PROJECT TITLE :
A System for Predicting and Alerting Patients of Potential Heart Attack

GROUP MEMBERS :
1) VAIBHAV GUPTA B22CS058
2) AMBATI RAHUL REDDY B22CS088

1 SOFTWARE REQUIREMENT SPECIFICATIONS

1.1 Introduction

1.1.1 Purpose:

- The purpose of software we designed is to predict and alert patients about potential heart attacks based on **environmental data and personal health metrics** to provide proactive management and support for individuals living with risk of heart attack.
- By analyzing factors such as environmental conditions and individual health data, the software offers early warnings of impending heart attack episodes, allowing patients to take preemptive measures and mitigate the risk of heart failure.
- This not only enhances the quality of life for individuals with risk of heart attack by reducing the uncertainty and fear associated with unpredictable attacks but also fosters greater awareness and understanding of its triggers and symptoms.
- Furthermore, the software facilitates remote monitoring and communication between patients and healthcare providers, enabling timely intervention and personalized care.
- Through data-driven insights and continuous health tracking, the software empowers patients to make informed decisions about their health and includes healthcare professionals to optimize treatment plans and support strategies, ultimately improving heart management and patient outcomes.

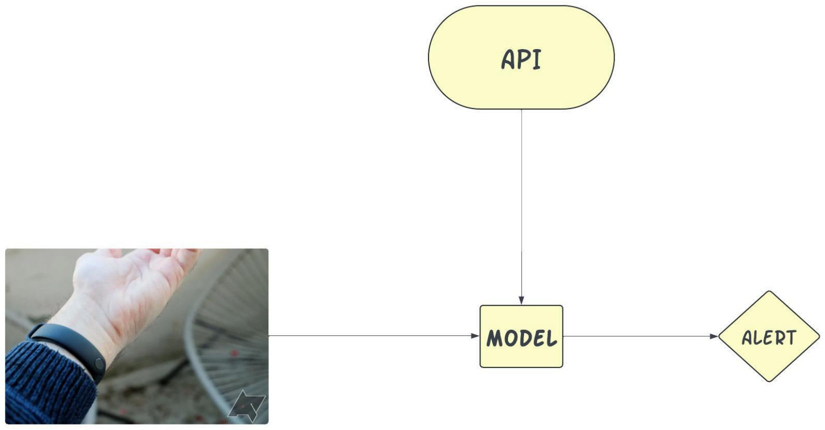
1.1.2 Scope:

The scope of project encompasses the development of a comprehensive system consisting of both **hardware and software** components. It involves the integrating environmental data of that particular place in which the user is residing along with his current health conditions with which our software would predict if he would get an heart attack or not.

1.2 System Description

1.2.1 System Overview

- The Heart Attack Prediction System encompasses its core functionalities, components, and interactions.
- At its core, the system integrates **environmental data, personal health metrics, and user-specific information** to predict and alert potential risk patients about heart attacks.
- It consists of several key components including a **Smart Band for health data collection, an API for fetching environmental data, a prediction engine for analyzing and forecasting heart risks, and a notification service for alerting patients**.
- The system collects user details such as heart rate, high blood pressure, and cholesterol levels to personalize predictions and recommendations.
- Through continuous monitoring and analysis, the system provides early warnings, suggests preventive measures, and facilitates remote communication between patients and healthcare providers.
- Its overall goal is to empower patients with proactive tools and insights for better risk management, ultimately enhancing their quality of life and well-being.



1.2.2 Functional Requirements

1. User Profile Management :
   - Users shall have the ability to create and update profiles.
   - Profile updates should encompass basic details such as age, sex, Email-id, habits of smoking or drinking and other health details of the user.
2. Heath Database :
   - The system must store historical health and environmental data for analysis and trend identification.
   - Enable users and healthcare providers to review past data to identify triggers, patterns, and trends related to heart attacks.
3. Data Collection and Integration :
   - Collect and integrate health data from wearable devices like Smart Bands, including metrics such as heart rate, blood pressure, and exercise hours per week.
4. Data analysis and Prediction :
   - Analyze collected health and geographic data to predict the likelihood of heart attacks.
   - Utilize algorithms to assess patterns, trends, and correlations to forecast potential heart failure episodes.
5. Risk Assessment and Alerting :
   - Evaluate the risk level associated with predicted heart attacks based on analyzed data.
   - If the risk is high, trigger alerts and notifications to inform users about the potential heart attack, along with recommended preventive measures like contacts of nearest hospitals.
6. Notification System :
   - Users shall receive notifications via Email, providing information such as title, risk, and preventive measures.
   - Notifications can be customized to suit user preferences.

1.3 Non-Functional Requirements

1.3.1 Performance

- The system should respond promptly to user interactions and data processing tasks, ensuring **minimal latency and efficient resource utilization** even during peak usage periods.

1.3.2 Usability

- The system interface should be **intuitive, user-friendly,** and accessible to individuals of varying technical proficiency and physical abilities. It should prioritize ease of navigation, clear presentation of information, and logical workflow.

1.3.3 Security

- The system should implement robust security measures to protect sensitive user data, including health information and personal details, from unauthorized access, disclosure, alteration, or destruction. This includes **encryption, access controls, and secure transmission protocols.**

1.4 System Constraints

1.4.1 Hardware

- The system needs devices such as Smart Band or some other wearable gadgets to get the current users health details and also a computer or any system for giving alerts and running the software.

1.4.2 Connectivity

- The system must have a **stable internet connection** for getting the API data and the real-time geographical updates and also Bluetooth or any other thing to connect the smart band to the system.

1.5 Conclusion

The Heart Attack prediction System is an integrated software solution designed to proactively predict and alert risky individuals about potential heart attacks based on environmental data and personal health metrics. It collects data from various sources, including wearable devices like Smart Bands for health monitoring and external API for environmental data. By analyzing these data streams, the system generates personalized predictions of heart attack risks and provides timely alerts to users, suggesting preventive measures to mitigate potential risks. The system also supports remote monitoring and communication between users and healthcare providers, enabling timely intervention and optimized health management strategies. With a user-friendly interface and robust connectivity, the Heart Attack Prediction System empowers individuals with risk to take proactive control of their health and improve their overall quality of life.

2 UML Diagrams

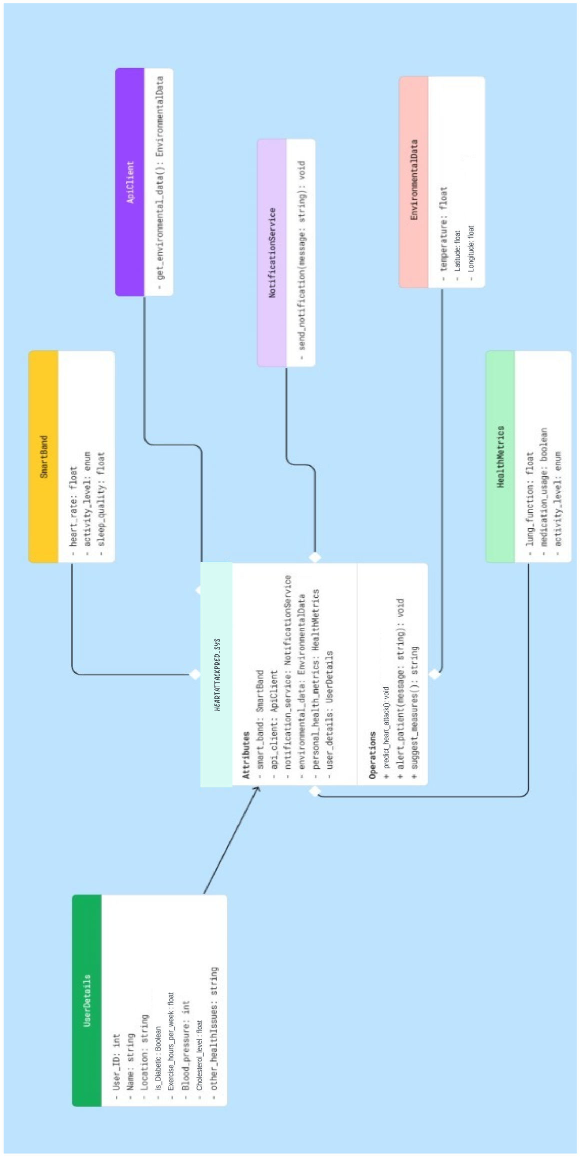

SmartBand
- heart_rate: float
- activity_level: enum
- sleep_quality: float
ApiClient
- get_environmental_data(): EnvironmentalData
NotificationService
- send_notification(message: string): void
EnvironmentalData
- temperature: float
- Latitude: float
- Longitude: float
HealthMetrics
- lung_function: float
- medication_usage: boolean
- activity_level: enum
HEARTATTACKPRED_SYS
Attributes
- smart_band: SmartBand
- api_client: ApiClient
- notification_service: NotificationService
- environmental_data: EnvironmentalData
- personal_health_metrics: HealthMetrics
- user_details: UserDetails
Operations
+ predict_heart_attack(): void
+ alert_patient(message: string): void
+ suggest_measures(): string
UserDetails
- User_ID: int
- Name: string
- Location: string
- Is_Diabetic: Boolean
- Exercise_hours_per_week: float
- Blood_pressure: int
- Cholesterol_level: float
- other_healthIssues: string

Figure 1: class diagram

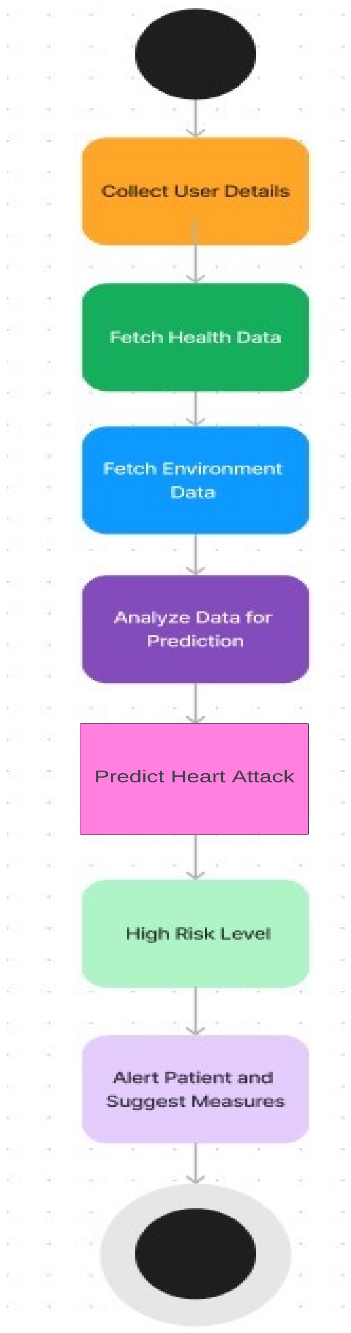


Figure 2: activity diagram

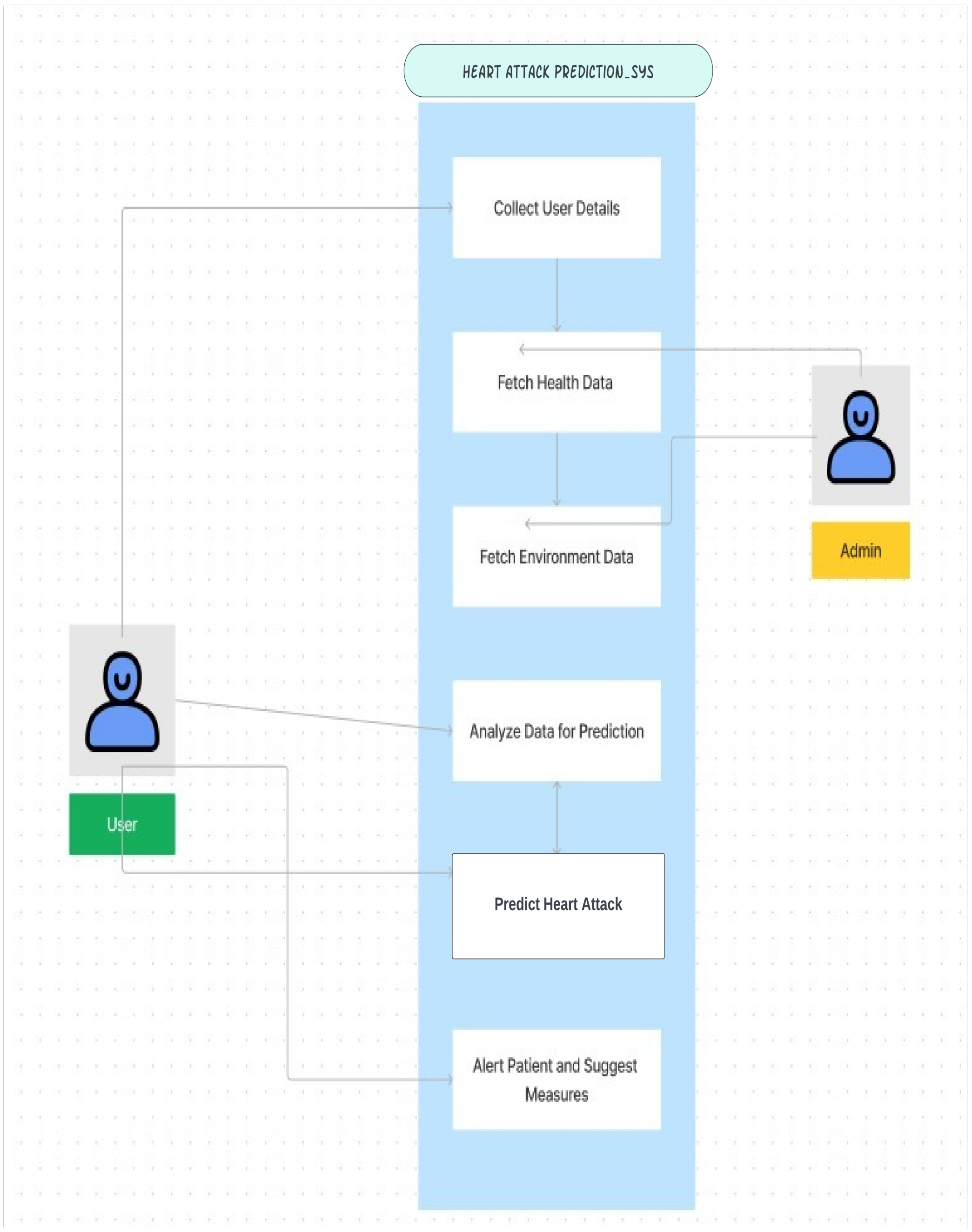

HEART ATTACK PREDICTION_SYS
Collect User Details
Fetch Health Data
Fetch Environment Data
Analyze Data for Prediction
Predict Heart Attack
Alert Patient and Suggest Measures
Admin
User

Figure 3: usecase diagram

3 SOFTWARE ARCHITECTURE AND DESIGN

3.1 Software Architecture

3.1.1 Introduction

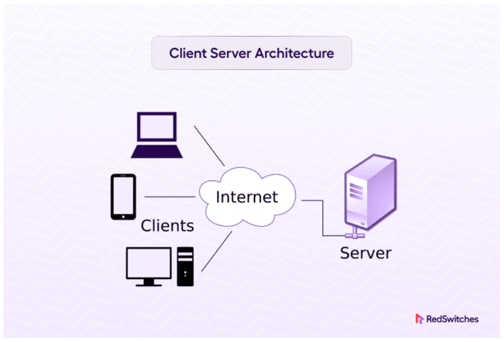
The purpose of this is to analyze and propose suitable software architectural patterns for the development of a system that analyzes the current user health details and the environment in order to alert an risky patients of an heart attack.

3.1.2 Architectural Patterns Selection

After careful consideration, the Client-Server Architecture is selected as the best architectural patterns for this project.

Description :

It is a computing model where tasks or processes are divided between clients and servers. In this architecture, clients are devices or applications that request services or resources from servers, which are powerful computers or software applications that provide those services. Clients initiate requests, such as accessing files, retrieving data, or processing information, and servers respond to these requests by providing the requested resources or executing the requested tasks. This architecture facilitates distributed computing, where tasks are distributed across multiple interconnected devices, allowing for scalability, efficiency, and centralized management of resources and data.



Application to the Project :

In the client-server architecture, the heart attack prediction system operates with the server hosting essential functionalities such as data processing, analysis, and storage. Clients, which can be mobile applications on users' devices, interact with the server to access the system's features. Users input personal health data through the client application, which the server analyzes to predict potential heart attacks based on environmental and health metrics. The server sends back alerts and recommendations to the client application, notifying users of potential risks and suggesting preventive measures. Communication between clients and the server occurs over secure network connections using standard protocols like HTTP or HTTPS, ensuring data integrity and confidentiality. This architecture enables distributed computing, scalability, and efficient resource management while providing users with timely insights and support for managing heart risks effectively.

3.1.3 APPLICATION OF CLIENT- SERVER ARCHITECTURAL PATTERN AS EXAMPLE:

1. Server Side

   - The server hosts core functionalities of the Heart Attack Prediction System, including data processing, analysis, and storage.
   - It communicates with external API to fetch environmental data and integrates with databases to store user profiles, health metrics, and system configurations.
   - The server handles client requests, performs predictive analytics based on the received data, and generates alerts when potential heart attacks are detected.

2. Client Side

   - Clients, such as mobile applications installed on users' smartphones or tablets, interact with the server to access the Heart Prediction System's features.
   - Users input personal health data through the client application, which is transmitted to the server for analysis.
   - The client application receives alerts and recommendations from the server, notifying users of potential heart attack risks and suggesting preventive measures.
   - Users can view their health data, receive real-time alerts, and communicate with healthcare providers through the client application interface.

3. Communication

   - Communication between clients and the server occurs over a network, typically the internet, using standard communication protocols such as HTTP or HTTPS.
   - Clients send requests to the server, which processes the requests, performs necessary computations, and sends back responses containing alerts, recommendations, or other relevant information.
   - Secure communication protocols ensure data integrity and confidentiality during transmission between clients and the server.

3.2 Design Considerations

1. Centralized management:
    - The server hosts the core functionalities of the system, including data processing, analysis, and storage. This centralization allows for easier management and maintenance of the system, as updates and changes can be applied centrally on the server side.
2. Resource Sharing:
    - With a client-server architecture, resources such as databases, processing power, and storage can be shared among multiple clients. This enables efficient resource utilization and reduces redundancy in hardware and software components.
3. Scalability:
    - Client-server architecture allows for easy scalability by adding more clients or servers as the system grows. Additional clients can connect with the server to access the system's features, while more servers can be added to handle increased computational loads and user requests.
4. Reliability and Fault Tolerance:
    - By distributing tasks between clients and servers, client-server architecture enhances system reliability and fault tolerance. If one server fails, other servers can continue to handle client requests, ensuring uninterrupted service availability.
5. Security:
    - Client-server architecture enables centralized security measures to be implemented on the server side, including access controls, encryption, and authentication mechanisms. This helps protect sensitive user data and ensures secure communication between clients and the server.

3.3 Conclusion

In summary, the Heart Attack Prediction System follows a client-server architecture where server hosts the system's core functionalities and data processing capabilities, while clients interact with the system through client applications to receive alerts, view health data, and communicate with healthcare providers. This architecture allows for distributed computing, scalability, and efficient management of resources while providing users with timely insights and support for heart failure risks management.

4 MODEL TRAINING AND DEPLOYMENT

4.1 Data Preprocessing and Visualization :

The data set that we have used has been taken from kaggle. :
`https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset?resource=download`

- At first we looked into what are the various features that are present in the dataset – 'Patient ID', 'Age', 'Sex', 'Cholesterol', 'Blood Pressure', 'Heart Rate', 'Diabetes', 'Family History', 'Smoking', 'Obesity', 'Alcohol Consumption', 'Exercise Hours Per Week', 'Diet', 'Previous Heart Problems', 'Medication Use', 'Stress Level', 'Sedentary Hours Per Day', 'Income', 'BMI', 'Triglycerides', 'Physical Activity Days Per Week', 'Sleep Hours Per Day', 'Country', 'Continent', 'Hemisphere', 'Heart Attack Risk'

- In all, it contains 26 columns out of which there the column named "Heart Attack Risk" which serves as our target.We then tested to find if there is missing data in the dataset which came out to be 0%.

- Next up, we did a few data plots: refer figure 5 in next page.Then we dropped the comparatively lesser important and the complex ones as well.

- Then, we did web scraping and found out the average temp of each country as after doing research we found temperature variation may lead to heart attack risk.Finally, we are left with 15 columns.

- As an example lets consider,
  Country: Nigeria, Average Yearly Temperature: 27.00 °C (80.60 °F)
  Country: Sri Lanka, Average Yearly Temperature: 26.95 °C (80.51 °F)
  Country: Indonesia, Average Yearly Temperature: 25.85

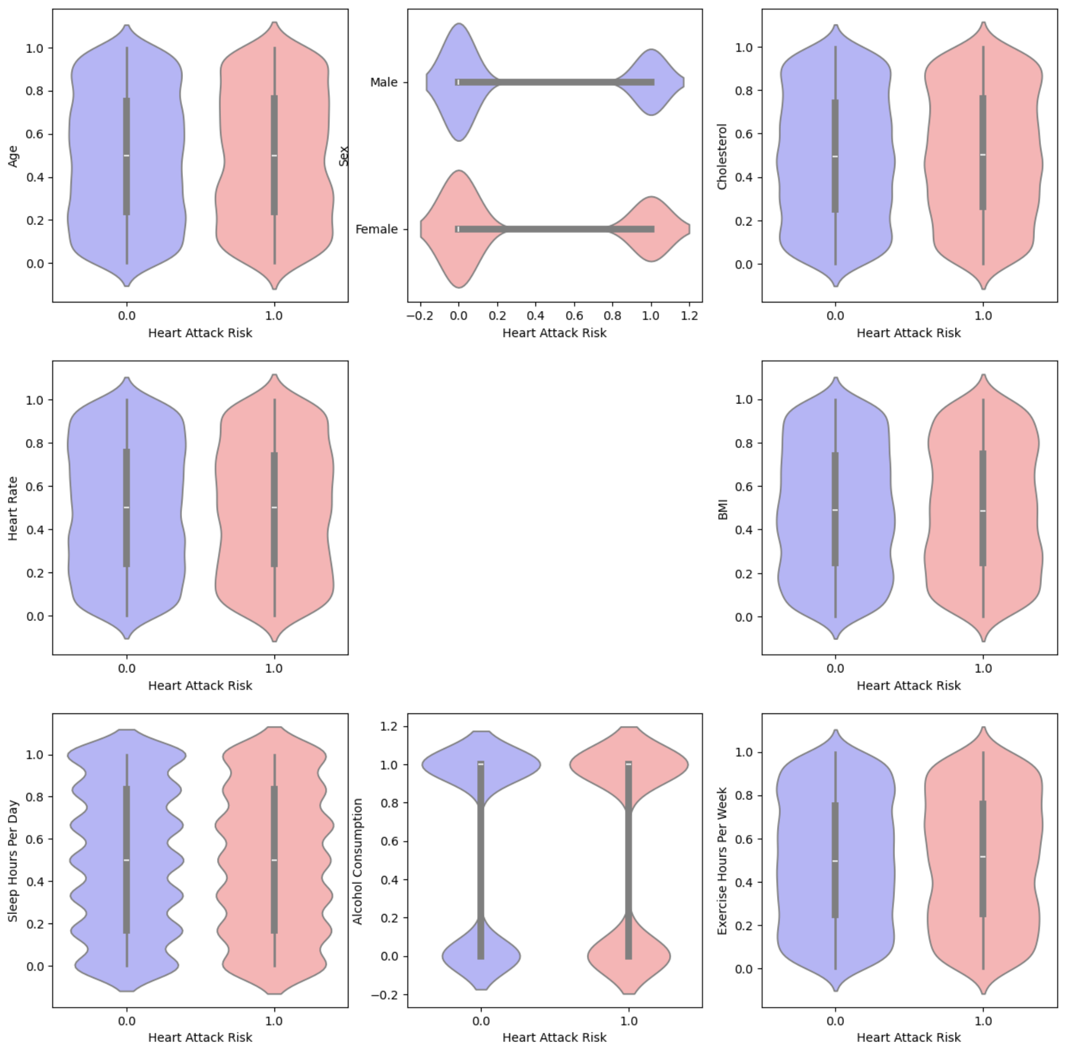


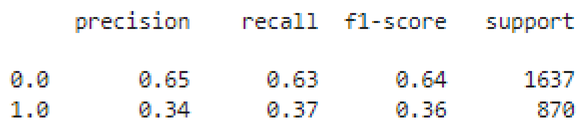
Figure 4: Outliers in Numerical Features

4.2 Model Selection

We have used a 70-30 train-test split in this case and trained the data accordingly.

4.2.1 Decision Tree :

- A Decision Tree is a hierarchical model that makes sequential decisions based on features to classify or regress data, often used for its simplicity and interpretability in machine learning tasks.
- The accuracy observed via this model is 53.6% approximately.

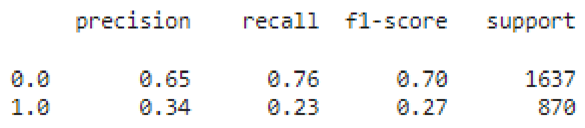


```
          precision    recall  f1-score   support

     0.0       0.65      0.63      0.64      1637
     1.0       0.34      0.37      0.36       870
```

Figure 5: classification report for 4.2.1

4.2.2 K-Nearest Neighbours :

- K-Nearest Neighbors (KNN) is a simple yet effective algorithm for classification and regression tasks. It predicts the label of a data point by averaging the labels of its k nearest neighbors, where "k" is a user-defined parameter.
- The accuracy observed via this model is 57.47% approximately.

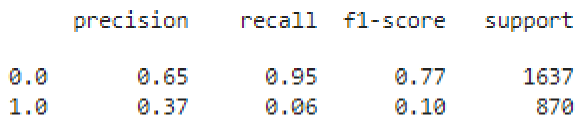


```
          precision    recall  f1-score   support

     0.0       0.65      0.76      0.70      1637
     1.0       0.34      0.23      0.27       870
```

Figure 6: classification report for 4.2.2

4.2.3 Random Forest Classification :

- Random Forest Classification is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of individual trees. It improves accuracy and reduces overfitting by averaging the results of many decision trees.
- The accuracy observed via this model is 63.98% approximately.

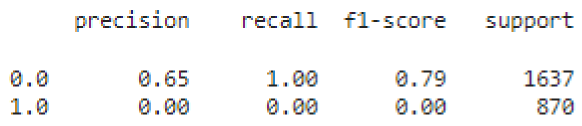


```
              precision    recall  f1-score   support

         0.0       0.65      0.95      0.77      1637
         1.0       0.37      0.06      0.10       870
```

Figure 7: classification report for 4.2.3

4.2.4 Support Vector Machines(SVM/SVC) :

- Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It finds the optimal hyperplane that best separates the data points into different classes while maximizing the margin between classes, making it effective in high-dimensional spaces.
- The accuracy observed via this model is 65.29% approximately.



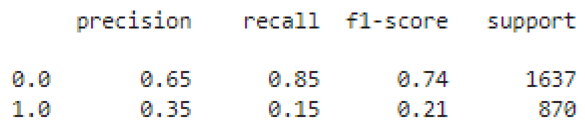
```
              precision    recall  f1-score   support

         0.0       0.65      1.00      0.79      1637
         1.0       0.00      0.00      0.00       870
```

Figure 8: classification report for 4.2.4

4.2.5 Bagging :

- Bagging (Bootstrap Aggregating) is an ensemble learning technique that combines multiple models (often decision trees) trained on different subsets of the training data using bootstrap sampling. It reduces variance and helps prevent overfitting by averaging or voting the predictions of individual models to make the final prediction.

- The accuracy observed via this model is 60.78% approximately.



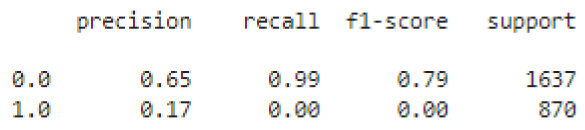
```
              precision    recall  f1-score   support

         0.0       0.65      0.85      0.74      1637
         1.0       0.35      0.15      0.21       870
```

Figure 9: classification report for 4.2.5

4.2.6 Neural Networks :

- Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They consist of interconnected nodes organized in layers, where each node performs a simple computation. Through training on labeled data using algorithms like backpropagation, neural networks learn to make predictions or decisions based on input features, often excelling in complex pattern recognition tasks.

- The accuracy observed via this model is 64.97% approximately.



```
              precision    recall  f1-score   support

         0.0       0.65      0.99      0.79      1637
         1.0       0.17      0.00      0.00       870
```

Figure 10: classification report for 4.2.6

4.3 Model Selected :

We tried the models finally decided to use the Random Forest Classification to be our models base as it came out to be the best over a various number of times.

5 Software Metrics

1. ABC Metric:
   - A metric that evaluates the complexity of assignments, branches, and conditions in code.
   - Given our code, ABC Metric came out to be **194**.
2. Comment density:
   - Measures the ratio of comments to code in a software component.
   - Given our code, Comment density: came out to be **0.2756183745583039**.
3. Cyclomatic complexity:
   - Measures the complexity of a program by counting the number of linearly independent paths through the code.
   - Given our code, Cyclomatic complexity came out to be **257** .
4. Source lines of code:
   - Measures the number of lines of code in a software component.
   - Given our code, SLOC came out to be **281**.
5. Program execution time:
   - Measures the time taken by a program to execute.
   - Given our code, execution time came to be **52.319993019104004**.

Colab Link in which we have trained our models and made report on :
`https://colab.research.google.com/drive/1fP7-01IkTMU_ncQmc3CJMmgfit6EgUHJ?usp=sharing`

6 Website Development and Navigation

- So basically, our website can be divided into three main sections :
    1. The first would be the information section which contains few details about Heart Attack, its various causes and the preventive measures to avoid heart attacks. It also contains a portion of bibliography and also displays the email-id and phone-no of PHC.
    2. The second portion includes a form which accepts various required information from the user.
    3. The last section contains the results of prediction.
- The form requests input details such as age, sex, cholesterol levels, bmi etc. It also asks permission to use location so that we can check the latitude & longitude and then call an API to get city.
- If it cannot be loaded an option to manually enter city is available. Now, from the city we call an API to find the current temperature of that place which we then use to predict the risk of heart attack.
- If the data entered is incomplete or wrong a popup page which says wrong data appears.

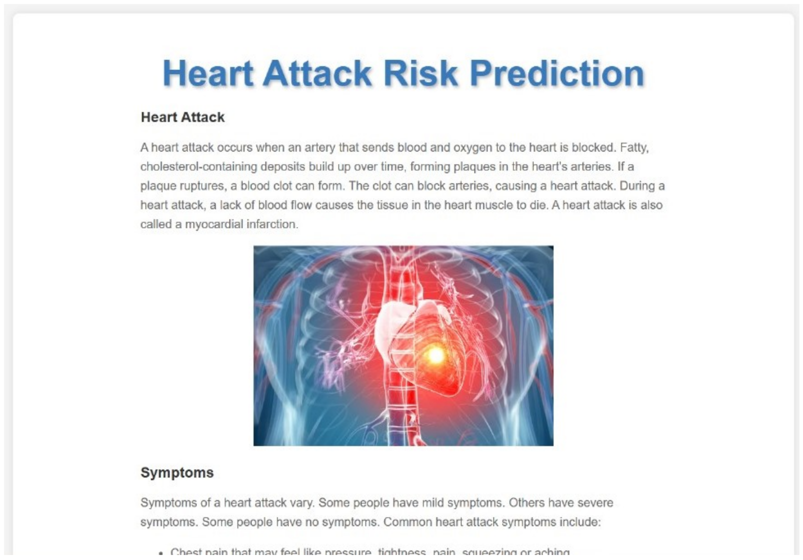
We have used the following links in order to find the current location and temperature.

1) `http://api.openweathermap.org/data/2.5/weather`

2) `https://nominatim.openstreetmap.org/reverse?format=json&lat=${latitude}&lon=${longitude}`

- After predicting, we display the results and if the result comes out to be yes, there is a risk of heart attack, then an email is also sent to the person notifying him/her about the risk.

6.1 Glimpses Of Website and its Workings



Heart Attack Risk Prediction

Heart Attack

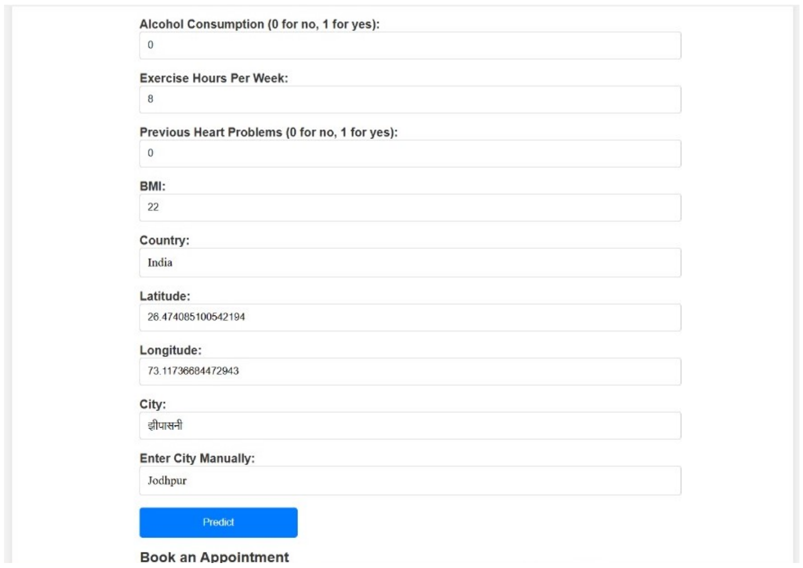
A heart attack occurs when an artery that sends blood and oxygen to the heart is blocked. Fatty, cholesterol-containing deposits build up over time, forming plaques in the heart's arteries. If a plaque ruptures, a blood clot can form. The clot can block arteries, causing a heart attack. During a heart attack, a lack of blood flow causes the tissue in the heart muscle to die. A heart attack is also called a myocardial infarction.

Symptoms

Symptoms of a heart attack vary. Some people have mild symptoms. Others have severe symptoms. Some people have no symptoms. Common heart attack symptoms include:

- Chest pain that may feel like pressure, tightness, pain, squeezing or aching

Figure 11: Home Page



Alcohol Consumption (0 for no, 1 for yes): 0

Exercise Hours Per Week: 8

Previous Heart Problems (0 for no, 1 for yes): 0

BMI: 22

Country: India

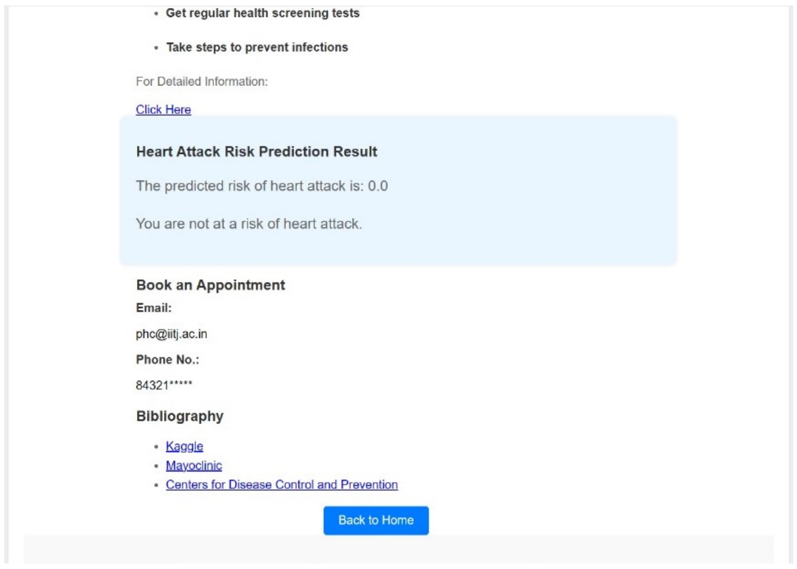
Latitude: 26.474085100542194

Longitude: 73.11736684472943

City: झीपासनी

Enter City Manually: Jodhpur

Figure 12: Input 1



- Get regular health screening tests
- Take steps to prevent infections

For Detailed Information:

Click Here

Heart Attack Risk Prediction Result

The predicted risk of heart attack is: 0.0

You are not at a risk of heart attack.

Book an Appointment

Email:

phc@iitj.ac.in

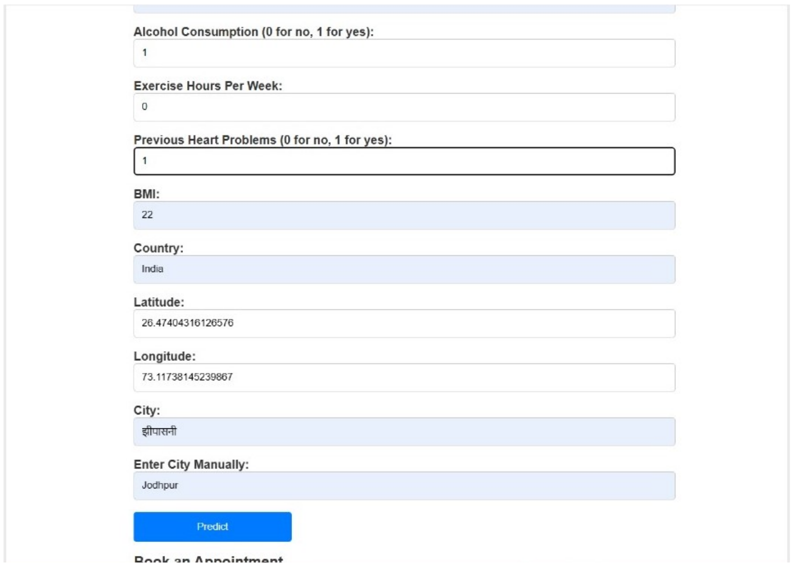
Phone No.:

84321*****

Bibliography

- Kaggle
- Mayoclinic
- Centers for Disease Control and Prevention

Figure 13: Output 1



Alcohol Consumption (0 for no, 1 for yes): 1

Exercise Hours Per Week: 0

Previous Heart Problems (0 for no, 1 for yes): 1

BMI: 22

Country: India

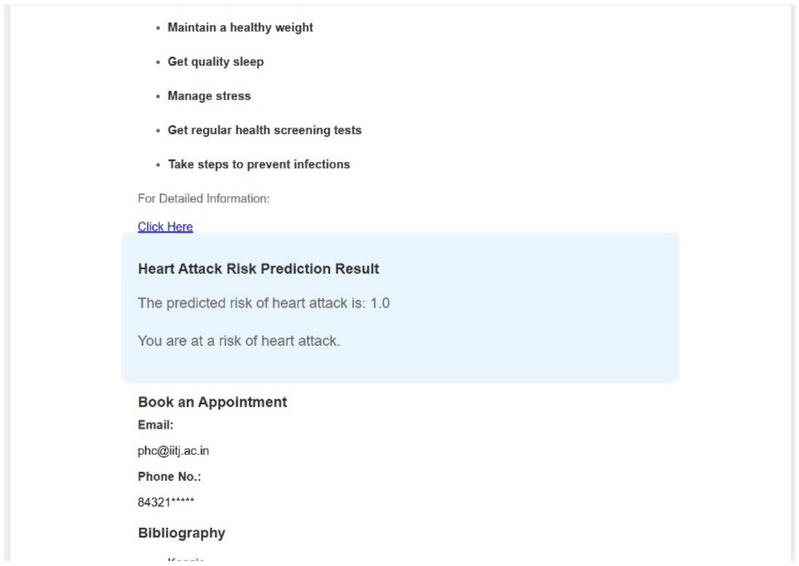
Latitude: 26.47404316126576

Longitude: 73.11738145239867

City: झीपासनी

Enter City Manually: Jodhpur

Figure 14: Input 2



- Maintain a healthy weight
- Get quality sleep
- Manage stress
- Get regular health screening tests
- Take steps to prevent infections

For Detailed Information:

Click Here

Heart Attack Risk Prediction Result

The predicted risk of heart attack is: 1.0

You are at a risk of heart attack.

Book an Appointment

Email:

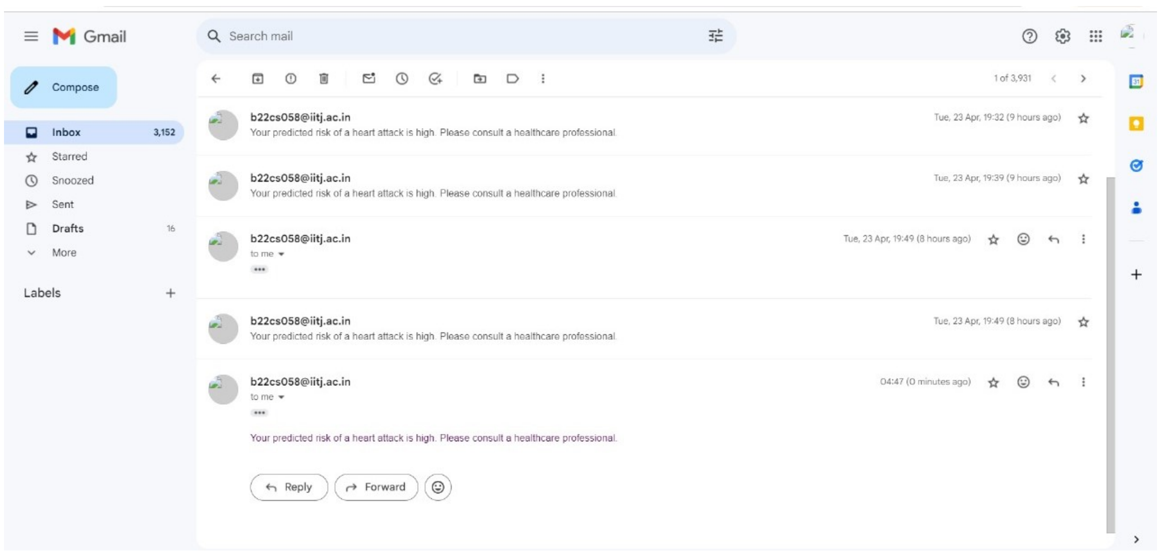
phc@iitj.ac.in

Phone No.:

84321*****

Bibliography

Figure 15: Output 2



b22cs058@iitj.ac.in
Your predicted risk of a heart attack is high. Please consult a healthcare professional.
Tue, 23 Apr, 19:32 (9 hours ago)

b22cs058@iitj.ac.in
Your predicted risk of a heart attack is high. Please consult a healthcare professional.
Tue, 23 Apr, 19:39 (9 hours ago)

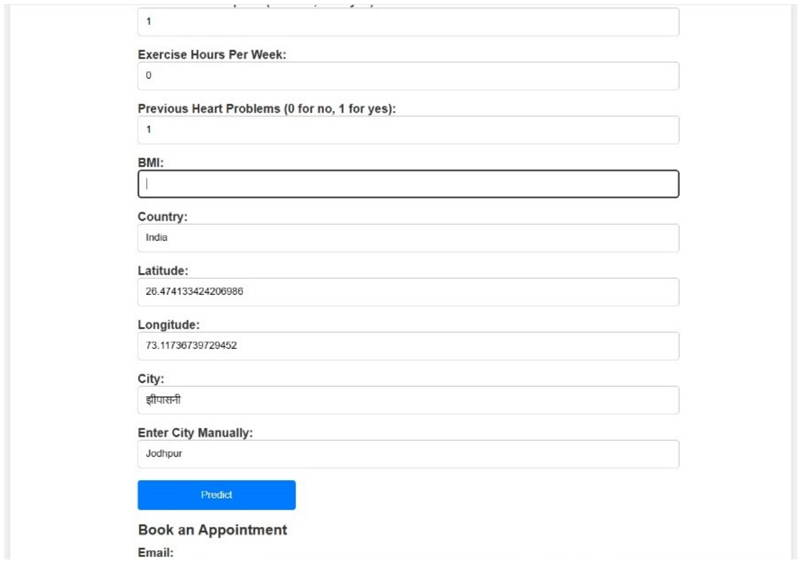
b22cs058@iitj.ac.in
to me
Tue, 23 Apr, 19:49 (8 hours ago)

b22cs058@iitj.ac.in
Your predicted risk of a heart attack is high. Please consult a healthcare professional.
Tue, 23 Apr, 19:49 (8 hours ago)

b22cs058@iitj.ac.in
to me
04:47 (0 minutes ago)

Your predicted risk of a heart attack is high. Please consult a healthcare professional.

Figure 16: Email Notification



1

Exercise Hours Per Week:
0

Previous Heart Problems (0 for no, 1 for yes):
1

BMI:
|

Country:
India

Latitude:
26.474133424206986

Longitude:
73.11736739729452

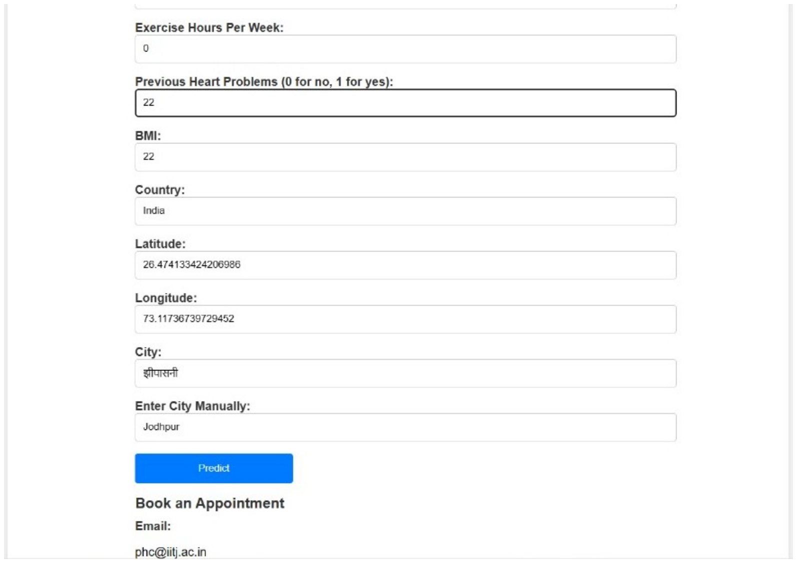
City:
झीपासनी

Enter City Manually:
Jodhpur

Book an Appointment

Email:

Figure 17: Data incomplete



Exercise Hours Per Week:
0

Previous Heart Problems (0 for no, 1 for yes):
22

BMI:
22

Country:
India

Latitude:
26.474133424206986

Longitude:
73.11736739729452

City:
झीपासनी

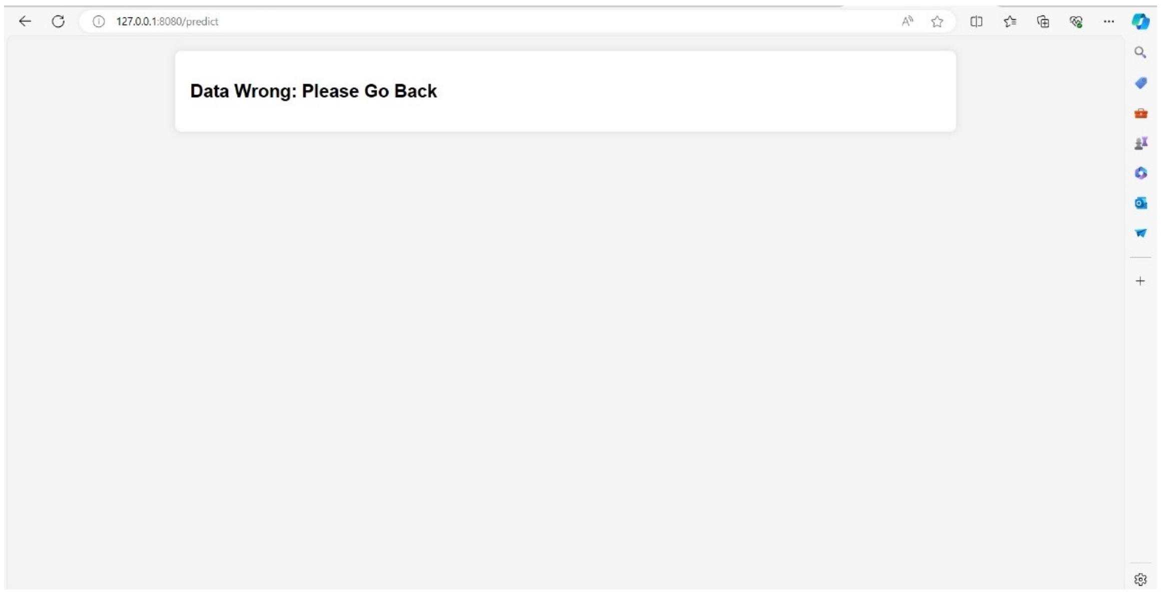
Enter City Manually:
Jodhpur

Book an Appointment

Email:

phc@iitj.ac.in

Figure 18: Data Incorrect



Data Wrong: Please Go Back

Figure 19: Error

7 SOFTWARE TESTING

1. random test cases :

   - Code used for testing is given below

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import random

# Function to generate random test cases for all test cases
def generate_random_test_cases(features, n_cases=10):
random_cases = []
for _ in range(n_cases):
case = []
for feature in features.columns:
if features[feature].dtype == 'object':
unique_values = features[feature].unique()
random_value = random.choice(unique_values)
case.append(random_value)
else:
min_val = features[feature].min()
max_val = features[feature].max()
random_value = random.uniform(min_val, max_val)
case.append(random_value)
random_cases.append(case)
return pd.DataFrame(random_cases, columns=features.columns)

# Assuming 'df' is your dataset
# Drop rows with missing values
df = df.dropna()

# Define features and target variable
features = df.drop('Heart␣Attack␣Risk', axis=1)
target = df['Heart␣Attack␣Risk']

# Split the dataset into training and testing sets
# (70% training, 30% testing)
X_train, X_test, y_train, y_test =
         train_test_split(features, target, test_size=0.3,
          random_state=62)

# Generate random test cases for the first 10 test cases
random_test_cases = generate_random_test_cases(X_test, n_cases=10)

# Predict on random test cases using the already trained classifier
y_pred_random = rf_clf.predict(random_test_cases)

# Create a DataFrame to display input data, predicted output,
# and actual output for those 10 cases
random_test_df =
         pd.DataFrame(random_test_cases, columns=X_test.columns)
random_test_df['Predicted␣Output'] = y_pred_random

# Get the true labels for the first 10 random test cases
true_labels = y_test[:10].reset_index(drop=True)
random_test_df['Actual␣Output'] = true_labels

# Calculate accuracy on these 10 random cases
accuracy_random = accuracy_score(true_labels, y_pred_random)

# Display the accuracy
print("Accuracy␣on␣10␣random␣cases:", accuracy_random)
```

- Accuracy on 10 random cases: 0.6

2. <u>Incomplete form :</u>

   - When any of the input slot is left unfilled we were getting the output be 0 always we fixed that by popping and error page.

3. <u>Wrong data input :</u>

   - When any of the input slot is left wrong or out of available range we addressed user to fill meaningful data using a popup page showing error.