TABLE OF CONTENTS

LIST OF FIGURES

PROBLEM STATEMENT

In the present scenario, the problem of ECG classification is tackled with the help of the doctor. This process is very time-consuming and tedious. Therefore, an IoT-based heart monitoring system will be very helpful which is user friendly and provides quick results which helps in analyzing the irregularities in the heart rate.

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION OF THE PROJECT

A heart rate monitoring system provides accurate measurements of heart rate samples along with the time stamp information. This technology enables individuals to monitor their heart rate in various contexts, such as during exercise, stress, or leisure activities, allowing for timely interventions when necessary. The combination of IoT technology and Artificial Intelligence allows the detection of abnormalities. This system also emphasis on user friendly data visualization approaches which not only enhances accessibility but also makes it well-suited for users.

The Internet of Things (IoT) provides a valuable approach for heart rate monitoring, delivering real-time data to the users. The Heart Rate Monitoring System uses Arduino, NodeMCU, and a pulse rate sensor to track heart rate efficiently and accurately. The core of our system is the Arduino, a user-friendly microcontroller that processes signals from the heart. Heart rate ranges vary throughout individuals in different age groups. Here we are creating a dataset that classifies heart rate data based on age, gender, and heartbeat characteristics. By collecting data during various activities, the paper aims to provide comprehensive insights into irregularities of heart rate across different demographic groups. By considering the unique heart rate ranges of various age groups and genders, we can understand individual heart rate patterns. The incorporation of AI-driven decision tree algorithms further enhances the functionality and utility of the heart rate monitoring system. By employing these advanced algorithms, the system can intelligently classify the sensed data into various categories.

Overall, heart rate monitoring systems play a crucial role in detecting irregularities by leveraging IoT and AI technologies. The integration of these technologies has revolutionized the field, making it not only more effective but also user-friendly. This advancement enhances accessibility, enabling individuals to actively monitor and manage irregularities in their heart health with greater ease and precision.

1.2 OBJECTIVES OF THE PROJECT

The primary objectives of our project is:

1. To enhance healthcare monitoring for patients by providing continuous, real-time heart rate monitoring that detects irregularities in the heart rate and allows for medical intervention when necessary.
2. To classify the detected irregularities of heart rate based on the AI technique.

CHAPTER 2

LITERATURE SURVEY

The paper titled "Secured Pulse Rate Monitoring System using IoT and Cloud" by Aritra Ray and Hena Ray [1] addresses the critical need for continuous monitoring of heart rate, especially for individuals at high risk of cardiovascular diseases. It proposes a system that utilizes Internet of Things (IoT) technology and cloud computing to monitor heart rates in real-time. The system ensures data privacy through end-to-end encryption using AES 128 scheme and facilitates communication through an Android application platform. The paper extensively discusses the prevalence of heart diseases globally and highlight the importance of timely monitoring, especially considering the significant percentage of deaths attributed to cardiovascular diseases. The proposed system architecture involves data acquisition units equipped with heartbeat sensors, data encryption using AES scheme, and real-time transmission to the medical cloud for processing and alert generation. The results indicate a high level of accuracy in monitoring heart conditions, validated against electrocardiography (ECG) results. Overall, the paper provides a comprehensive overview of a secure and efficient pulse rate monitoring system using IoT and cloud technologies, with promising results validated through experimentation.

The paper titled "IoT based system for Heart Rate Monitoring and Heart Attack Detection" [2] presents a comprehensive overview of the application of Internet of Things (IoT) technology in monitoring heart health. It introduces a prototype system that utilizes NodeMCU, pulse sensors, Adafruit, and Blynk cloud to monitor heartbeats in real-time and detect abnormalities. The system integrates GPS technology for live location tracking and employs a local server for security and privacy. By storing patient data in the cloud, the system facilitates future studies on heart health. The paper discusses various heart-related diseases, their causes, and the normal heart rate ranges for different age groups. Additionally, it describes the software and hardware components used in the implementation of the proposed system, including Adafruit and Blynk platforms, NodeMCU, and pulse sensors. The paper concludes by highlighting the advantages of their system, such as real-time monitoring, location tracking, and immediate medical assistance during emergencies.

The literature survey presented in the paper titled "Heart Attack Recognition and Heart Rate Monitoring System Using IoT" by S. Gopi and Dr. E. Punarselvam[3] explores the intersection of Internet of Things (IoT) technology with healthcare applications, particularly focusing on patient monitoring and health management. The proposed system architecture involves the deployment of various sensing devices, such as the Pulse Sensor Amped, which can collect real-time data on heart rate. These sensors are connected to microcontrollers like NodeMCU ESP8266 WiFi Module, facilitating wireless transmission of data to a centralized data center accessible via the internet. This setup enables remote monitoring of a person's health status without the need for physical presence. The paper emphasizes the significance of IoT integration with other technologies, such as Structural Health Monitoring (SHM) frameworks, to enhance the scalability and versatility of healthcare solutions. Moreover, it discusses potential future enhancements for the system, indicating ongoing advancements in IoT-based healthcare applications. In conclusion, the literature survey sheds light on current trends, challenges, and opportunities in IoT-enabled healthcare systems. It provides valuable insights for researchers and practitioners, guiding future research endeavours in this dynamic and rapidly evolving field.

The paper titled "IoT based System for Heart Rate Monitoring" by Sahana S Khamitkar and Prof. Mohammed Rafi [4] presents a comprehensive exploration of heart rate monitoring using IoT technology. The system aims to detect heartbeats and monitor heart rates to assess the risk of heart attacks and enable regular checkups. Utilizing Photoplethysmography (PPG) as a non-invasive technique, the system captures heart rate signals from patients' fingertips. These signals are then transmitted wirelessly via Bluetooth to a computer or Android application for further analysis and monitoring. The integration of IoT in healthcare allows for remote patient monitoring, facilitating real-time assessment of heart health and timely intervention if needed. The system's potential extends to telemedicine applications, enabling clinicians to access patient data remotely and make informed decisions. Additionally, the paper discusses the importance of continuous monitoring in managing heart-related diseases and proposes future enhancements for the system, such as incorporating alarm systems for abnormal heart rates and logging heart rate measurements for trend analysis. Overall, the study underscores the significance of IoT in revolutionizing healthcare delivery, particularly in the domain of cardiovascular health monitoring.

The paper titled "Arduino based Wireless Heart-rate Monitoring system with Automatic SOS Message and/or Call facility using SIM900A GSM Module" [5] presents a novel approach to addressing the critical issue of monitoring heart rates, particularly in scenarios where individuals may require immediate medical attention, such as senior citizens. The system integrates several key components, including the ARDUINO Lilypad microcontroller, pulse sensor, ARDUINO UNO, SIM900A GSM module, and RF transceiver module, to create a wearable and wireless monitoring solution. By leveraging the capabilities of the Lilypad microcontroller and Pulse sensor, the system can detect abnormal heart rates and initiate appropriate responses, such as sending SOS messages or making calls to predefined mobile numbers. One of the notable aspects of this paper is its emphasis on efficiency and simplicity, as all computations and data transmission processes are handled within the microcontrollers, without the need for additional processing units. The use of RF modules for wireless communication reduces costs and dependencies compared to other connectivity options like Bluetooth or internet-based solutions. Additionally, the system offers flexibility in notification methods, allowing for both message alerts and call-rings to notify caregivers or emergency contacts.

The paper titled "Heart Rate Monitoring System using IR-based Sensor & Arduino Uno" [6] presents a practical solution for remote monitoring of patients, particularly focusing on home-bound individuals who require continuous health monitoring. By leveraging the capabilities of an IR-based heart rate sensor, Arduino Uno microcontroller, and GSM module, the system enables real-time monitoring of heart rates and transmission of data to healthcare providers via SMS. This approach addresses the challenge of providing timely assistance to patients in rural areas where internet connectivity may be limited. The methodology involves integrating the heart rate sensor output with the Arduino Uno board and programming it to calculate BPM based on the detected pulses. Communication between the Arduino Uno and GSM module is facilitated using AT commands, enabling the transmission of SMS alerts to designated mobile numbers. Additionally, an LCD display provides real-time feedback on heart rate measurements, ensuring immediate visualization even in the absence of GSM network connectivity.

CHAPTER 3

DETAILED DESIGN

3.1 SEQUENCE DIAGRAM

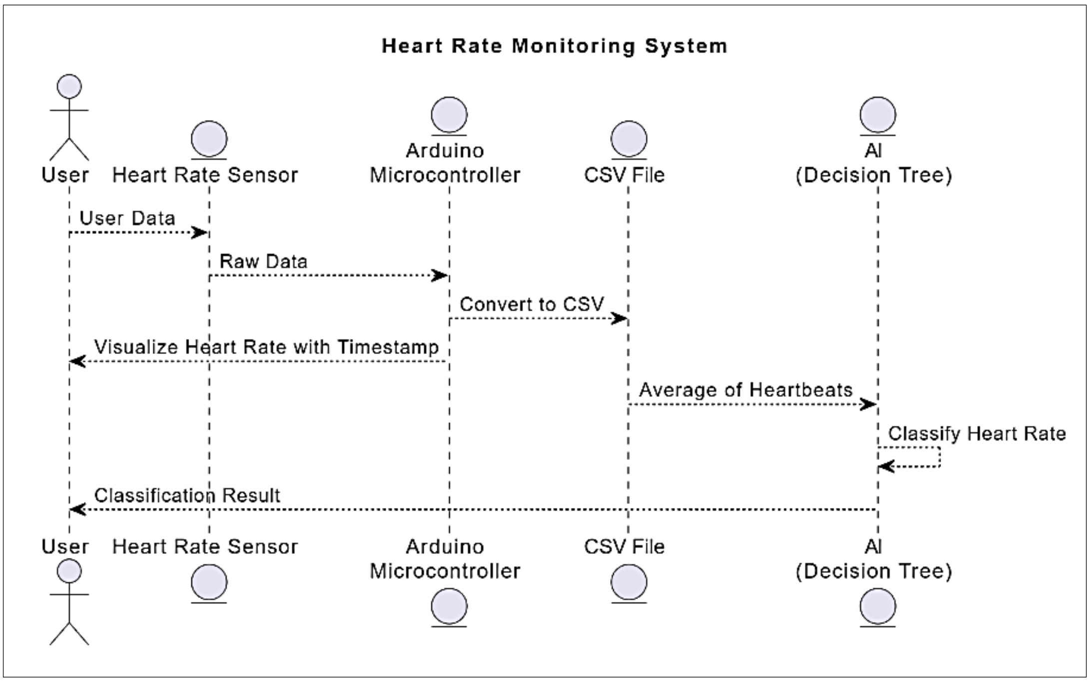


Fig 1. Sequence Diagram

For a heart rate monitoring system, the initial state or representation could be derived from various sources of data, each contributing to a comprehensive understanding of the user's heart rate dynamics.

1. **Actor Initialization:** The diagram starts with the actor User initiating the process by sensing the pulse rate. This implies that the user is using pulse sensor to measure their pulse rate.

2. **Data Collection:** The pulse rate sensor receives the sensed pulse rate data from the user and forwards it to the microcontroller.

3. **Data Processing:** The Arduino Microcontroller processes the received data. This step involves any necessary calculations required to prepare the data for display.

4. **Data Visualization:** After processing, the microcontroller sends the processed data to the display. This step represents the communication between the microcontroller and the display component in the system.

5. **Data Analysis:** The Arduino sends an average of 6 heartbeats (a subset of the CSV data) to an AI module, specifically a Decision Tree AI model.

6. **Decision Tree Analysis:** The AI module processes the averaged data to classify the heart rate. The classification process could involve determining if the heart rate is normal, elevated, or indicative of a health issue based on predefined thresholds.

7. **Classification Results:** Finally, the AI module sends the classification result back to the User. This outcome informs the user about their heart rate status, potentially prompting further action such as consulting a healthcare provider if irregularities are detected.

3.2 BLOCK DIAGRAM

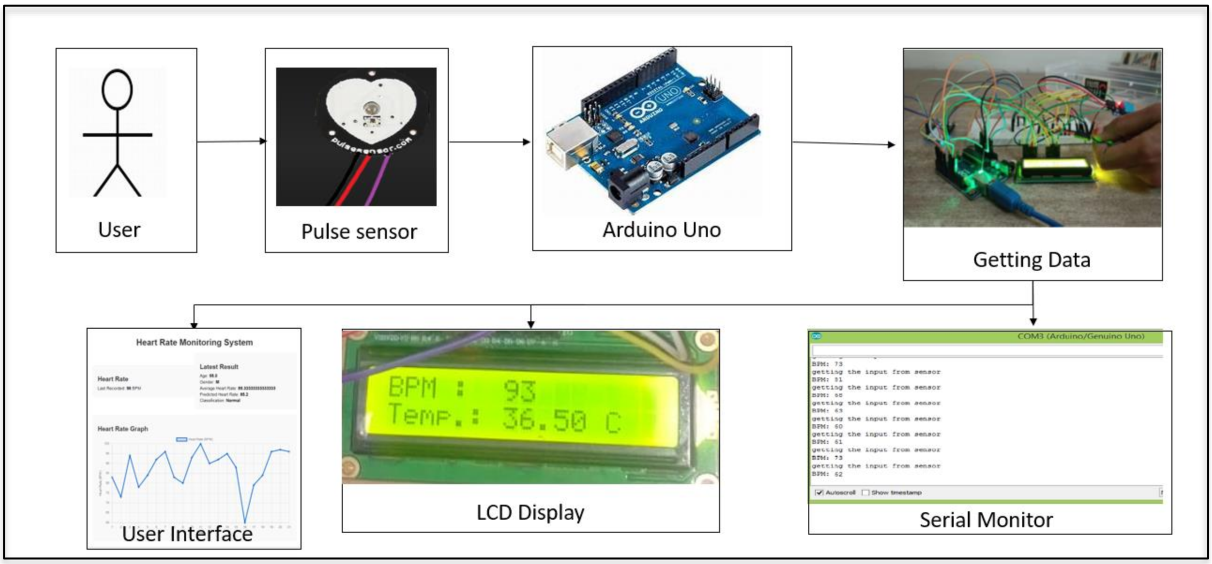


Fig 2. Block Diagram

The above Block Diagram represents the functionality of the Heart Rate Monitoring System. The user input is received via the pulse rate sensor, after which the data undergoes processing through the Arduino UNO microcontroller. Through Arduino and other circuit connections, the data is further processed. Subsequently, the processed data is displayed on the LCD screen, facilitating real-time feedback to the user. Additionally, the system visualizes the results through a user interface, enhancing accessibility and interpretability of the heart rate data.

3.3 USE CASE DIAGRAM

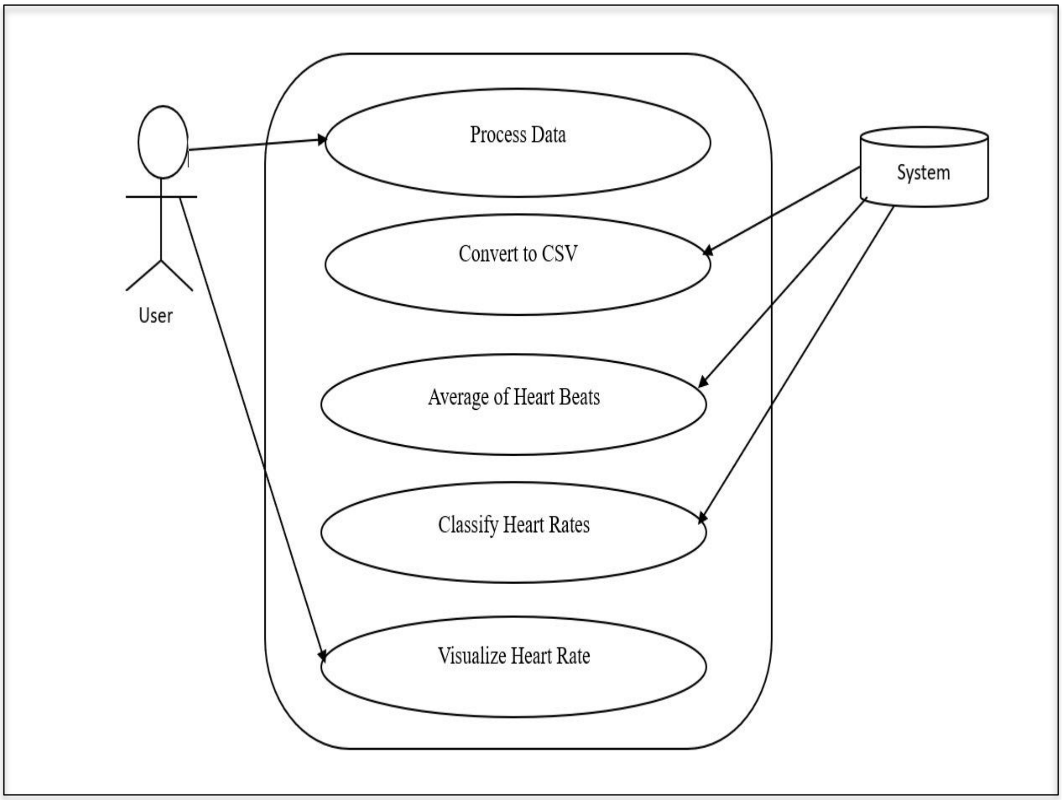


Fig 3. Use Case Diagram

The "Use Case Diagram" outlines the functional interactions within the Heart Rate Monitoring System project, depicting the roles of the User and the System components. Users engage with the system by placing their finger on the heart rate sensor, initiating the data processing task denoted by "Process Data." The processed data undergoes conversion to CSV format for analysis and visualization in the "Convert to CSV" and "Visualize Heart Rate" tasks, respectively. Subsequently, the system calculates the average heart rate in the "Average of Heartbeats" task and classifies heart rate patterns based on predefined criteria such as age and gender in the "Classify Heart Rates" task. Through these interactions, users receive classification results, facilitating informed decisions regarding their heart health.

3.4 ACTIVITY DIAGRAM

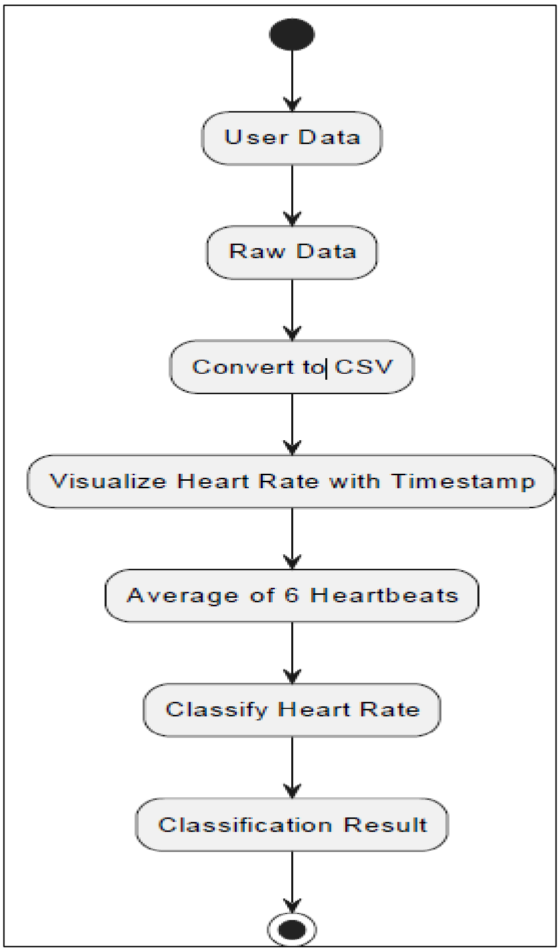


Fig 4. Activity Diagram

- User Data Collection: In this stage, user data, including heart rate measurements, is collected from various sources such as sensors.
- Conversion to CSV: The processed data is converted into a CSV format.
- Calculation of Average Heartbeats: The system computes the average heart rate based on a specified number of heartbeats.
- Heart Rate Classification: Using decision tree algorithm, the system classifies the user's heart rate data.
- Classification and Visualize Result: Finally, the system presents the classified heart rate information to the user. The result is visualized to provide a representation of heart rate trends over time.

CHAPTER 4

PROJECT SPECIFIC REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

- Processor: Intel Dual Core CPU and above
- Microcontroller: Arduino UNO at mega328p
- Hardware Components: Arduino UNO, LCD display, Bread Board, Jumper Wires, Pulse rate sensor, LED

4.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 7 and above
- Framework: Python, C++
- IDE: Arduino IDE, VS Code

4.3 COMPONENTS REQUIRED

Here, various components used for Heart Rate Monitoring System are explained.

4.3.1 PULSE RATE SENSOR

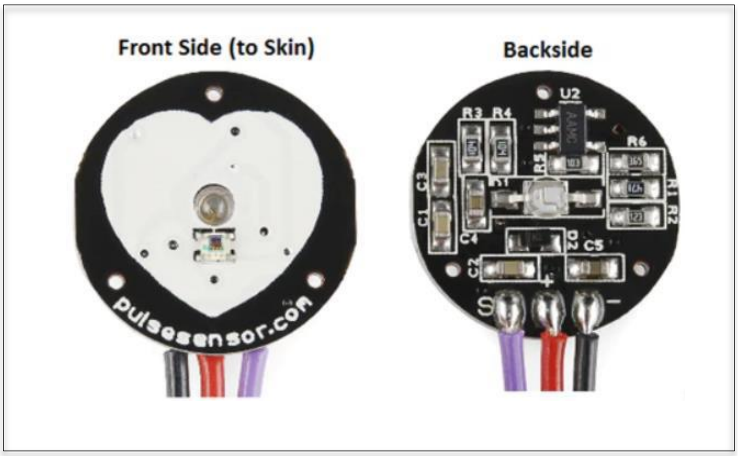


Fig 5. Pulse Rate Sensor

The Pulse Rate Sensor is a key component in health monitoring and wearable technology, designed to measure the heart rate of an individual. It typically employs an optical sensor to detect blood flow through capillaries, allowing for the non-invasive measurement of heart rate. Similar to the NodeMCU and Arduino Uno, the Pulse Rate Sensor is characterized by its user-friendly interface and integration capabilities. It often interfaces with microcontrollers like Arduino Uno through analog or digital pins, making it compatible with a wide range of projects. With real-time pulse rate data, it enables developers to create health-related applications, fitness trackers, and medical monitoring devices. The simplicity of integration, coupled with its accuracy and non-invasiveness, makes the Pulse Rate Sensor a valuable component for projects focused on health monitoring and biometric data acquisition.

4.3.2 ARDUINO UNO

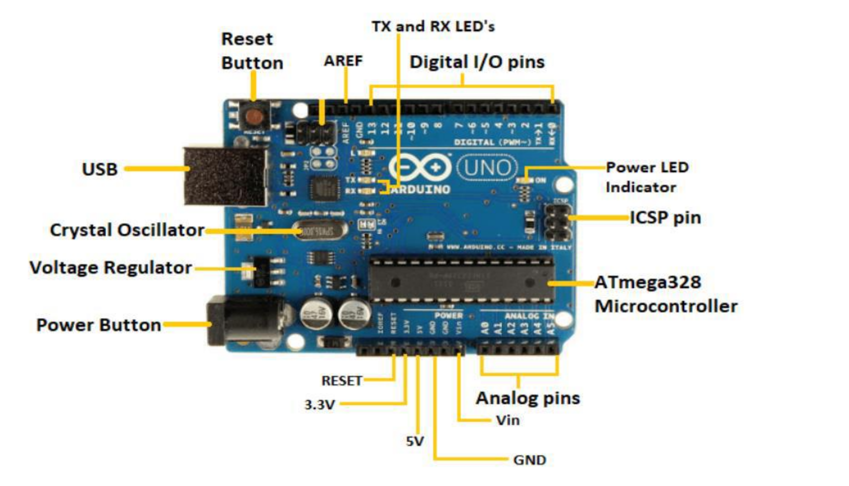


Fig 6. Arduino UNO

The Arduino Uno is a widely used open-source microcontroller board that is based on the ATmega328P microcontroller. It features 2 KB of SRAM and 32 KB of Flash memory for program storage. Its simplicity, affordability, and a large community of developers make it an excellent choice for various projects, particularly in the realm of embedded systems and prototyping. The Arduino Uno is equipped with 14 digital GPIO pins and 6 analog input pins, offering versatility in connecting to various sensors, actuators, and other electronic components. Its compatibility with the Arduino Integrated Development Environment (IDE) simplifies programming, allowing users to write and upload code easily. The Arduino Uno's straightforward design and extensive library support contribute to its accessibility, making it an ideal platform for both beginners and experienced makers to bring their electronic projects to life. Additionally, the Arduino Uno's compact form factor and low power consumption make it suitable for battery-powered applications, enabling portable and wireless projects. Its robust design and wide availability of shields and expansion boards further extend its capabilities, allowing users to enhance functionality and add features seamlessly.

4.3.3 NodeMCU

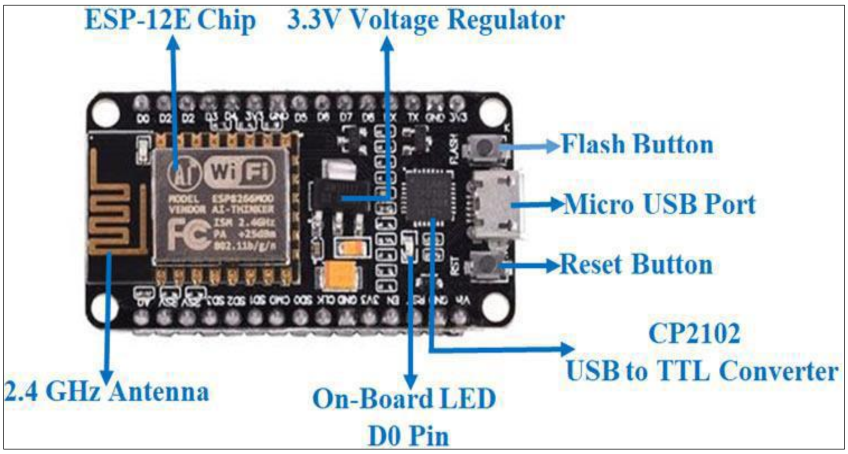


Fig 7. NodeMCU

The NodeMCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive system on a Chip called the ESP8266. It has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. It Contains 17 GPIOs (General Purpose Input Output). GPIO is a standard interface that connects the microcontroller to other electronic devices. For example, LEDs, Buzzers and other sensors are GPIOs. The NodeMCU ESP8266 stands out for its user-friendly design, allowing seamless development and prototyping in IoT projects. The NodeMCU supports a variety of interfaces, including I2C, SPI, and UART, expanding its connectivity options for diverse sensor integration. With a compact form factor and the ability to operate in deep sleep mode to conserve power, the NodeMCU is well-suited for battery-powered IoT applications.

CHAPTER 5

IMPLEMENTATION

5.1 CIRCUIT DIAGRAM

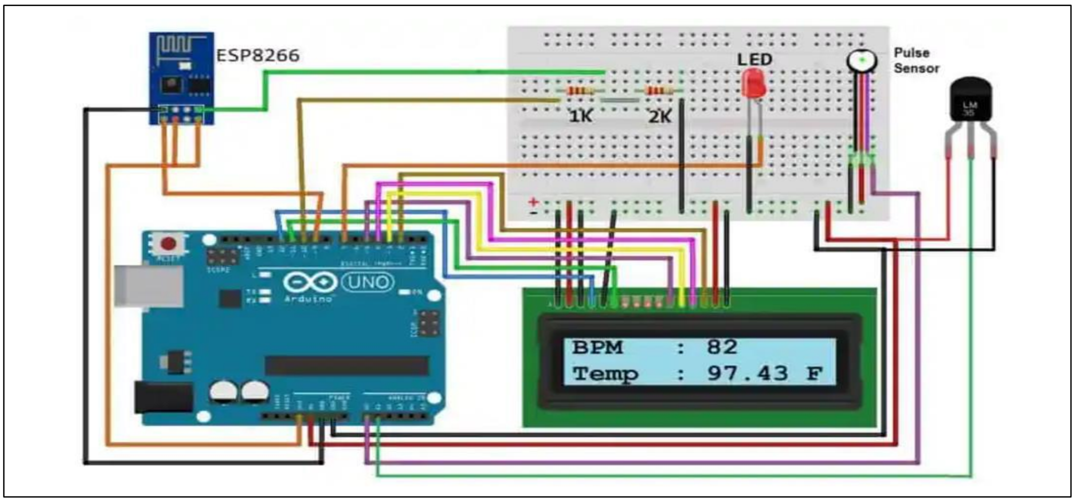


Fig 8. Circuit Diagram

1. Connect Pulse Sensor output pin to A0 of Arduino and other two pins to VCC & GND.
2. Connect LM35 Temperature Sensor output pin to A1 of Arduino and other two pins to VCC & GND.
3. Connect the LED to Digital Pin 7 of Arduino via a 220-ohm resistor.
4. Connect Pin 1,3,5,16 of LCD to GND.
5. Connect Pin 2,15 of LCD to VCC.
6. Connect Pin 4,6,11,12,13,14 of LCD to Digital Pin12,11,5,4,3,2 of Arduino.
7. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting the 2.2K & 1K resistor.
8. The RX pin of the ESP8266 is connected to pin 10 of Arduino through the resistors.
9. Connect the TX pin of the ESP8266 to pin 9 of the Arduino.

5.2 CREATION OF DATASET

People from different age groups have different ranges for maximum and minimum values of heart rate. The below figure represents Heart Rate Range of different age groups:

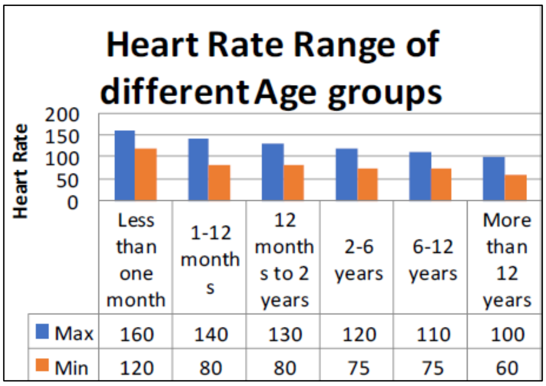


Fig. 9 Heart Rate Range

Our data set considers both the age group and gender of the users, allowing us to classify heart rate information accurately. Our dataset is enriched with labels for age and gender, alongside recordings of six different heartbeats. This collection includes data from both active and resting individuals. This table is a representative sample of the comprehensive data we have gathered to ensure our heart rate monitoring system is precise, reliable, and adaptable to the specific needs of various user groups.

The heart rate values for 5 different people from various age groups was obtained. Some of the dataset collected is as follows:

Age	Gender	Average Heart Sample	Activity
18	Female	78	Resting
28	Female	55	Running (Athlete)
60	Male	84	Resting
21	Male	54	Running (Athlete)
5	Male	102	Active

5.3 AI IMPLEMENTATION

To implement AI within the heart rate monitoring system, the process begins with gathering of dataset comprising heart rate data along with relevant user attributes. Subsequently, an appropriate machine learning model, such as decision tree algorithms, is selected and trained on the dataset to classify heart rates into categories like normal, low, or high. Integration with the Arduino and sensor involves establishing communication for real-time data transmission to the AI system, where the received CSV-formatted data is processed for classification. Upon classification, results are relayed back to the user interface, enabling informed decision-making.

5.3.1 DECISION TREE ALGORITHM

The decision tree algorithm is a popular method in machine learning that can be effectively applied to classify heart rate data. This algorithm creates a model that looks like a tree structure, with branches representing decisions based on data features, and leaves representing outcomes. For classifying heart rate, the decision tree will consider features such as age, gender, heart rate readings, and whether the person is active or resting. The process begins by choosing the best feature that provides the most significant information gain or reduces uncertainty the most. This feature forms the root of the tree. From there, the tree branches out based on the possible values of this feature, segmenting the dataset into smaller subsets. The process is repeated recursively on each subset using the remaining features. The construction continues until a certain stopping condition is met, such as when no further information gain is possible, or a predefined tree depth is reached.

CHAPTER 6

RESULTS

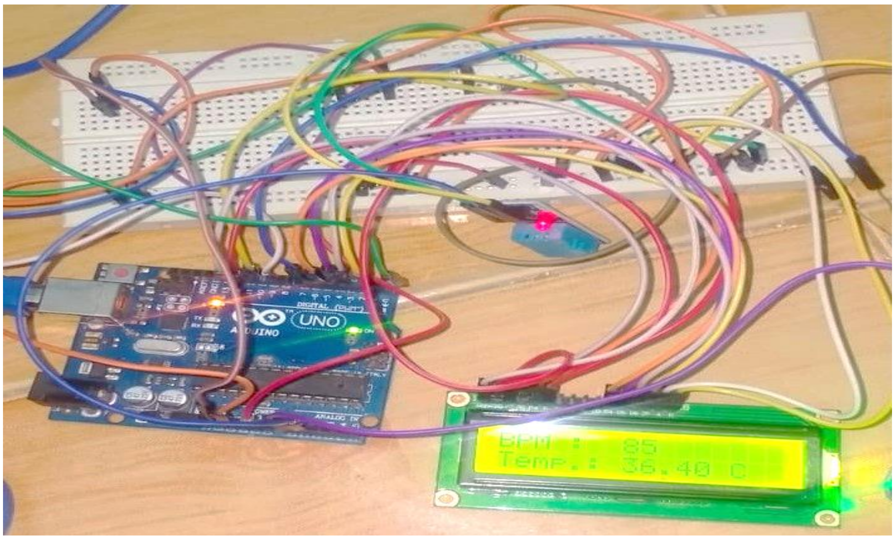
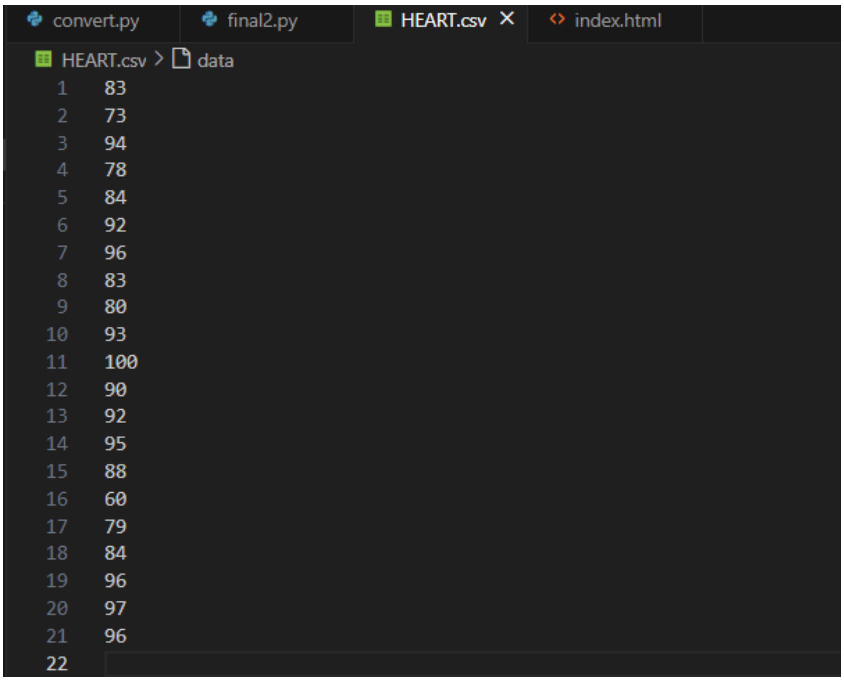


Fig 10. Circuit connections

In the above Fig 10, the heart rate monitoring system's circuit involve connecting the pulse sensor module to the microcontroller's analog input pin for real-time heart rate signal acquisition. Additionally, connecting the microcontroller to the display module enables immediate visualization of heart rate data, ensuring seamless monitoring and user interaction.

Fig 11. LCD Display

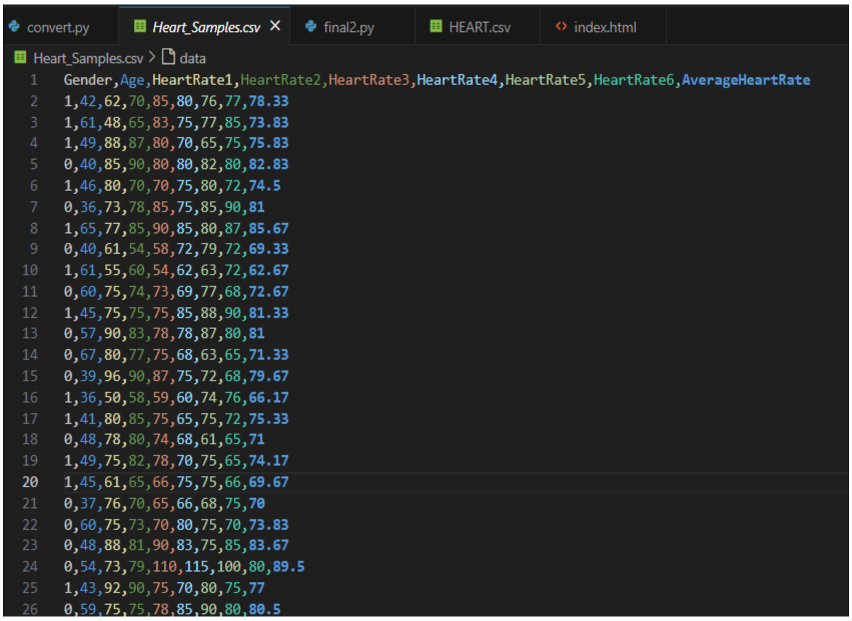
In the above Fig 11, display real-time heart rate and temperature readings on the LCD screen, providing users with immediate and accurate information at a glance.




```
83
73
94
78
84
92
96
83
80
93
100
90
92
95
88
60
79
84
96
97
96
```

Fig 12. Converted CSV file from sensed input

Fig 12 , depicts the converted CSV file generated from sensed input data. The figure provides a visual representation of the data captured from the sensor input including heart rate and temperature readings over time, in a format compatible with CSV files.




```
Gender,Age,HeartRate1,HeartRate2,HeartRate3,HeartRate4,HeartRate5,HeartRate6,AverageHeartRate
1,42,62,70,85,80,76,77,78.33
1,61,48,65,83,75,77,85,73.83
1,49,88,87,80,70,65,75,75.83
0,40,85,90,80,80,82,80,82.83
1,46,80,70,70,75,80,72,74.5
0,36,73,78,85,75,85,90,81
1,65,77,85,90,85,80,87,85.67
0,40,61,54,58,72,79,72,69.33
1,61,55,60,54,62,63,72,62.67
0,60,75,74,73,69,77,68,72.67
1,45,75,75,75,85,88,90,81.33
0,57,90,83,78,78,87,80,81
0,67,80,77,75,68,63,65,71.33
0,39,96,90,87,75,72,68,79.67
1,36,50,58,59,60,74,76,66.17
1,41,80,85,75,65,75,72,75.33
0,48,78,80,74,68,61,65,71
1,49,75,82,78,70,75,65,74.17
1,45,61,65,66,75,75,66,69.67
0,37,76,70,65,66,68,75,70
0,60,75,73,70,80,75,70,73.83
0,48,88,81,90,83,75,85,83.67
0,54,73,79,110,115,100,80,89.5
1,43,92,90,75,70,80,75,77
0,59,75,75,78,85,90,80,80.5
```

Fig 13. Collected Dataset

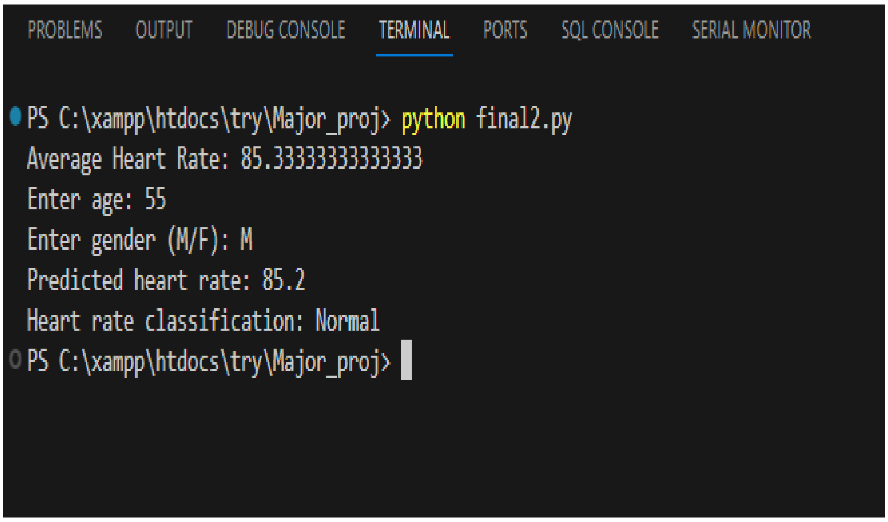


Fig 14. Classification Result

The classification result description utilizes age, gender, and heart rate data to categorize and interpret the sensed input.

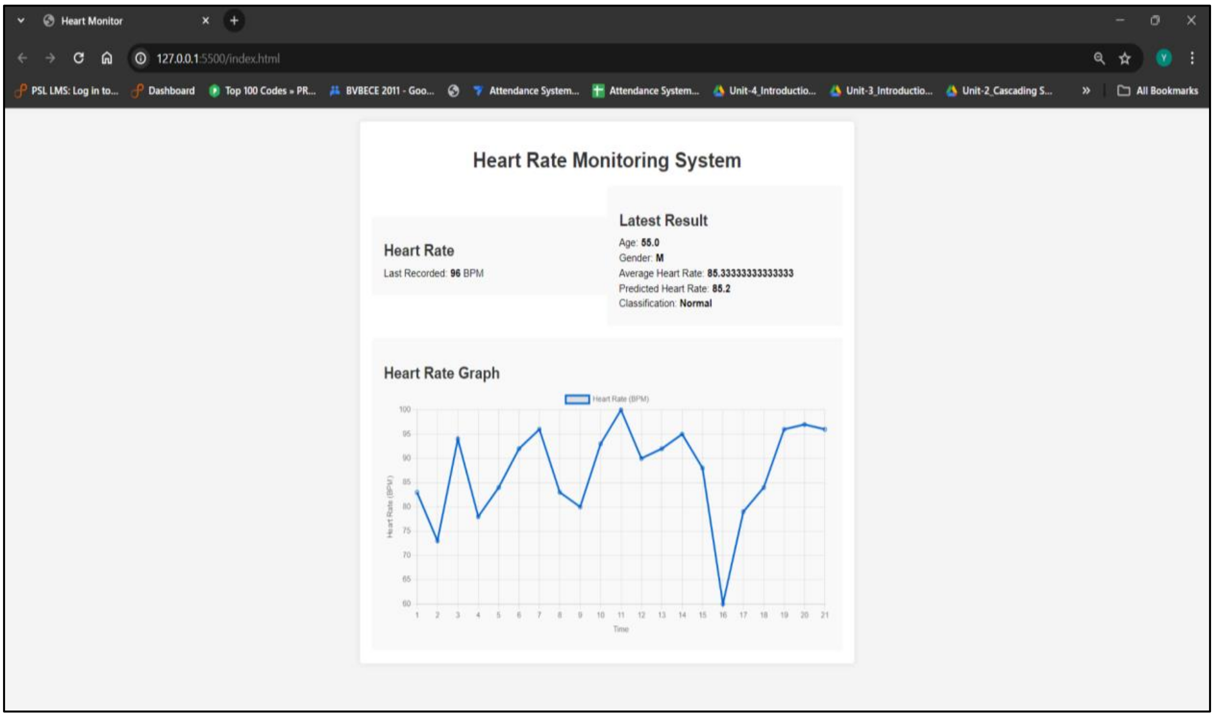


Fig 15. User Interface

The user interface provides a user-friendly platform for interacting with the heart rate monitoring system. It features clear and visually appealing displays of real-time heart rate data, temperature readings, and any detected irregularities.

CHAPTER 7

CONCLUSION

The heart rate monitoring system aims to revolutionize healthcare monitoring by providing continuous, real-time heart rate tracking with the detection of irregularities. By capturing heart rate sample data, it ensures accurate monitoring over time. Utilizing the Decision Tree algorithm technique, the system efficiently classifies irregularities based on age and gender and also provides effective, user-friendly visualization along with graphical representation and real-time predicted data of the user.

7.1 FUTURE SCOPE

Our model employs minimal number of sensors, integrating additional sensors to capture comprehensive health data, including blood pressure and activity level, enhances its capabilities. Advancements in AI can predict cardiac events, enabling proactive interventions. Adapting it for remote monitoring offers personalized care beyond clinics. Cloud-based storage and analytics can enable population-level health monitoring, advancing cardiovascular management on a broader scale.

REFERENCES

[1] Aritra Ray, Hena Ray,“Secured Pulse Rate Monitoring System using IoT and Cloud” Cornell University Library, UTC from IEEE Xplore.

[2] TV Sethuraman, Kartik Singh Rathore, Amritha G, Kanimozhi G "IoT based system for Heart Rate Monitoring and Heart Attack Detection," International Journal of Engineering and Advanced Technology (IJEAT), 2019, ISSN: 2249-8958.

[3] S.Gopi, Dr.E.Punarselvam, “Heart Attack Recognition And Heart Rate Monitoring System Using Iot”, International Journal on Applications in Engineering and Technology, 2019, ISSN : 2455 – 0523.

[4] Sahana S Khamitkar, Prof. Mohammed Rafi “IoT based System for Heart Rate Monitoring”, International Journal of Engineering Research & Technology (IJERT) 2020, ISSN: 2278-018.

[5] Puneet Bansal, Meenal Malik, Ricktam Kundu, “Smart Heart Rate Monitoring System” 2018 IEEE.

[6] S Gopi, Dr. E. Punarselvam,“Heart Attack Recognition And Heart Rate Monitoring System Using IoT”, International Journal on Applications in Engineering and Technology, ISSN: 2455 – 0523.

[7] Saikat Mukherjee, Arpita Ghosh, Subir Kumar Sarkar, “Arduino based Wireless Heart-rate Monitoring System”, International Conference On Vision Towards Emerging Trends In Communication And Networking – 2019.