can be checked whether the heart rate is normally distributed, skewed or bimodal. If the histogram shows peak, then the data might be normally distributed.

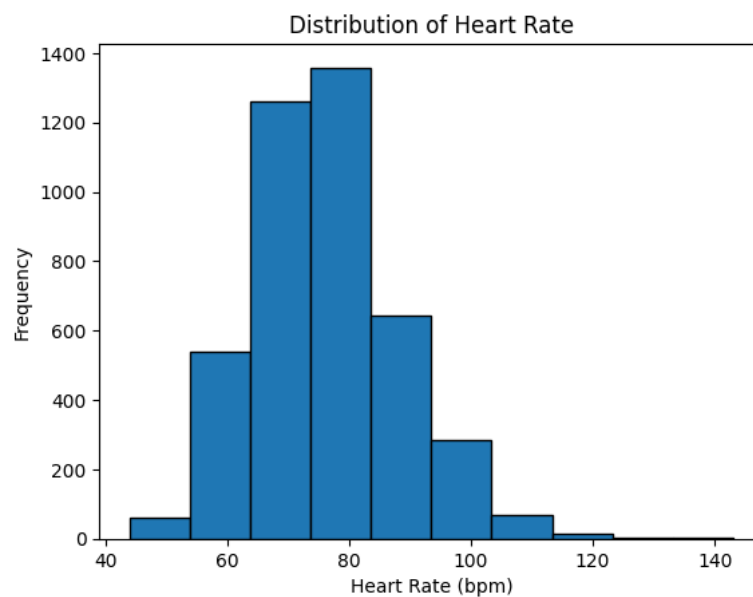


Fig 3.2: Histogram of Heart Rate

- A pairplot has also been created to show the relationships between variables in the dataset.

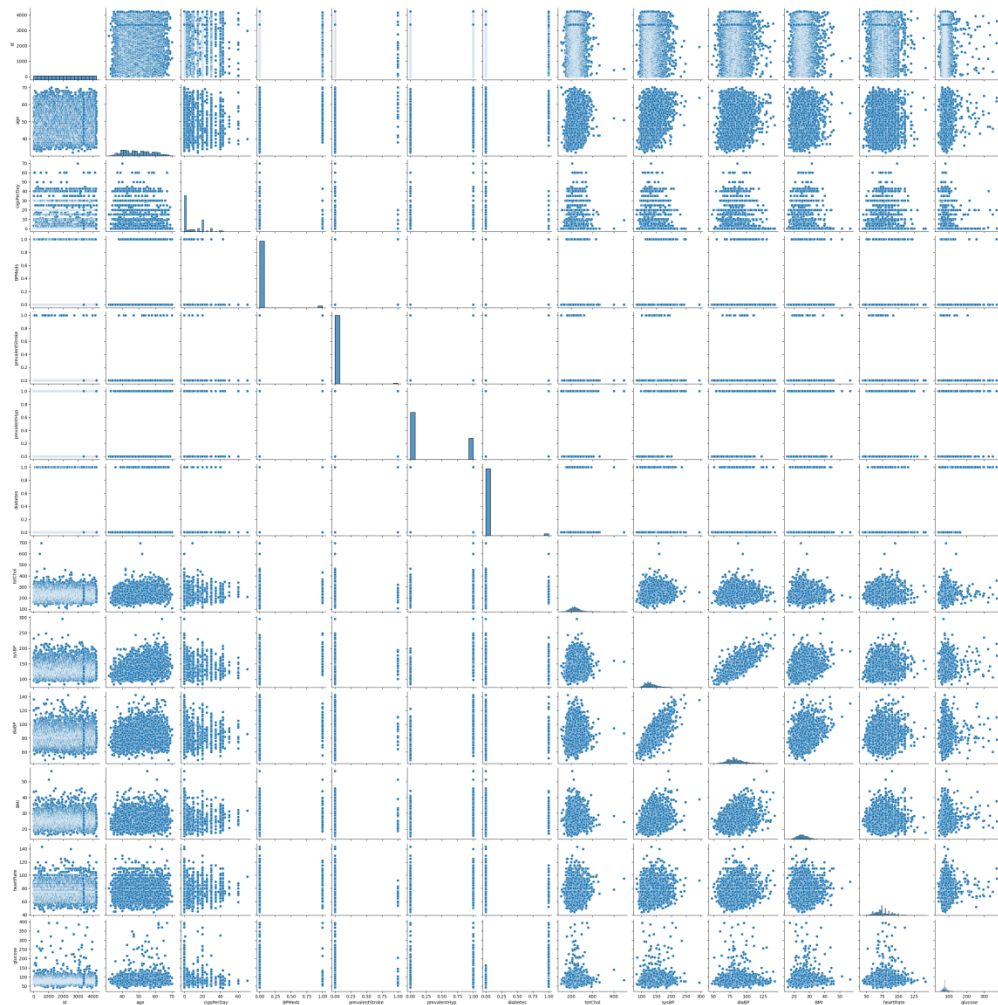


Fig 3.3: Pairplot of patient data

## 6. Training and testing

- Before training the model, the data has to be split into two subsets: training data and testing data.
- In most cases, about 70-80% of the data is used for training. And the remaining 20-30% is used for testing. In our project, we have utilized 70% of the data for training and 30% for testing in order to prevent over fitting.
- During training, the model is fed data from which it learns patterns. The input data contains features such as :

1. Age
2. cigsPerDay
3. BPMeds
4. prevalentStroke
5. prevalentHyp
6. Diabetes
7. totChol
8. sysBP
9. diaBP
10. BMI
11. heartrate
12. Glucose.

- If the model performs poorly on training data then it may show issues such as under fitting which did not happen in our case.
- Once the training is done, the model is tested on new, unseen data. The model makes predictions and the results are compared to the actual outcomes to evaluate its performance.
- The testing phase is crucial because it makes sure that the model is not just memorizing the training data but also making accurate predictions on new unseen data.

## 7. Models

- The models that we have used in our backend are Logistic Regression, Decision Tree and Random Forest.
- Logistic Regression: This is used to predict if the patient has heart diseases or not. '1' if the patient has heart diseases and '0' if the patient does not have heart diseases based on features such as Age and cholesterol.
- Decision Tree: This is a flowchart which is used to make decisions

based on different features. In our patient case similarity model, this model uses patient features to split the data into branches which leads to an accurate prediction.

- Random Forest: It is a collection of multiple decision trees. An ensemble method that combines multiple decision trees instead of just one to make an accurate and stable prediction.

## 8. Clustering

- Clustering is a method in data analysis that groups similar objects or data points together.
- In our project, we have implemented K-Means clustering.
- K-Means Clustering is a well-known machine learning method used to divide data into a number of groups. It is an unsupervised machine learning technique.
- Our code groups patient data into three clusters using K-Means Clustering and shows the results with a chart.
- Next, the average values of each feature for each cluster are calculated and shown. A scatter plot is made with patient age on the x-axis and cholesterol levels on the y-axis, and the points are colored based on their cluster.

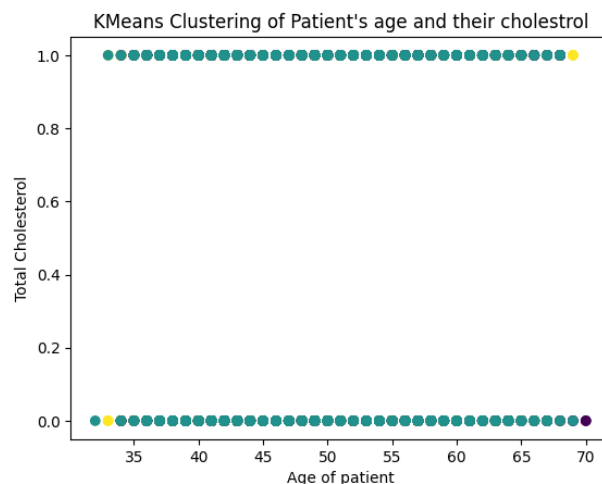


Fig 3.4: Scatter plot for K-Means Clustering



## 9. Silhouette score

- The Silhouette Score checks how well data points are grouped together by calculating the score for different numbers of clusters using K-Means.
- It compares how similar each point is to its own group compared to other groups.
- It prints the score for each number of clusters to help us choose the best one.
- Higher score means the groups are clear and better formed.

## **CHAPTER-5**

### **OBJECTIVES**

- The goal of this project is to make use of historical patient data in order to make accurate and useful predictions about heart diseases based on patient similarities.
- The main focus of the project is data driven decision making.
- To develop predictive models which aim to analyze patient similarities by early and accurately predict heart diseases.
- By improving the diagnostic accuracy, the model aims to reduce healthcare costs. It can reduce the cost of unnecessary test costs and hospital fees.
- To identify early detection of heart diseases so that better prevention steps can be taken and a better personal plan can be made.
- To improve healthcare efficiency by automating the identification of patient similarities, saving the time and resources of healthcare providers.
- Enhancement of clinical research by giving the researchers a tool that helps them in their studies about heart diseases.
- The model aims to create more awareness about strengthening preventive care, taking precautions and making future generations aware of the horrors of heart diseases.
- Enhancing patient engagement by developing a user-friendly website that can be used and understood by the masses.

## CHAPTER-6

### SYSTEM DESIGN & IMPLEMENTATION

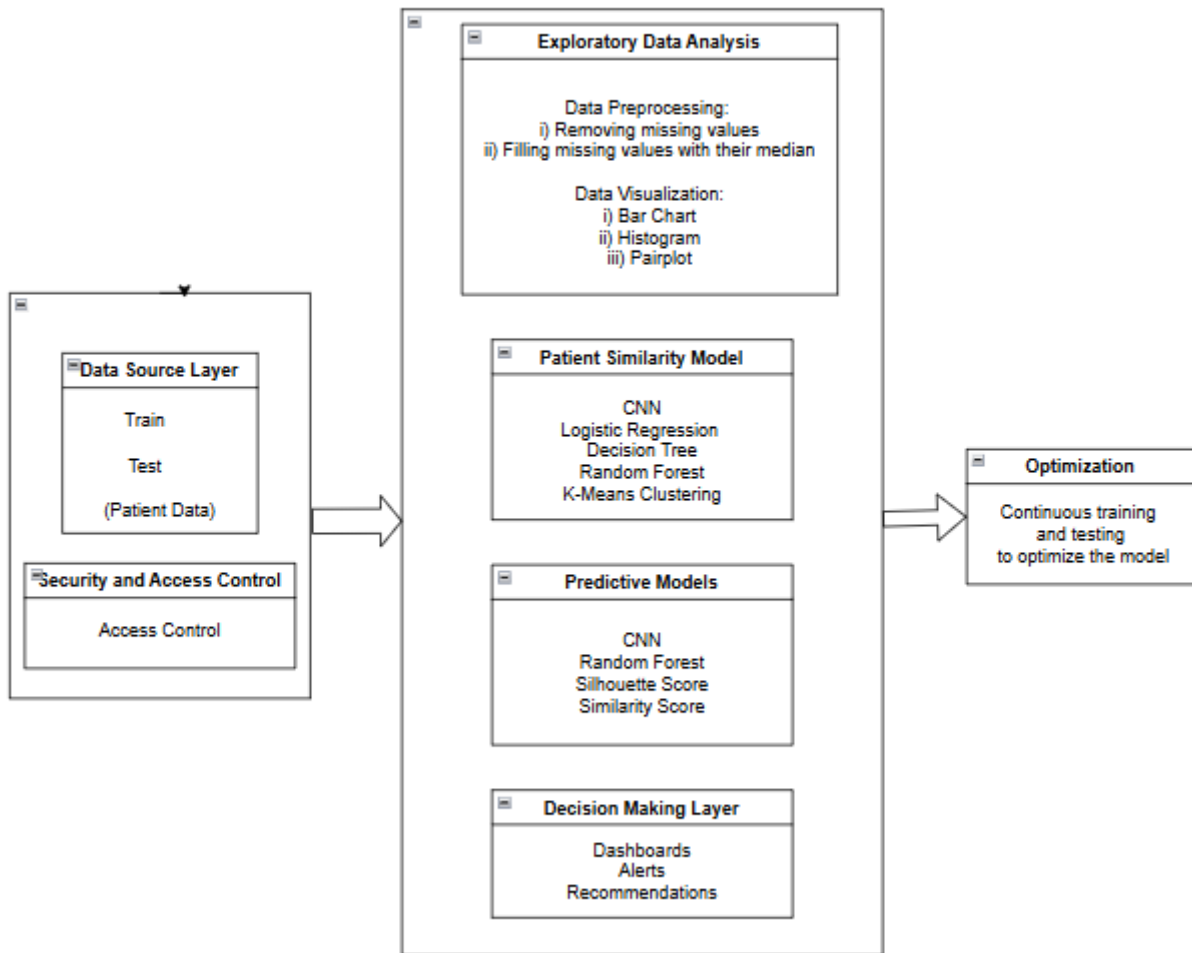


Fig 6.1 : System Design

## CHAPTER-7

### TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)

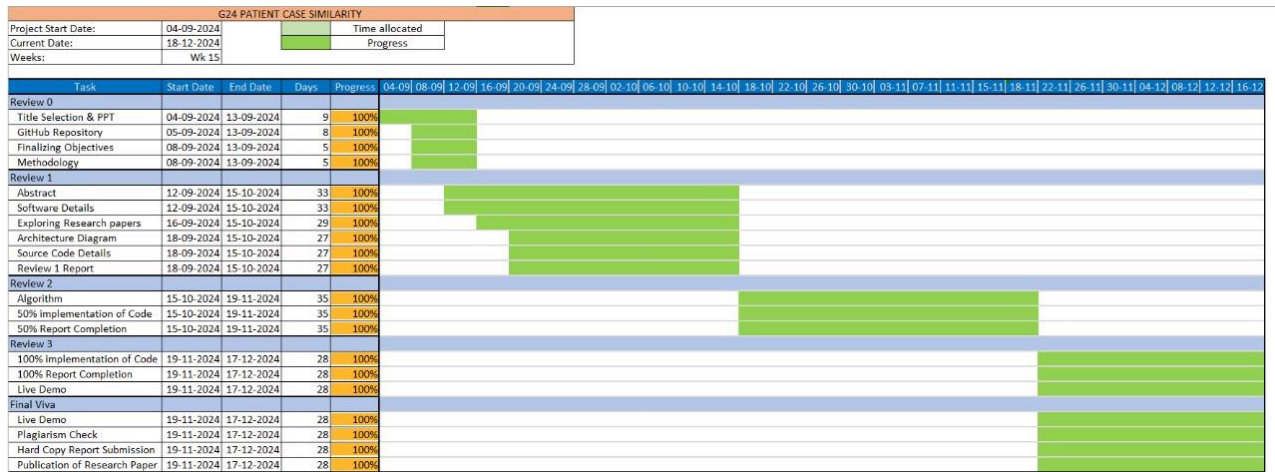


Fig 7.1: Timeline using Gantt Chart

## CHAPTER-8

### OUTCOMES

#### **Most Similar Patients:**

Patient ID: 10, Similarity: 0.9894803221396218

Patient ID: 11, Similarity: 0.9896825252977868

Patient ID: 14, Similarity: 0.9972892831540058

Patient ID: 21, Similarity: 0.9921180983850746

Patient ID: 23, Similarity: 0.9886220303454638

#### **Most Similar Patients:**

Patient ID: 1, Similarity: 0.7798798968758358

Patient ID: 6, Similarity: 0.8224959896114975

Patient ID: 10, Similarity: 0.773295464154435

Patient ID: 14, Similarity: 0.782637663524027

Patient ID: 21, Similarity: 0.7797529557896715

## **CHAPTER-9**

### **RESULTS AND DISCUSSIONS**

- The dataset “test.csv” and “train.csv” were merged for data analysis and modeling.
- Missing values in columns such as `cigsPerDay`, `BPMeds`, `totChol`, `BMI`, `heartRate`, `glucose`, were filled with their median values.
- Valuable insights were gained from visualization. A bar chart showed the relationship between age of patients and their heart rates. Histogram showed a normal distribution of heart rate values. Pairwise relationships between numeric variables were visualized using Pairplot.
- Logistic Regression gave an accuracy of 85.84%, Decision Tree Classifier gave an accuracy of 75.42% and Random Forest Classifier gave an accuracy of 85.05%. The best model was determined by these accuracies, that is, Logistic Regression Model.
- K-Means clustering grouped patients together based on their features and for these clusters a Silhouette score was determined. Different ranges of k-values gave various silhouette scores, some even higher scores, which suggested better defined clusters.
- We have successfully determined the similarity score of a new patient in comparison to five patients from the historical data. The number of patients can be adjusted as desired and the model will still accurately compute the similarity scores.

## **CHAPTER-10**

### **CONCLUSION**

The project "Patient Case Similarity" has successfully been used to demonstrate the power of Machine Learning in order to enhance clinical decision support systems. Early diagnoses of heart diseases can be made by making use of the historical patient data and identifying similarities among the patients. Machine Learning models such as Logistic Regression, Decision Trees and Random Forests have been used for reliable predictive performance. K-Means clustering has been used for grouping together of patients profiles based on similarity among them.

The project provides immense benefits because it not only reduces the overall healthcare costs by eliminating unnecessary tests but also promotes medical decision-making by improving efficiency. The user-friendly website promotes accessibility for medical professionals and patients improving user interaction and user satisfaction. It also reduces the gap between clinical practices and data driven insights by encouraging personalized medicine and better healthcare.

In the future, the project could be enhanced by expanding the dataset and combining Deep Learning for better predictive accuracy and clustering efficiency.

## REFERENCES

- i. Use abstracted patient-specific features to assist an information-theoretic measurement to assess similarity between medical cases – 2008:  
<https://www.sciencedirect.com/science/article/pii/S1532046408000440>
- ii. Patient Similarity: Emerging Concepts in Systems and Precision Medicine – 2016:  
<https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2016.00561/full>
- iii. Patient similarity for precision medicine: A systematic review – 2018: <https://doi.org/10.1016/j.jbi.2018.06.001>
- iv. A patient-similarity-based model for diagnostic prediction – 2019: <https://doi.org/10.1016/j.ijmedinf.2019.104073>
- v. Measurement and application of patient similarity in personalized predictive, modelling based on electronic medical records – 2019: <https://doi.org/10.1186/s12938-019-0718-2>
- vi. Measuring Patient Similarities via a Deep Architecture with Medical Concept Embedding – 2019: <https://arxiv.org/pdf/1902.03376>
- vii. Patient-Case Similarity – 2020:  
<https://www.researchpublish.com/upload/book/Patient%20Case%20Similarity-8606.pdf>
- viii. Patient similarity: methods and applications – 2020: <https://arxiv.org/pdf/2012.01976>
- ix. Patient similarity analytics for explainable clinical risk prediction – 2021:  
<https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/>



[s12911-021-01566-y](#)

- x. **A Novel Patient Similarity Network (PSN) Framework Based on Multi-Model Deep Learning for Precision Medicine – 2022:**  
<https://doi.org/10.3390/jpm12050768>
- xi. **Deep Dynamic Patient Similarity Analysis: Model Development and Validation in ICU – 2022:**  
<https://www.sciencedirect.com/science/article/pii/S0169260722004151>
- xii. **Patient Case Similarity - 2024:**  
<https://www.doi.org/10.56726/IRJMETS48246>

## APPENDIX-A

### PSUEDOCODE

#### FRONTEND

##### Mainpage.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="ISO-8859-1">
```

```
<title>Patient Case Similarity</title>
```

```
<style>
```

```
body {
```

```
    background-image:
```

```
url('C:/Users/Dhruv/Desktop/Prerna_capstone/capstone/capstone/assets.jpg');
```

```
    background-size: cover;
```

```
    font-family: Arial, sans-serif;
```

```
}
```

```
.header-image {
```

```
    width: 100%; /* Full width */
```

```
    height: 720px; /* Maintain aspect ratio */
```

```
    display: block;
```

```
    position: relative; /* Needed for overlay positioning */
```

```
}
```

```
.header-text {
```

```
    position: absolute;
```

```
    top: 20%; /* Adjust to position the text vertically */
```

```
    left: 50%;
```

```
transform: translate(-50%, -50%);
text-align: center;
color: black; /* Changed to black */
text-shadow: 2px 2px 8px rgba(255, 255, 255, 0.6); /* Light shadow for
contrast */
}
```

```
.header-text h1 {
  font-size: 56px;
  font-weight: bold;
}
```

```
.header-text h2 {
  font-size: 48px;
  font-weight: bold;
}
```

```
.rib {
  background-color: #333; /* Hospital-theme blue */
  overflow: hidden;
  display: flex;
  justify-content: center;
  align-items: center;
  width: 50%;
  max-width: 615px;
  height: 60px;
  position: absolute;
  top: 70%; /* Align with the full image */
  left: 50%;
}
```

```
transform: translate(-50%, -50%);
border-radius: 10px;
}
```

```
.rib a {
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 20px;
  margin: 0 10px;
}
```

```
.rib a:hover {
  background-color: #ddd;
  color: black;
}
```

```
.rib a.active {
  background-color: #04AA6D;
  color: white;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<br/>
```

```
<div style="position: relative; text-align: center;">
```

```
  
```

```
<div class="header-text">
  <h1>Patient Case Similarity System</h1>
  <h2>How can we help You?</h2>
</div>
<div class="rib">
  <a href="/mainpage.html">Home</a>
  <a href="/about.html">About Us</a>
  <a href="/locateus.html">Find a location</a>
  <a href="/registrationpage.html">Patient Portal</a>
</div>
</div>

</body>
</html>
```

### **About.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us - Patient Case Similarity System</title>
  <style>
    body {
      background-image:
url('https://t3.ftcdn.net/jpg/07/86/34/68/360_F_786346890_E4BqkCpTHWZg4
LIcBnqG2MdyB1Afxz1z.jpg');
      background-size: cover;
      font-family: Arial, sans-serif;
      color: white;
```

```
}
```

```
h1, h2 {  
    color: white;  
    text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.6);  
}
```

```
h1 {  
    font-size: 48px;  
    font-weight: bold;  
    text-align: center;  
    margin-top: 20px;  
}
```

```
h2 {  
    font-size: 36px;  
    font-weight: bold;  
    text-align: center;  
    margin-top: 20px;  
}
```

```
.rib {  
    background-color: #333;  
    overflow: hidden;  
    display: block;  
    width: 615px;  
    margin: 0;  
    padding: 0;  
    height: 60px;
```

```
        overflow: auto;
        margin: 20px auto;
        border-radius: 10px;
    }

    .rib a {
        float: left;
        color: #f2f2f2;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 25px;
    }

    .rib a:hover {
        background-color: #ddd;
        color: black;
    }

    .rib a.active {
        background-color: #04AA6D;
        color: white;
    }

    .about-container {
        width: 80%;
        margin: 20px auto;
        background-color: rgba(0, 0, 0, 0.6);
        padding: 20px;
```

```
border-radius: 10px;
box-shadow: 0 0 15px rgba(0, 0, 0, 0.5);
text-align: justify;
}
```

```
.about-container h2 {
margin-bottom: 15px;
text-align: center;
}
```

```
.about-content {
font-size: 18px;
line-height: 1.8;
color: #f0f0f0;
}
```

```
.about-content strong {
color: #04AA6D;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Patient Case Similarity System</h1>
```

```
</header>
```

```
<nav>
```

```
<div class="rib">
```

```
<center>
```



```
<a href="/mainpage.html">Home</a>
<a href="/about.html" class="active">About Us</a>
<a href="/locateus.html">Find a location</a>
<a href="/registrationpage.html">Patient Portal</a>
</center>
</div>
</nav>
```

```
<section class="about">
  <div class="about-container">
    <h2>About Us</h2>
    <div class="about-content">
      <p>
```

**Patient Case Similarity System** is a revolutionary platform designed to enhance patient care by leveraging advanced technologies like Artificial Intelligence and Machine Learning. Our goal is to connect patients with medical professionals and healthcare facilities through a personalized, data-driven approach.

```
</p>
```

```
<p>
```

Our system analyzes patient medical histories, symptoms, and case details to identify similar cases from an extensive database. This empowers doctors with valuable insights, helping them make more informed decisions and offer precise diagnoses.

```
</p>
```

```
<p>
```

We collaborate with some of the best hospitals and doctors to ensure that every patient receives exceptional care. By bridging the gap between patients and top medical professionals, we strive to make healthcare

more accessible, efficient, and effective.

</p>

<p>

Our core values include:

</p>

<ul>

<li><strong>Innovation:</strong> Continuously improving through technology.</li>

<li><strong>Compassion:</strong> Putting patients first in every decision.</li>

<li><strong>Excellence:</strong> Partnering with leading healthcare providers.</li>

<li><strong>Integrity:</strong> Maintaining transparency and trust.</li>

</ul>

<p>

Thank you for choosing <strong>Patient Case Similarity System</strong>. Together, we aim to redefine healthcare and save lives.

</p>

</div>

</div>

</section>

</body>

</html>

## **Locateus.html**

<!DOCTYPE html>

<html lang="en">

<head>

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hospitals and Doctors for Heart Diseases</title>
<style>
  body {
    background-image:
url('https://t3.ftcdn.net/jpg/07/86/34/68/360_F_786346890_E4BqkCpTHWZg4
LIcBnqG2MdyB1Afxz1z.jpg');
    background-size: cover;
    font-family: Arial, sans-serif;
    color: white;
  }

  h1, h2 {
    color: white;
    text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.6);
  }

  h1 {
    font-size: 48px;
    font-weight: bold;
    text-align: center;
    margin-top: 20px;
  }

  h2 {
    font-size: 36px;
    font-weight: bold;
    text-align: center;
```

```
}

.rib {
    background-color: #333;
    overflow: hidden;
    display: block;
    width: 615px;
    margin: 0;
    padding: 0;
    height: 60px;
    overflow: auto;
    margin: 20px auto;
    border-radius: 10px;
}

.rib a {
    float: left;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    font-size: 25px;
}

.rib a:hover {
    background-color: #ddd;
    color: black;
}
```

```
.rib a.active {
  background-color: #04AA6D;
  color: white;
}

.hospital-container {
  width: 80%;
  margin: 20px auto;
  background-color: rgba(0, 0, 0, 0.6);
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 15px rgba(0, 0, 0, 0.5);
  text-align: center;
}

.hospital-img {
  width: 100%;
  height: 300px;
  object-fit: cover;
  border-radius: 10px;
  margin-bottom: 15px;
}

.address {
  font-size: 18px;
  margin: 10px 0;
}

ul {
```

```
text-align: left;
list-style-type: none;
padding-left: 0;
}
```

```
ul li {
margin: 5px 0;
font-size: 18px;
}
```

```
h3 {
font-size: 20px;
margin-top: 10px;
font-weight: bold;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>Patient Case Similarity System </h1>
```

```
</header>
```

```
<section class="hospitals">
```

```
<div class="rib">
```

```
<center>
```

```
<a href="./mainpage.html">Home</a>
```

```
<a href="./about.html">About Us</a>
```

```
<a href="./locateus.html">Find a location</a>
```

```
<a href="/registrationpage.html">Patient Portal</a>
</center>
</div>

<div class="hospital-container">
  
  <h2>Manipal Hospital</h2>
  <p class="address">Malleswaram West, Bangalore</p>
  <h3>Doctors in Manipal Hospital:</h3>
  <ul>
    <li>Dr. Prabhakara Shetty Heggunde - Cardiologist</li>
    <li>Dr. Mohammed Rehan Sayeed - Cardiothoracic Surgeon</li>
    <li>Dr. Srikanth Vijaysimha - General Surgeon</li>
  </ul>
  <h3>Rating: 4.5/5</h3>
</div>

<div class="hospital-container">
  
  <h2>Aster Cmi Hospital</h2>
  <p class="address">Sahakara Nagar, Bangalore</p>
  <h3>Doctors in Aster Cmi Hospital:</h3>
  <ul>
    <li>Dr. Arul D Furtado - Cardiac Surgeon</li>
    <li>Dr. Ganesh Krishnan Iyer - Cardiothoracic Surgeon</li>
    <li>Dr. Y M Prashanth - Cardiothoracic and Vascular Surgeon</li>
  </ul>
```

<h3>Rating: 4/5</h3>

</div>

<div class="hospital-container">



<h2>Apollo Hospital</h2>

<p class="address">Bannerghatta Road, Bangalore</p>

<h3>Doctors in Apollo Hospital:</h3>

<ul>

<li>Dr. Raghavendra Chikkatur - Cardiothoracic Surgeon</li>

<li>Dr. Umesh Satish Gheewala - Vascular Surgeon</li>

</ul>

<h3>Rating: 4.5/5</h3>

</div>

<div class="hospital-container">



<h2>Trilife Hospital</h2>

<p class="address">Kalyan Nagar, Bangalore</p>

<h3>Doctors in Trilife Hospital:</h3>

<ul>

<li>Dr. Sudhakar P - Cardiologist</li>

<li>Dr. Prashant Wankhade - Cardiac Surgeon</li>

<li>Dr. Soorampallay Vijay - Cardiologist</li>

</ul>

<h3>Rating: 4.5/5</h3>

</div>



```
<div class="hospital-container">
  
  <h2>Manipal Hospital</h2>
  <p class="address">Millers Road, Bangalore</p>
  <h3>Doctors in Manipal Hospital:</h3>
  <ul>
    <li>Dr. Girish Godbole - Cardiac Surgeon</li>
    <li>Dr. Praveen Kumar A V - Cardiac Surgeon</li>
    <li>Dr. G Sridhara - Cardiac Surgeon</li>
  </ul>
  <h3>Rating: 4.5/5</h3>
</div>

<div class="hospital-container">
  
  <h2>Tathagat Heart Hospital</h2>
  <p class="address">Crescent Road, Bangalore</p>
  <h3>Doctors in Tathagat Heart Hospital:</h3>
  <ul>
    <li>Dr. Mahantesh R Charantimath - Cardiologist</li>
  </ul>
  <h3>Rating: 4/5</h3>
</div>
</section>
</body>
</html>
```

**Registrationpage.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Patient Registration Form</title>
  <script src="./app.js"></script>
  <style>
    body {
      background-image:
url('https://t3.ftcdn.net/jpg/07/86/34/68/360_F_786346890_E4BqkCpTHWZg4
LIcBnqG2MdyB1Afxz1z.jpg');
      background-size: cover;
      font-family: Arial, sans-serif;
      color: white;
    }

    .rib {
      background-color: #333;
      overflow: hidden;
      display: block;
      width: 615px;
      margin: 20px auto;
      height: 60px;
      border-radius: 10px;
    }
```

```
.rib a {  
    float: left;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
    font-size: 25px;  
}  
  
.rib a:hover {  
    background-color: #ddd;  
    color: black;  
}  
  
.rib a.active {  
    background-color: #04AA6D;  
    color: white;  
}  
  
h1, h2 {  
    text-align: center;  
    color: white;  
    text-shadow: 2px 2px 8px rgba(0, 0, 0, 0.6);  
}  
  
h1 {  
    font-size: 48px;  
    font-weight: bold;  
}
```

```
h2 {
  font-size: 36px;
  font-weight: bold;
}

.container {
  width: 50%;
  margin: 50px auto;
  background-color: rgba(0, 0, 0, 0.7);
  padding: 20px;
  border-radius: 10px;
}

form {
  width: 80%;
  margin: 0 auto;
}

input[type="text"], input[type="number"], input[type="tel"],
input[type="submit"] {
  width: 100%;
  padding: 10px;
  margin: 10px 0;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-sizing: border-box;
}
```

```
input[type="submit"] {
    background-color: #007bff;
    color: white;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #0056b3;
}
</style>
</head>
<body>
    <center>
        <h1>Patient Case Similarity System</h1>
    </center>

    <div class="rib">
        <a href="./mainpage.html">Home</a>
        <a href="./about.html">About Us</a>
        <a href="./locateus.html">Find a location</a>
        <a href="./registrationpage.html">Patient Portal</a>
    </div>

    <center>
        <h2>Registration Form</h2>
    </center>

    <div class="container">
        <form name="registrationForm" onsubmit="return submitForm(event)">
```

<label for="name">Patient Name:</label>

<input type="text" id="name" name="name" required>

<label for="id">Patient ID:</label>

<input type="number" id="id" name="id" required>

<label for="phone">Phone Number:</label>

<input type="tel" id="phone" name="phone" required>

<label for="age">Age:</label>

<input type="number" id="age" name="age" required>

<label for="sex">Sex:</label>

<input type="text" id="sex" name="sex" required>

<label for="cigsPerDay">Cigarette Consumption Per Day:</label>

<input type="number" id="cigsPerDay" name="cigsPerDay" required>

<label for="BPMeds">Blood Pressure Medications Taken:</label>

<input type="number" id="BPMeds" name="BPMeds" required>

<label for="prevalentStroke">Prevalent Stroke:</label>

<input type="number" id="prevalentStroke" name="prevalentStroke"  
required>

<label for="prevalentHyp">Hypertension:</label>

<input type="number" id="prevalentHyp" name="prevalentHyp"  
required>

```
<label for="diabetes">Diabetes:</label>
<input type="number" id="diabetes" name="diabetes" required>

<label for="totChol">Cholesterol Level:</label>
<input type="number" id="totChol" name="totChol" required>

<label for="sysBP">Systolic Blood Pressure:</label>
<input type="number" id="sysBP" name="sysBP" required>

<label for="diaBP">Diastolic Blood Pressure:</label>
<input type="number" id="diaBP" name="diaBP" required>

<label for="BMI">Body Mass Index:</label>
<input type="number" id="BMI" name="BMI" required>

<label for="heartRate">Heart Rate:</label>
<input type="number" id="heartRate" name="heartRate" required>

<label for="glucose">Glucose Level:</label>
<input type="number" id="glucose" name="glucose" required>

<input type="submit" value="Submit">
</form>
</div>
</body>
</html>
```

**app.js**

```
function submitForm(event) {
```

```
event.preventDefault();
console.log("called here")
let name = document.forms["registrationForm"]["name"].value;
let id = document.forms["registrationForm"]["id"].value;
let phone = document.forms["registrationForm"]["phone"].value;
let age = document.forms["registrationForm"]["age"].value;
let sex = document.forms["registrationForm"]["sex"].value;
let cigsPerDay = document.forms["registrationForm"]["cigsPerDay"].value;
let BPMeds = document.forms["registrationForm"]["BPMeds"].value;
let prevalentStroke =
document.forms["registrationForm"]["prevalentStroke"].value;
let prevalentHyp =
document.forms["registrationForm"]["prevalentHyp"].value;
let diabetes = document.forms["registrationForm"]["diabetes"].value;
let totChol = document.forms["registrationForm"]["totChol"].value;
let sysBP = document.forms["registrationForm"]["sysBP"].value;
let diaBP = document.forms["registrationForm"]["diaBP"].value;
let BMI = document.forms["registrationForm"]["BMI"].value;
let heartRate = document.forms["registrationForm"]["heartRate"].value;
let glucose = document.forms["registrationForm"]["glucose"].value;

if (name == "" || id == "" || phone == "" || age == "" || sex == "" ||
    cigsPerDay == "" || BPMeds == "" || prevalentStroke == "" ||
    prevalentHyp == "" || diabetes == "" || totChol == "" ||
    sysBP == "" || diaBP == "" || BMI == "" || heartRate == "" || glucose == "")
{
    alert("All fields must be filled out.");
    return false;
}
```



```
}
```

```
if (isNaN(id) || isNaN(phone) || isNaN(age)) {  
    alert("Patient ID, Phone, and Age must be numbers.");  
    return false;  
}
```

```
let patientData = {  
    name: name,  
    id: id,  
    phone: phone,  
    age: age,  
    sex: sex,  
    cigsPerDay: cigsPerDay,  
    BPMeds: BPMeds,  
    prevalentStroke: prevalentStroke,  
    prevalentHyp: prevalentHyp,  
    diabetes: diabetes,  
    totChol: totChol,  
    sysBP: sysBP,  
    diaBP: diaBP,  
    BMI: BMI,  
    heartRate: heartRate,  
    glucose: glucose  
};
```

```
fetch('http://127.0.0.1:5000/predict', {  
    method: 'POST',  
    mode: 'cors',
```

```
headers: {
  'Content-Type': 'application/json'
},
body: JSON.stringify(patientData)
})
.then(response => response.json())
.then(data => {
  let resultDiv = document.createElement('div');
  resultDiv.style.padding = "20px";
  resultDiv.style.backgroundColor = "#444";
  resultDiv.style.marginTop = "20px";
  resultDiv.style.color = "white";
  resultDiv.innerHTML = "<h3>Most Similar Patients:</h3>";

  let similarPatients = data;
  for (let patientId in similarPatients) {
    resultDiv.innerHTML += `<p>Patient ID: ${patientId}, Similarity:
${similarPatients[patientId]}</p>`;
  }
  document.body.appendChild(resultDiv);
})
.catch(error => {
  console.error("Error:", error);
  alert("Error ");
});
}

document.addEventListener('DOMContentLoaded', () => {
});
```

---

## BACKEND

### Heart\_diseases\_capstone.ipynb

```
import pandas as pd
test = pd.read_csv('/content/test.csv')
train = pd.read_csv('/content/train.csv')
data = pd.concat([test,train])
data.head()
data.info()
data.isnull().sum()
#filling all the null values with their median
for column in ['cigsPerDay', 'BPMeds', 'totChol', 'BMI', 'heartRate', 'glucose']:
    data[column].fillna(data[column].median(), inplace=True)
data.isnull().sum()
data.info()
data.describe()
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print(data.columns)
plt.figure(figsize=(10,5))
plt.bar(data['age'],data['heartRate'])
plt.xlabel('Age of patient')
plt.ylabel('Heart Rate of patient')
plt.title('Age of patient VS their HeartRate')
plt.show()
plt.hist(data['heartRate'], bins=10, edgecolor='black')
plt.xlabel('Heart Rate (bpm)')
plt.ylabel('Frequency')
plt.title('Distribution of Heart Rate')
```

```
plt.show()
sns.pairplot(data)
plt.show()
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer
y=train['TenYearCHD']
X = train[data.columns.difference(['TenYearCHD'])]
original_columns = X.columns
cat_features = X.select_dtypes(include=['object', 'category']).columns.tolist()
# One-hot encoding
X_encoded = pd.get_dummies(X, columns=cat_features, drop_first=True)
imputer = SimpleImputer(strategy='median')
X_encoded = imputer.fit_transform(X_encoded)
X_encoded = pd.DataFrame(X_encoded, columns=original_columns)
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.3,
random_state=42)
log=LogisticRegression(max_iter=100)
log.fit(X_train,y_train)
log_y_pred=log.predict(X_test)
log_accuracy=accuracy_score(y_test,log_y_pred)
print("The accuracy score of the Logistic Model is ",log_accuracy)
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
dt_y_pred= dt.predict(X_test)
dt_accuracy = accuracy_score(y_test, dt_y_pred)
```

```
print("Decision Tree Accuracy is ", dt_accuracy)
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf_y_pred = rf.predict(X_test)
rf_accuracy= accuracy_score(y_test, rf_y_pred)
print("Random Forest Accuracy is ", rf_accuracy)
from sklearn.metrics import precision_score, recall_score, f1_score
p=precision_score(y_test, rf_y_pred)
r=recall_score(y_test, rf_y_pred)
f1=f1_score(y_test, rf_y_pred)
print("The precision score is",p)
print("The recall score is",r)
print("The f1 score is",f1)
#K-Means Clustering
from sklearn.cluster import KMeans
n=3
k=KMeans(n_clusters=n,random_state=42)
k.fit(X_encoded)
label=k.labels_
X_encoded['Cluster']=label
print(X_encoded.groupby('Cluster').mean())
plt.scatter(X_encoded['age'], X_encoded['totChol'], c=label)
plt.xlabel('Age of patient')
plt.ylabel('Total Cholesterol')
plt.title("KMeans Clustering of Patient's age and their cholestrol")
plt.show()
#Silhoutte Score - assessing the appropriateness of the cluster results
from sklearn.metrics import silhouette_score
s=silhouette_score(X_encoded,label)
```

```
print("The average silhouette score is :",s)
#Checking other possibilities for silhouette score
for n in range(2,11):
    k=KMeans(n_clusters=n,random_state=42)
    k.fit(X_encoded)
    l=k.labels_
    s=silhouette_score(X_encoded,l)
    print("For cluster = ",n," the average silhouette score is ",s)
```

### **test.py**

```
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.metrics.pairwise import cosine_similarity
from flask import Flask, jsonify, request
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

def preprocess_data():
    test = pd.read_csv('./test.csv')
    train = pd.read_csv('./train.csv')
    data = pd.concat([test, train])
    data = data.drop(columns=["TenYearCHD", 'is_smoking',
'education'])

    for column in ['cigsPerDay', 'BPMeds', 'totChol', 'BMI', 'heartRate',
```

```
'glucose']:
```

```
    data[column] = data[column].fillna(data[column].median())
```

```
    return data, train
```

```
def encode_data(X, original_columns):
```

```
    for column in X:
```

```
        if column in ['age', 'cigsPerDay', 'BPMeds', 'prevalentStroke',  
'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
```

```
                    'diaBP', 'BMI', 'heartRate', 'glucose']:
```

```
            X[column] = pd.to_numeric(X[column], errors='coerce')
```

```
    cat_features = X.select_dtypes(include=['object',  
'category']).columns.tolist()
```

```
    X_encoded = pd.get_dummies(X, columns=cat_features,  
drop_first=True)
```

```
    imputer = SimpleImputer(strategy='median')
```

```
    X_encoded_imputed = imputer.fit_transform(X_encoded) # This  
will return a numpy.ndarray
```

```
    X_encoded_imputed = pd.DataFrame(X_encoded_imputed,  
columns=X_encoded.columns)
```

```
    return X_encoded_imputed
```

```
def encode_input_data(input_data, X_encoded):
```

```
    input_df = pd.DataFrame([input_data])
```

```
    for column in input_df.columns:
```

```
        if column in ['age', 'cigsPerDay', 'BPMeds', 'prevalentStroke',
```

```
'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
    'diaBP', 'BMI', 'heartRate', 'glucose']:
    input_df[column] = pd.to_numeric(input_df[column],
errors='coerce')

cat_features = input_df.select_dtypes(include=['object',
'category']).columns.tolist()

input_df_encoded = pd.get_dummies(input_df,
columns=cat_features, drop_first=True)

input_df_encoded =
input_df_encoded.reindex(columns=X_encoded.columns,
fill_value=0)

input_df_encoded = pd.DataFrame(input_df_encoded,
columns=X_encoded.columns)

return input_df_encoded

def find_similar_patients(input_data, X_encoded):
    input_df_encoded = encode_input_data(input_data, X_encoded)
    combined_data = pd.concat([X_encoded,
input_df_encoded],ignore_index=True)
    print(combined_data)
    sim_matrix = cosine_similarity(combined_data)
    sim_df = pd.DataFrame(sim_matrix, index=combined_data.index,
columns=combined_data.index)
```



```
patient_id = len(X_encoded)
similarities =
sim_df.iloc[patient_id].sort_values(ascending=False).iloc[1:6]
return similarities
```

```
@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    input_data = {
        'BMI': data['BMI'],
        'BPMeds': data['BPMeds'],
        'age': data['age'],
        'cigsPerDay': data['cigsPerDay'],
        'diaBP': data['diaBP'],
        'diabetes': data['diabetes'],
        'glucose': data['glucose'],
        'heartRate': data['heartRate'],
        'id': data['id'],
        'prevalentHyp': data['prevalentHyp'],
        'prevalentStroke': data['prevalentStroke'],
        'sex': data['sex'],
        'sysBP': data['sysBP'],
        'totChol': data['totChol'],
    }
    pushing_data = {
```

```
'id': data['id'],
'age': data['age'],
'education': 1,
'sex': data['sex'],
'is_smoking': 'NO',
'cigsPerDay': data['cigsPerDay'],
'BPMed': data['BPMed'],
'prevalentStroke': data['prevalentStroke'],
'prevalentHyp': data['prevalentHyp'],
'diabetes': data['diabetes'],
'totChol': data['totChol'],
'sysBP': data['sysBP'],
'diaBP': data['diaBP'],
'BMI': data['BMI'],
'heartRate': data['heartRate'],
'glucose': data['glucose'],
'TenYearCHD': 0,
}
data, train = preprocess_data()
y = train['TenYearCHD']
X = train[data.columns.difference(['TenYearCHD'])]
original_columns = X.columns
X_encoded = encode_data(X, original_columns)
similar_patients = find_similar_patients(input_data, X_encoded)
print(similar_patients)
```

```
try:
    existing_data = pd.read_csv('./train.csv')
    csv_columns = existing_data.columns.tolist()
    new_patient      =      pd.DataFrame([pushing_data],
columns=csv_columns)

    df_complete=pd.concat([existing_data,new_patient],axis=0)
    df_complete.to_csv('./train.csv', index=False)

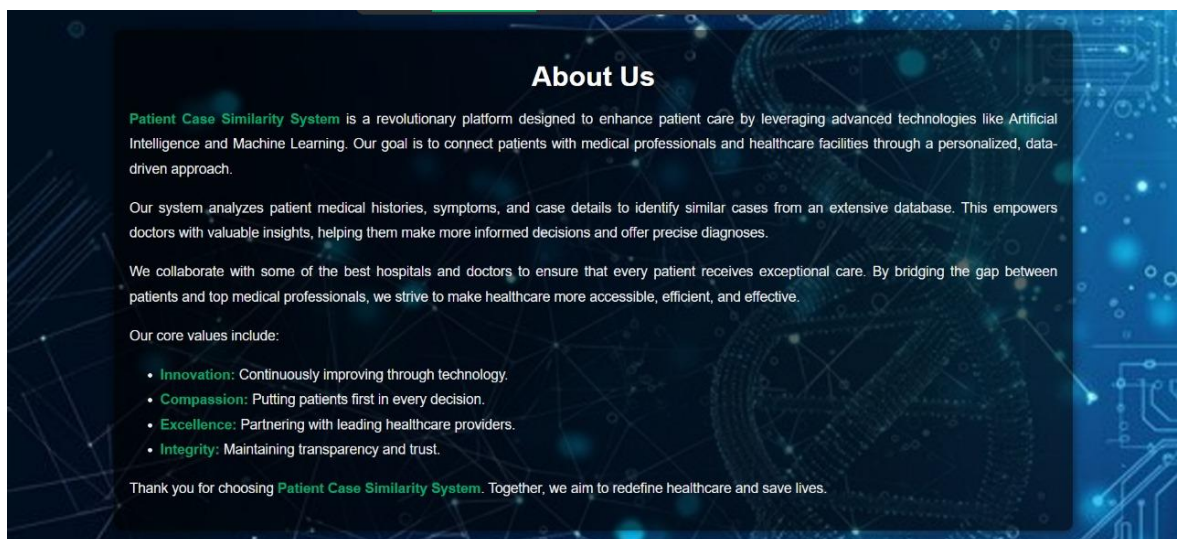
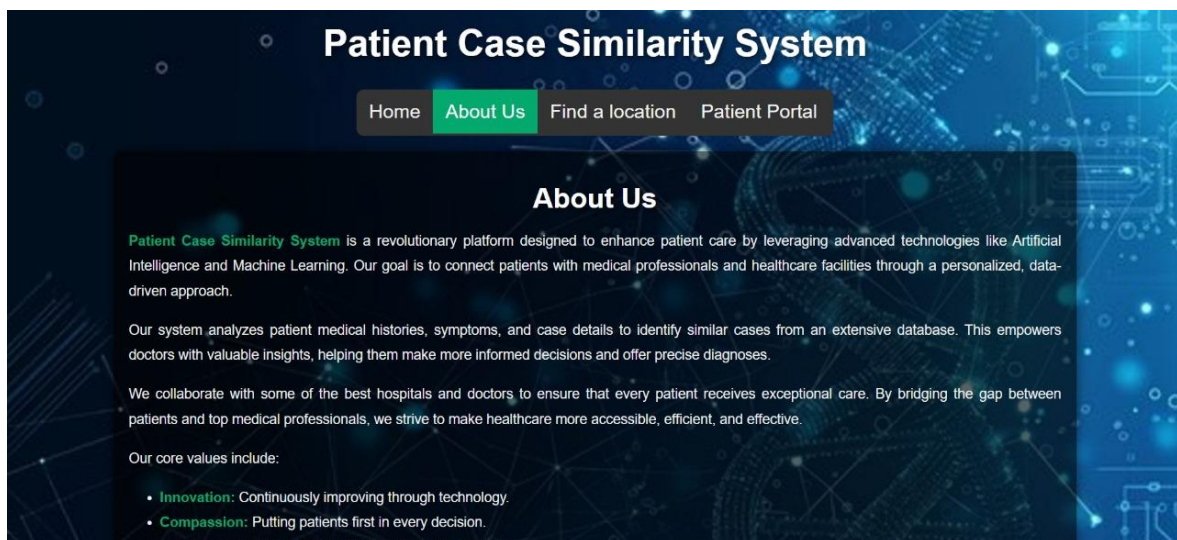
except Exception as e:
    return jsonify({'error': str(e)}), 500
except Exception as e:
    return jsonify({'error': str(e)}), 500

return jsonify(similar_patients.to_dict())

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

## APPENDIX-B

### SCREENSHOTS



## Patient Case Similarity System

[Home](#) [About Us](#) [Find a location](#) [Patient Portal](#)



### Manipal Hospital

Malleswaram West, Bangalore

Doctors in Manipal Hospital:



### Manipal Hospital

Malleswaram West, Bangalore

Doctors in Manipal Hospital:

Dr. Prabhakara Shetty Heggunde - Cardiologist  
Dr. Mohammed Rehan Sayeed - Cardiothoracic Surgeon  
Dr. Srikanth Vijaysimha - General Surgeon

Rating: 4.5/5



### Aster Cmi Hospital

Sahakara Nagar, Bangalore

Doctors in Aster Cmi Hospital:

Dr. Arul D Furtado - Cardiac Surgeon  
Dr. Ganesh Krishnan Iyer - Cardiothoracic Surgeon  
Dr. Y M Prashanth - Cardiothoracic and Vascular Surgeon

Rating: 4/5





### Apollo Hospital

Bannerghatta Road, Bangalore

Doctors in Apollo Hospital:

Dr. Raghavendra Chikkatur - Cardiothoracic Surgeon  
Dr. Umesh Satish Gheewala - Vascular Surgeon

Rating: 4.5/5



### Trilife Hospital

Kalyan Nagar, Bangalore

Doctors in Trilife Hospital:

Dr. Sudhakar P - Cardiologist  
Dr. Prashant Wankhade - Cardiac Surgeon  
Dr. Soorampallay Vijay - Cardiologist

Rating: 4.5/5



### Manipal Hospital

Millers Road, Bangalore

Doctors in Manipal Hospital:

Dr. Girish Godbole - Cardiac Surgeon  
Dr. Praveen Kumar A V - Cardiac Surgeon  
Dr. G Sridhara - Cardiac Surgeon

Rating: 4.5/5

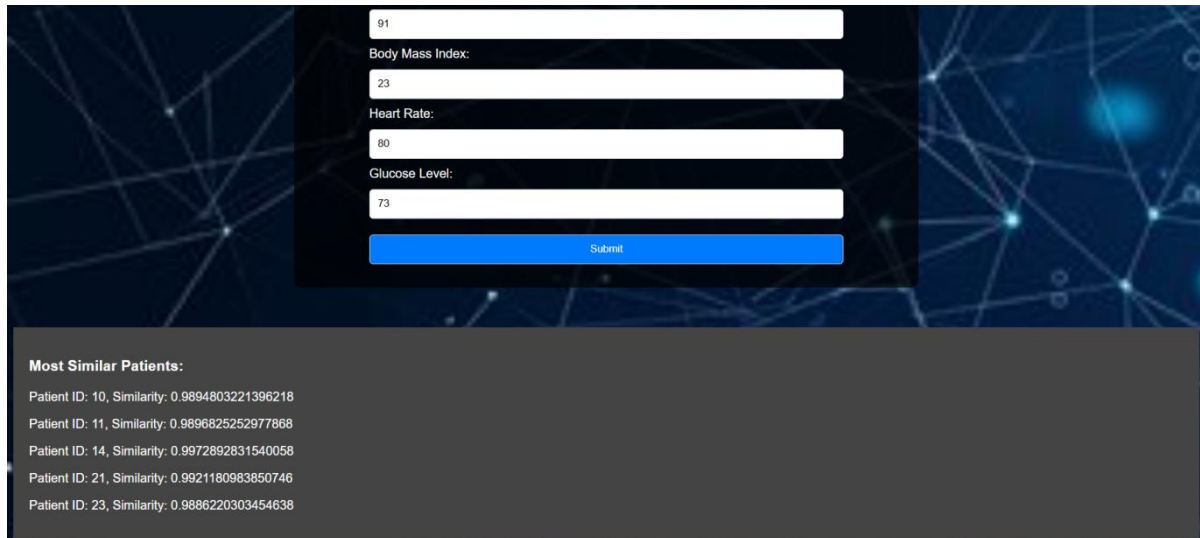


The screenshot shows the "Patient Case Similarity System" registration form. The form is titled "Registration Form" and has a navigation bar with links: "Home", "About Us", "Find a location", and "Patient Portal". The form fields are as follows:

Field Label	Value
Patient Name:	N
Patient ID:	78
Phone Number:	8898745632
Age:	50
Sex:	M

The screenshot shows the continuation of the "Patient Case Similarity System" registration form. The form fields are as follows:

Field Label	Value
Cigarette Consumption Per Day:	1
Blood Pressure Medications Taken:	3
Prevalent Stroke:	0
Hypertension:	0
Diabetes:	1
Cholesterol Level:	150
Systolic Blood Pressure:	125
Diastolic Blood Pressure:	91
Body Mass Index:	23



A web form for patient case similarity. The form has a dark blue background with a network diagram. It contains four input fields for patient data: Patient ID (91), Body Mass Index (23), Heart Rate (80), and Glucose Level (73). A blue 'Submit' button is at the bottom of the input section. Below the input section, a dark grey box titled 'Most Similar Patients:' lists five patient IDs and their similarity scores.

91

Body Mass Index:

23

Heart Rate:

80

Glucose Level:

73

Submit

**Most Similar Patients:**

Patient ID: 10, Similarity: 0.9894803221396218

Patient ID: 11, Similarity: 0.9896825252977868

Patient ID: 14, Similarity: 0.9972892831540058

Patient ID: 21, Similarity: 0.9921180983850746

Patient ID: 23, Similarity: 0.9886220303454638



## **APPENDIX-C**

### **ENCLOSURES**

**1. Journal publication/Conference Paper Presented Certificates of all students.**

**2. Include certificate(s) of any Achievement/Award won in any project-related event.**

**3. Similarity Index / Plagiarism Check report clearly showing the Percentage (%). No need for a page-wise explanation.**

#### 4. Details of mapping the project with the Sustainable Development Goals (SDGs).



- Our project is mapped to SDG-3, that is, Good health and Well Being.
- Our project contributes to the improvement of people's health and enhancing their quality of life.
- By getting the similarity score between historical patients and the new patients, we make it easier to cluster patients based on heart diseases.
- Improving the diagnostic accuracy and predictive models, we ensure healthy lives and promoting well-being for people at all ages.