International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Healthy Me (Health Monitoring System using IOT and ML)

Irusumalla Venkata Surya Kalanjali [a], Muvvala Sandhya [b], Bunga Sona Jasmine [c], Barukula Rithima Sanjana [d], Dr. M. Anil Kumar [e]

[a] Department of Computer Science and Engineering, Aditya College Of Engineering and Technology, Surampalem, 533437, India, anjaliirusumalla@gmail.com

[b] Department of Computer Science and Engineering, Aditya College Of Engineering and Technology, Surampalem, 533437, India, sandhyamuvvala513@gmail.com

[c] Department of Computer Science and Engineering, Aditya College Of Engineering and Technology, Surampalem, 533437, India, sonajasmine03@gmail.com

[d] Department of Computer Science and Engineering, Aditya College Of Engineering and Technology, Surampalem, 533437, India, rithimasanjana@gmail.com

[e] Guide, M. Tech., Ph.D Professor & HoD, Department of Computer Science and Engineering, Aditya College of engineering and technology , Andhra Pradesh , India, mail: lettertoanil@gmail.com

ABSTRACT:

The Internet of Things connects billions of people and things in real world environment. Among most researches, current research is mostly on health monitoring system. This system integrates IoT sensors placed on medical equipment and patients to continuously gather vital signs, such as heart rate, blood pressure, oxygen saturation and temperature of the human being. The collected data are transmitted wirelessly to a centralized platform where advanced algorithms process and analyze the information. Through this IoT-enabled monitoring system, healthcare professionals gain immediate access to comprehensive patient data, enabling timely interventions and personalized treatment plans. Additionally, the system facilitates remote monitoring capabilities, allowing medical staff to oversee multiple patients simultaneously and respond promptly to any emergent situations. Overall, the Health monitoring system powered by IoT promises to enhance patient outcomes, streamline clinical workflows, and improve resource utilization in critical care settings.

This paper introduces the concept of using RFID and IOT based technology in Health Monitoring system. It provides a solution using RFID concept connected with Internet of Things (IoT).

Intensive Care Unit or HEALTH is where the patients who are critically ill are admitted for treatment. For such critical conditions the Doctors need to have an all-time update patient's health related parameters like their blood pressure, heart pulse and temperature. To do manually, this is too tedious atask and also for multiple patients it becomes close to impossible. For this type of situations this IOT based system can bring about an automation that can keep the Doctors updated all time over internet. IOT Based Health Patient Monitoring System is a Raspberry Pi based system which collects patient's information with the help of few sensors. It uses Wifi module to communicat this information to the internet. There is this Blood pressure and heart beat of which

monitor module electrically connected to the system and physically to be worn by theser. On the press of button, the sensor senses the blood pressure in systolic and diastolic along with the heart beat and sends it to the central controller. The Temperature sensor senses the temperature of its ambience, so when this sensor is in close proximity of the user it reports the users' body temperature. Thus, the doctor can get access to these vital parameters pertaining to the patients' health over the IOT Gecko web interface from anywhere over the world. In this way IOT Based Health Patient Monitoring System is an enhanced system that helps in monitoring HEALTH Patients without any manual intervention.system effectively uses internet to monitor patient health stats and can save lives on time.

Automations in industrial, commercial or residential sectors mostly depend upon the power systems, which require distant controlling and monitoring. With the proliferation of wireless technologies, it is more efficient to implement an appropriate technology depending upon the cost, speed and distance requirements of the proposed system.

In this proposed system we have designed a Hospital Room monitoring system using IoT Technology. This system is interfaced with multiple sensor to Monitor the Hospital Room data and Upload it in Cloud Server. Air Quality Gas Sensor, Light and Temperature sensor is used to monitor the Patient room data and process it using Arduino Nano controller. This data is displayed in LCD module for local monitoring. Node mcu is used for WiFi connectivity and further upload this data on Google Sheets. User can view this data from anywhere in the World using Internet technology. Embedded C Language is used to write the program logic and further it is uploaded on Arduino Board using Onboard usb programmer.

Keywords : IOT, ML, Healthcare Industry.

Introduction :

What is IOT?

The term Internet of Things (often abbreviated IoT) was coined more than ten years ago by industry researchers but has emerged into mainstream public view only more recently. Some claim the Internet of Things will completely transform how computer networks are used for the next 10 or 100 years, while others believe IoT is simply hype that won't much impact the daily lives of most people.

Internet of Things represents a general concept for the ability of network devices to sense and collect data from the world around us, and then share that data across the Internet where it can be processed and utilized for various interesting purposes.

Some also use the term industrial Internet interchangeably with IoT. This refers primarily to commercial applications of IoT technology in the world of manufacturing. The Internet of Things is not limited to industrial applications, however.

What the Internet of Things Can Do for Us

Some future consumer applications envisioned for IoT sound like science fiction, but some of the more practical and realistic sounding possibilities for the technology include:

- receiving warnings on your phone or wearable device when IoT networks detect some physical danger is detected nearby
- self-parking automobiles
- automatic ordering of groceries and other home supplies
- automatic tracking of exercise habits and other day-to-day personal activity including goal tracking and regular progress reports

Potential benefits of IoT in the business world include:

- location tracking for individual pieces of manufacturing inventory
- fuel savings from intelligent environmental modeling of gas-powered engines
- new and improved safety controls for people working in hazardous environments

Network Devices and the Internet of Things

All kinds of ordinary household gadgets can be modified to work in an IoT system. Wi-Fi network adapters, motion sensors, cameras, microphones and other instrumentation can be embedded in these devices to enable them for work in the Internet of Things. Home automation systems already implement primitive versions of this concept for things like light bulbs, plus other devices like wireless scales and wireless blood pressure monitors that each represent early examples of IoT gadgets. Wearable computing devices like watches and glasses are also envisioned to be key components in future IoT systems.

The same wireless communication protocols like Wi-Fi and Bluetooth naturally extend to the Internet of Things also.

In recent years, the growth of internet is tremendous and has been further extended to connecting things through internet. IOT devices are used to collect, monitor, evaluate and notify the patient with the information. From the development of technologies (Internet of Things) is changing the human life into a new level. IOT is change the normal human life to smart life with new technology level. There are several process such as smart home, smart city, health monitoring systems are monitor using Internet of Things. Internet of Things is used for monitor all patients in any level. In this paper, patient's heart rate, patient's heart rate, body temperature, blood pressure reading those are like systolic and diastolic and humidity and body movements using raspberry pi.

The Health monitoring system using IOT is done with by using the raspberry pi which is connected tot the blood pressure module sensor, temperature and humidity sensor. It is an AVR based 8 bit microcontroller. it has 1KB electrically erasable programmable read only memory(EEPROM).Its excellent features include the cost efficiency, low power dissipation programming lock for security purposes, real timer counter.

1. Overview of Health Monitoring in IOT

Patient Health Monitoring:

- Continuous tracking of vital signs (blood pressure, heart rate, body temperature) using wearable sensors.
- Monitoring environmental factors (air quality, light, room temperature) with room-mounted sensors.

Wireless Connectivity and Data Transmission:

- Utilizing WiFi connectivity for seamless data transfer to a central monitoring platform accessible over the internet.
- Enabling remote access to patient data for medical staff from anywhere in the world.

Data Processing and Display:

- Processing sensor data locally using microcontrollers (Raspberry Pi, Arduino Nano).
- Displaying real-time data on LCD modules for onsite monitoring.
- Uploading data to cloud servers (Google Sheets) for storage and further analysis.

Power Management:

- Incorporating a stable power supply setup for continuous operation.

Software Implementation:

- Programming microcontrollers using Python and Embedded C.
- Integration with IoT platforms for seamless data transmission and remote access.

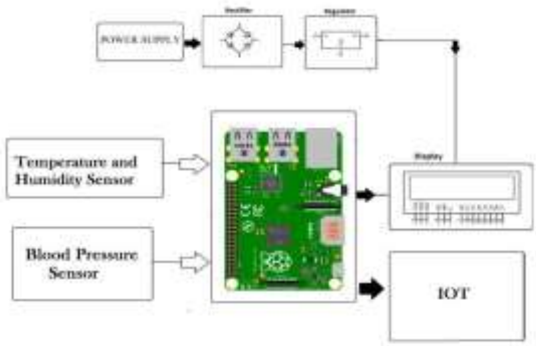
Potential Impact:

- Improving patient care by automating monitoring processes and enabling timely interventions.

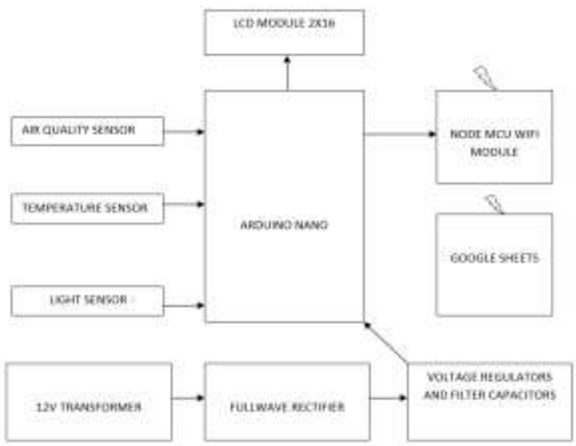
Room Monitoring:-

A room monitoring system using IoT technology integrates various sensors to create a smart environment that ensures comfort, safety, and energy efficiency. This system employs temperature sensors, such as DHT22, to monitor the room's temperature and maintain optimal climate conditions. Air quality sensors, like the MQ135, detect pollutants and volatile organic compounds, ensuring the air remains healthy and breathable. Light sensors, such as LDRs or TSL2561, adjust artificial lighting based on natural light levels, optimizing energy use and enhancing comfort. Gas sensors, like the MQ2, detect harmful gases such as methane and propane, alerting occupants to potential hazards. These sensors are connected to a microcontroller, such as an ESP32 or Raspberry Pi, which collects and processes the data, transmitting it to a cloud platform for real-time monitoring and analysis. Users can access this data via a web dashboard or mobile app, receiving alerts and visualizations to track room conditions and make informed decisions. This integration of IoT devices not only enhances the quality of the indoor environment but also contributes to energy savings and proactive safety measures.

2. About the Block Diagram Of Health Monitoring Using IOT



PATIENT MONITORING BLOCK DIAGRAM



ROOM MONITORING BLOCK DIAGRAM

3.Existing System:-

The current health monitoring system harnessing IoT technology is a multifaceted network that revolutionizes the way individuals manage and track their well-being. Through a combination of sensors, wearable devices, and medical equipment, this system collects real-time data on various physiological metrics such as heart rate, blood pressure, respiratory rate, and oxygen saturation levels. These data are then transmitted wirelessly to a centralized platform where healthcare professionals can access and analyze them in real-time.

By leveraging IoT capabilities, this system enables continuous monitoring and remote patient management, allowing medical staff to promptly detect any abnormalities or changes in patient's conditions and intervene as necessary. Additionally, the integration of IoT technology facilitates seamless communication between different medical devices and systems, optimizing workflow efficiency and ensuring accurate and timely data capture.

Bedside Monitors: Traditional bedside monitors were used to continuously monitor vital signs and other physiological parameters of ICU patients. These monitors were typically hardwired to the patient and displayed real-time data on a screen visible to healthcare providers.

Medical Sensors and Equipment: Various medical sensors and equipment, such as electrocardiography (ECG) machines, pulse oximeters, blood pressure cuffs, ventilators, and infusion pumps, were employed to monitor and manage patients' health in the ICU. These devices were often standalone and not connected to a centralized system

Centralized Monitoring Stations: In addition to bedside monitors, centralized monitoring stations were set up in the ICU to display aggregated data from multiple patients. Healthcare providers could view vital signs and alarms for each patient from these stations, allowing for continuous surveillance of the ICU environment.

Manual Documentation: Healthcare professionals manually recorded patient data, including vital signs, medications, interventions, and observations, in paper-based charts or electronic health records **(EHR)** systems. This documentation process required frequent manual input and was susceptible to errors and delays.

Alarm Systems: Bedside monitors and medical devices were equipped with alarm systems to alert healthcare providers of abnormal vital signs, equipment malfunctions, or other critical events requiring immediate attention. However, false alarms and alarm fatigue were common challenges associated with these systems.

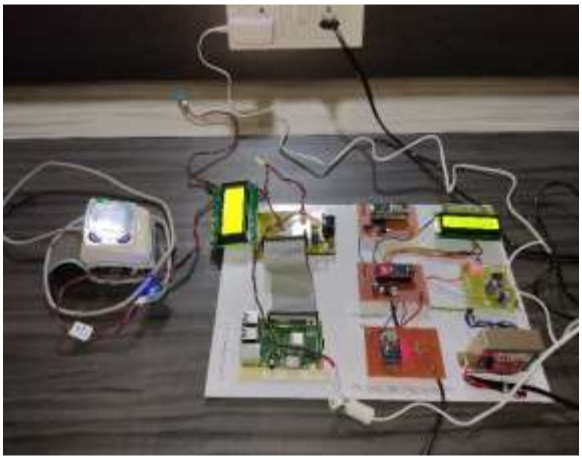
Limited Connectivity: The lack of connectivity between medical devices and centralized systems limited the ability to aggregate and analyze patient data comprehensively in real-time. This hindered the early detection of deteriorating patient conditions and timely intervention by healthcare providers.

Data Integration Challenges: Integrating data from different types of medical devices and equipment into a unified system posed significant technical challenges. Compatibility issues, proprietary data formats, and interoperability barriers hindered seamless data exchange and analysis.

Manual Surveillance: Healthcare professionals relied heavily on manual surveillance and observation to monitor patients' conditions, assess responses to treatment, and identify signs of deterioration. This required frequent rounds by nursing staff and limited the efficiency of patient monitoring.

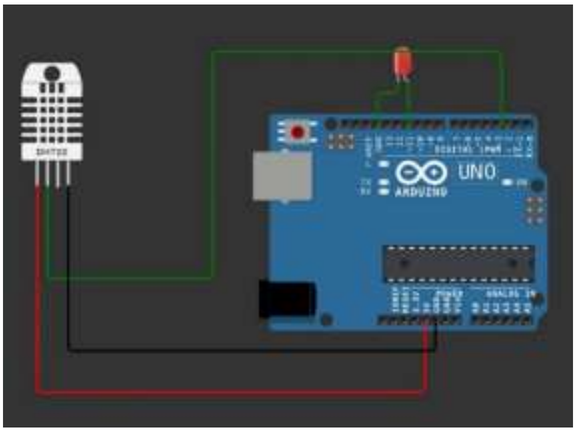
4.Extension of the existing project:-

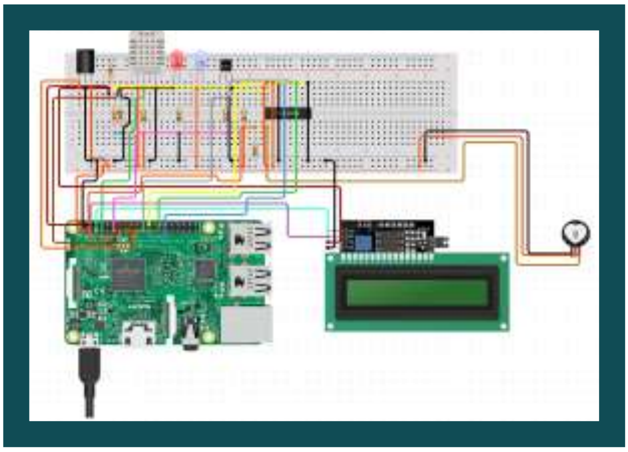
The Extension of the health monitoring using iot is by adding the room monitoring system using iot as below.



UI DESIGN FOR THE HEALTH MONITORING SYSTEM USING IOT

In the context of IoT (Internet of Things), UI (User Interface) design refers to the creation of interactive and user-friendly interfaces that allow users to interact with and control IoT devices and systems. This involves designing graphical and textual interfaces that display data collected by IoT devices and provide controls to manage these devices.





Data Visualization: Presenting real-time data from sensors and devices in an intuitive manner, often using graphs, charts, and dashboards. This helps users understand complex data at a glance.

User Controls: Providing interactive elements like buttons, sliders, and switches that enable users to control IoT devices remotely. This could include turning devices on/off, adjusting settings, or configuring alerts.

Responsiveness: Ensuring that the interface works seamlessly across various devices, including smartphones, tablets, and desktops, to provide a consistent user experience regardless of the platform.

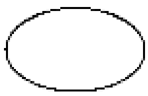
Usecase Diagram

Use case diagram is used to capture high level functionalities of a system. It establishes the relationship between the use cases and the actors. Name and additional components can be used for clarification. It represents a set of actions performed by a system for a specific goal. Actors are used in use case diagrams to define entities.

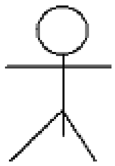
Use case diagram is used to capture high level functionalities of a system.

Name and additional components can be used for clarification.

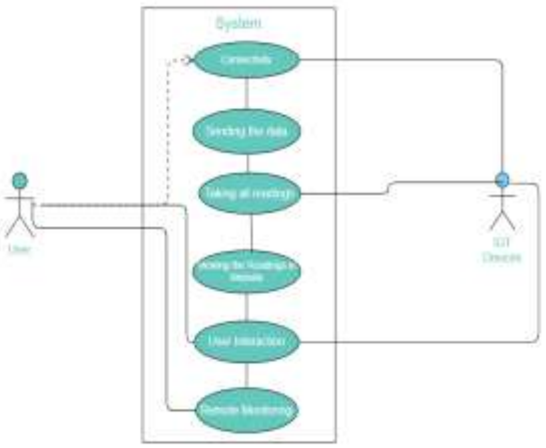
Use case

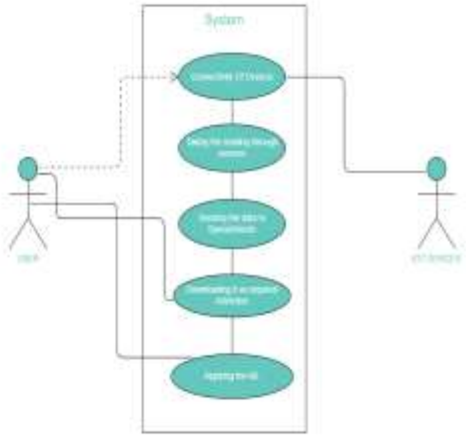


Actor is used in a use case diagram to describe entities.



Actor



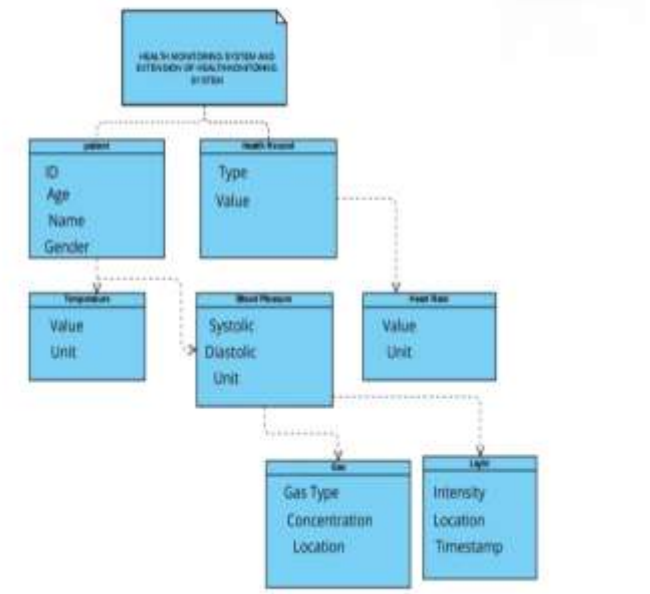


Class Diagram:-

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object- oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.



CLASS DIAGRAM FOR HEALTH MONITORING SYSTEM

Sequence Diagram:-

It is also called as an interaction diagram .It emphasizes the time ordering of messages.

UML sequence diagram are used to show how objects interact in a given situation.

Use of sequence diagram is to document the dynamics in an object-oriented system.

A sequence diagram in the context of IoT (Internet of Things) illustrates the interactions between different components of an IoT system over time.

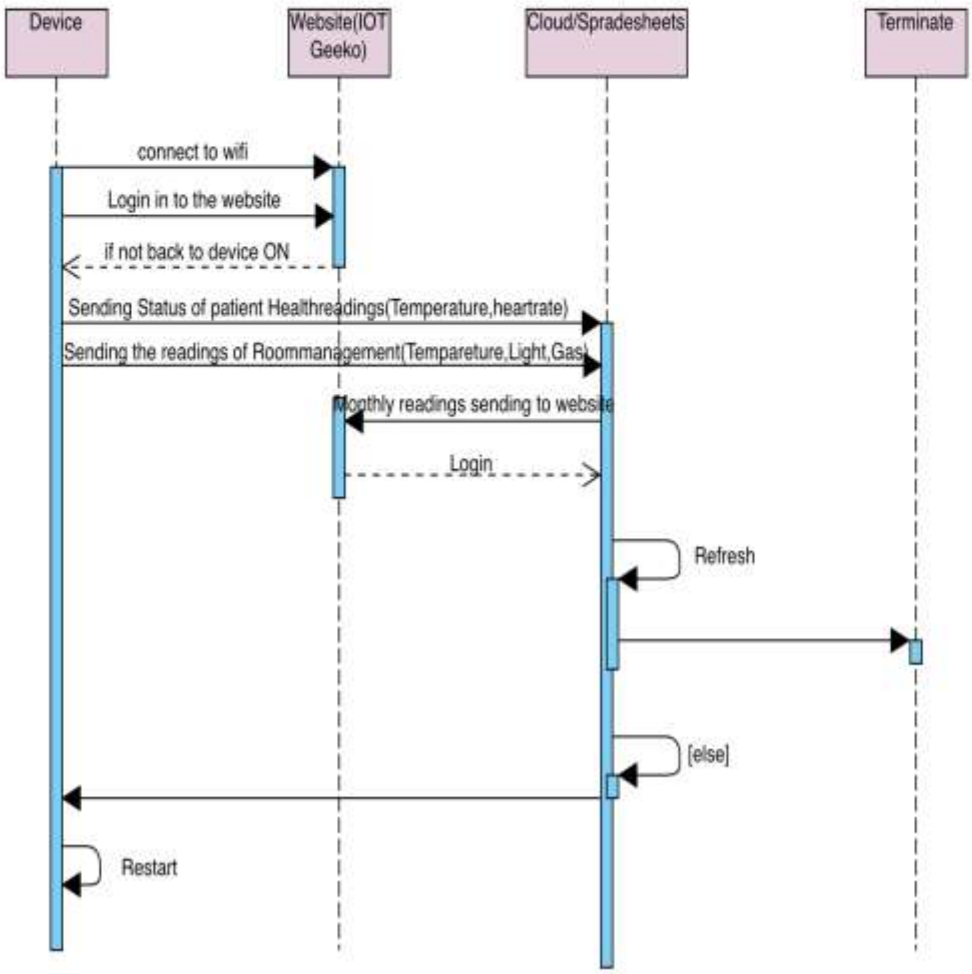
This type of diagram is part of the Unified Modeling Language (UML) and helps visualize how various entities, such as sensors, devices, gateways, cloud services, and users, communicate with each other to perform specific tasks.

User: The person monitoring and controlling the room environment.

Mobile App/Web Interface: The interface through which the user interacts with the system.

Microcontroller (e.g., ESP32): The central processing unit that collects data from sensors.

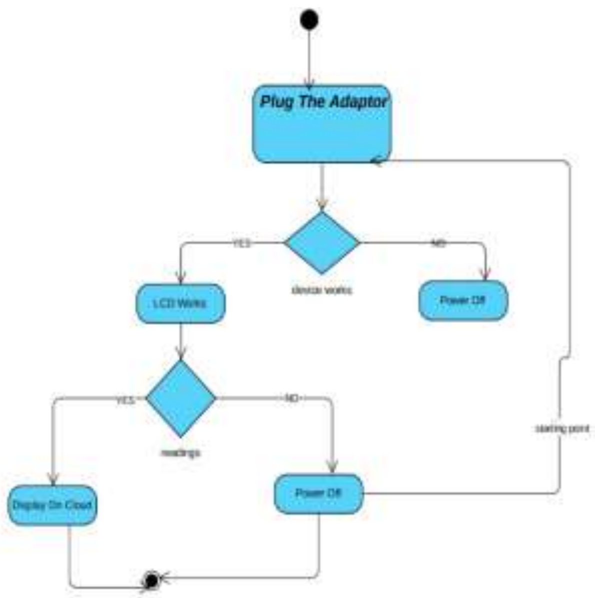
Sensors: Devices that measure temperature, air quality, light levels, and gas presence.



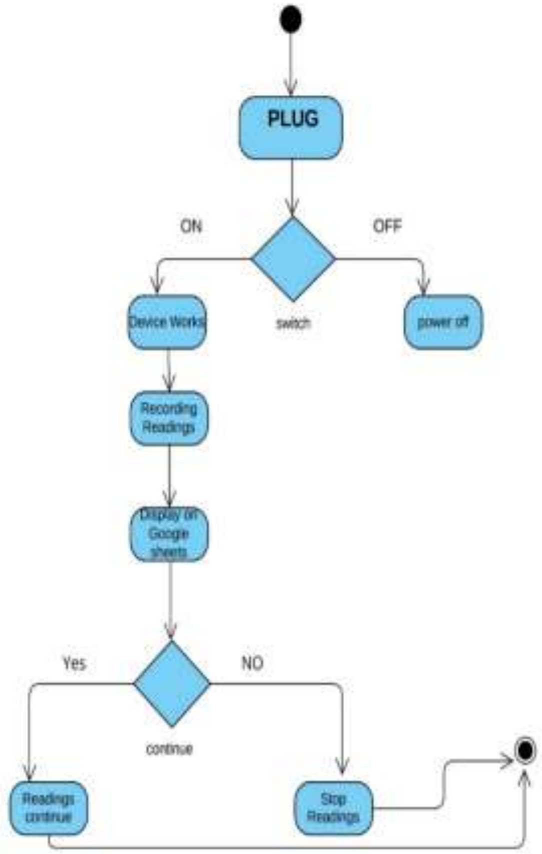
SEQUENCE DIAGRAM FOR HEALTH MONITORING SYSTEM

Activity Diagram:-

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. Activity is a particular operation of the system.



ACTIVITY DIAGRAM OF PATIENT MONITORING



ACTIVITY DIAGRAM OF ROOM MONITORING

7.1 TESTING LEVELS

SOFTWARE TESTING:

Functional Testing:

1. Data Collection and Processing:

Sensor Data Collection:

- Test Case 1: Sensor Connectivity
    - Description: Verify that the Python scripts establish successful communication with each connected sensor.
    - Preconditions: Sensors connected to Raspberry Pi, Python scripts running.
    - Steps:
        1. Power on the sensors.
        2. Execute the data collection script.
    - Expected Result: Sensors are detected and communication is established without errors.
    - Pass/Fail Criteria: All sensors are detected and communication is established without errors.
- Test Case 2: Sensor Data Accuracy
    - Description: Ensure that sensor data is accurately captured and stored by the system.
    - Preconditions: Sensors connected to Raspberry Pi, Python scripts running.
    - Steps:
        1. Apply known input values to each sensor.
        2. Run the data collection script.
    - Expected Result: Sensor data matches the known input values within an acceptable tolerance.
    - Pass/Fail Criteria: Sensor data matches the known input values within an acceptable tolerance.

Data Processing Accuracy:

- Test Case 3: Algorithm Accuracy
    - Description: Evaluate the accuracy of the algorithms used for processing sensor data.
    - Preconditions: Sensor data collected and stored.
    - Steps:
        1. Input known sensor data.
        2. Run the data processing algorithms.
    - Expected Result: Processed data matches the expected values within a specified tolerance.
    - Pass/Fail Criteria: Processed data matches the expected values within a specified tolerance.

Efficiency and Error Handling:

- Test Case 4: Efficiency Testing
    - Description: Assess the efficiency of the data processing algorithms under varying complexities of input data.
    - Preconditions: Data processing scripts running.
    - Steps:
        1. Inject test data with varying complexities.
        2. Monitor processing time and resource usage.
    - Expected Result: Data processed efficiently without significant delays or resource overutilization.

- Pass/Fail Criteria: Data processing completed within acceptable time limits without causing system slowdowns or failures.

- Test Case 5: Error Handling
  - Description: Evaluate the system's ability to handle errors gracefully during data processing.
  - Preconditions: Data processing scripts running.
  - Steps:
    1. Introduce simulated errors or anomalies in input data.
    2. Monitor system behaviour and error handling mechanisms.
  - Expected Result: Errors are detected and handled without causing system crashes or data corruption.
  - Pass/Fail Criteria: Errors are detected and handled appropriately without compromising system integrity.

2. Communication Testing:

Communication Establishment:

- Test Case 1: Connection Establishment
  - Description: Verify successful establishment of communication between Raspberry Pi and IoT Gecko platform.
  - Preconditions: Raspberry Pi connected to Wi-Fi network, IoT Gecko platform accessible.
  - Steps:
    1. Initiate communication from Raspberry Pi.
    2. Verify connection status on IoT Gecko platform.
  - Expected Result: Successful connection established between Raspberry Pi and IoT Gecko.
  - Pass/Fail Criteria: Connection established without errors.

Data Transmission Reliability and Security:

- Test Case 2: Data Transmission
  - Description: Test reliability and security of data transmission over Wi-Fi.
  - Preconditions: Connection established between Raspberry Pi and IoT Gecko.
  - Steps:
    1. Transmit test data from Raspberry Pi to IoT Gecko.
    2. Verify data reception on IoT Gecko platform.
  - Expected Result: Data transmitted reliably and securely without loss or compromise.
  - Pass/Fail Criteria: Data transmitted and received without errors, encryption and authentication protocols verified.

3.Dashboard Testing:

1. Usability and Functionality Evaluation:

- Test Case 1: User Interface Design
  - Description: Evaluate the design of the IoT Gecko dashboard for monitoring patient data.
  - Preconditions: IoT Gecko dashboard accessible.
  - Steps:
    1. Navigate through different sections of the dashboard.
    2. Assess the layout, colour scheme, and visual elements.
  - Expected Result: Intuitive and aesthetically pleasing user interface design.
  - Pass/Fail Criteria: User interface design meets usability standards and enhances user experience.
- Test Case 2: Navigation and Accessibility

- Description: Assess the navigation and accessibility features of the IoT Gecko dashboard.
- Preconditions: IoT Gecko dashboard accessible.
- Steps:
    1. Test navigation between different pages and sections.
    2. Evaluate accessibility options such as font size adjustments and screen reader compatibility.
- Expected Result: Easy navigation and accessibility for users with varying needs.
- Pass/Fail Criteria: Navigation is intuitive, and accessibility features enhance usability for all users.

4. Responsiveness and Performance Testing:

- Test Case 3: Dashboard Update Speed
    - Description: Measure the speed at which the IoT Gecko dashboard updates with real-time patient data.
    - Preconditions: IoT Gecko dashboard accessible, real-time data sources available.
    - Steps:
        1. Monitor the dashboard for updates in patient data.
        2. Introduce changes in real-time data sources and observe dashboard response.
    - Expected Result: Quick and seamless updates of patient data on the dashboard.
    - Pass/Fail Criteria: Dashboard updates promptly and accurately in response to changes in real-time data sources.
- Test Case 4: Performance under Load
    - Description: Evaluate the performance of the IoT Gecko dashboard under different loads and usage scenarios.
    - Preconditions: IoT Gecko dashboard accessible, simulated user interactions or data influx available.
    - Steps:
        1. Simulate concurrent user interactions with the dashboard.
        2. Increase the load by introducing a higher volume of data influx.
    - Expected Result: Dashboard maintains responsiveness and usability under varying loads.
    - Pass/Fail Criteria: Dashboard remains responsive and usable without significant slowdowns or disruptions under increased load.

5..Integration Testing:

1. Hardware-Software Integration:

- Test Case 1: Sensor-Script Interaction
    - Description: Ensure that Python scripts interact correctly with connected sensors.
    - Steps:
        1. Activate Python scripts responsible for sensor data collection.
        2. Verify that sensor data is accurately received and processed by the scripts.
    - Expected Result: Python scripts successfully collect and process sensor data without errors.
    - Pass/Fail Criteria: Sensor data is correctly interpreted by the scripts, and no communication errors occur.
- Test Case 2: API Functionality
    - Description: Verify the functionality of APIs for data transmission between the Raspberry Pi and IoT Gecko platform.
    - Steps:
        1. Send test data from Raspberry Pi to IoT Gecko using APIs.
        2. Confirm successful data transmission and reception on the IoT Gecko platform.
    - Expected Result: Test data is transmitted securely and accurately between the Raspberry Pi and IoT Gecko.

- Pass/Fail Criteria: Data transmission occurs without loss or compromise, and authentication mechanisms are verified.

6..Security Testing:

1. Data Transmission Security:

- Test Case 3: Encryption Protocols
  - Description: Verify the implementation of encryption protocols for data transmission over Wi-Fi.
  - Steps:
    1. Monitor data transmission between Raspberry Pi and IoT Gecko.
    2. Confirm the use of encryption algorithms to secure transmitted data.
  - Expected Result: Data transmitted over Wi-Fi is encrypted to prevent unauthorised access.
  - Pass/Fail Criteria: Encryption protocols are effectively implemented, and sensitive patient data remains protected during transmission.

HARDWARE TESTING:

1. Sensor Testing:

Test Case 1: Blood Pressure and Heart Rate Sensor Functionality

- Description: Verify that the blood pressure and heart rate sensor functions correctly.
- Preconditions: Sensor connected to Raspberry Pi.
- Steps:
- a. Apply known blood pressure and heart rate values to the sensor.
- b. Read sensor output using Raspberry Pi.
- Expected Result: Sensor output matches the input values within an acceptable tolerance.
- Pass/Fail Criteria: Sensor output matches the input values within acceptable tolerance.

Test Case 2: Temperature and Humidity Sensor Accuracy

- Description: Test the accuracy of the temperature and humidity sensor.
- Preconditions: Sensor connected to Raspberry Pi.

Steps:

a. Place the sensor in a controlled environment with known temperature and humidity.

- b. Read sensor output using Raspberry Pi.
- Expected Result: Sensor output matches the known temperature and humidity values within an acceptable tolerance.
- Pass/Fail Criteria: Sensor output matches the known values within acceptable tolerance.

2. Raspberry Pi Testing:

Test Case 3: Boot-Up and Hardware Interface

- Description: Verify that the Raspberry Pi boots up correctly and interfaces with connected hardware components.
- Preconditions: Raspberry Pi connected to hardware components.
- Steps:
- a. Power on the Raspberry Pi.
- b. Verify that all connected hardware components are detected.
- Expected Result: Raspberry Pi boots up successfully and interfaces with connected hardware components without errors.
- Pass/Fail Criteria: Raspberry Pi boots up successfully and detects all connected hardware components without errors.

3. Wi-Fi Modem Testing:

Test Case 4: Wi-Fi Connectivity

- Description: Verify the Wi-Fi connectivity of the modem and its ability to connect to the internet.
- Preconditions: Wi-Fi Modem connected to Raspberry Pi and configured.
- Steps:
- a. Power on the Wi-Fi Modem.
- b. Connect Raspberry Pi to Wi-Fi network.
- c. Verify internet connectivity.
- Expected Result: Raspberry Pi successfully connects to the Wi-Fi network and has internet access.
- Pass/Fail Criteria: Raspberry Pi connects to the Wi-Fi network and has stable internet access.

4. LCD Display Testing:

Test Case 5: Display Functionality

- Description: Verify the functionality of the LCD display for presenting information.
- Preconditions: LCD Display connected to Raspberry Pi and powered on.
- Steps:
- a. Display test information on the LCD screen.
- b. Verify readability and clarity of displayed information.
- Expected Result: Information is displayed clearly and legibly on the LCD screen.
- Pass/Fail Criteria: Information is displayed accurately and is easily readable on the LCD screen.

5. LEDs Testing:

Test Case 6: LED Functionality

- Description: Verify the functionality of LEDs for indicating system status or alerts.
- Preconditions: LEDs connected to Raspberry Pi and powered on.
- Steps:
- a. Trigger different system states or alerts.
- b. Verify that LEDs light up or blink accordingly.
- Expected Result: LEDs illuminate or blink as expected based on system states or alerts.
- Pass/Fail Criteria: LEDs respond correctly to system states or alerts without malfunction.

6. Power Supply Testing:

Test Case 7: Stability and Reliability

- Description: Test the stability and reliability of the power supply to ensure consistent operation of all hardware components.
- Preconditions: Power supply connected to Raspberry Pi and other hardware components.
- Steps:
- a. Monitor power supply voltage and current under normal operation.
- b. Introduce variations in load and monitor power supply stability.
- Expected Result: Power supply delivers consistent voltage and current under varying loads.
- Pass/Fail Criteria: Power supply maintains stability and reliability, ensuring consistent operation of all hardware components.

Hardware Setup Verification

- Description: Ensure that all necessary hardware components are properly connected and functioning.

- Preconditions: Hardware components assembled according to system diagram.

Steps:

a. Verify physical connections of all hardware components.

b. Power on the system and check for any physical damages or loose connections.

c. Verify LED indicators on Arduino Nano and Node MCU Wifi Module are lit.

- d. Check responsiveness of sensors.
- Expected Result: All hardware components are properly connected without any physical damages or loose connections. LED indicators are lit, indicating power supply. Sensors provide expected readings.
- Pass/Fail Criteria: Pass: All components are properly connected and functional. Fail: Any hardware component is damaged, connections are loose, or sensors are unresponsive.

Stability and Reliability

- Description: Test the stability and reliability of the power supply to ensure consistent operation of all hardware components.
- Preconditions: Power supply connected to Raspberry Pi and other hardware components.

Steps:

a. Monitor power supply voltage and current under normal operation.

- b. Introduce variations in load and monitor power supply stability.
- Expected Result: Power supply delivers consistent voltage and current under varying loads.
- Pass/Fail Criteria: Pass: Power supply maintains stability and reliability, ensuring consistent operation of all hardware components. Fail: Power supply exhibits instability or inconsistency in voltage and current delivery, leading to erratic behaviour or failure of hardware components.

Software Testing:

Test Case 5: Code Logic Verification

- Description: Ensure that the embedded C programming logic is correctly implemented on the Arduino Nano.
- Preconditions: Embedded C code written for the Arduino Nano controller.

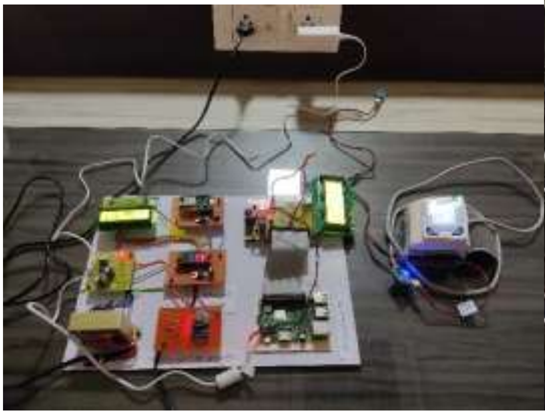
Steps:

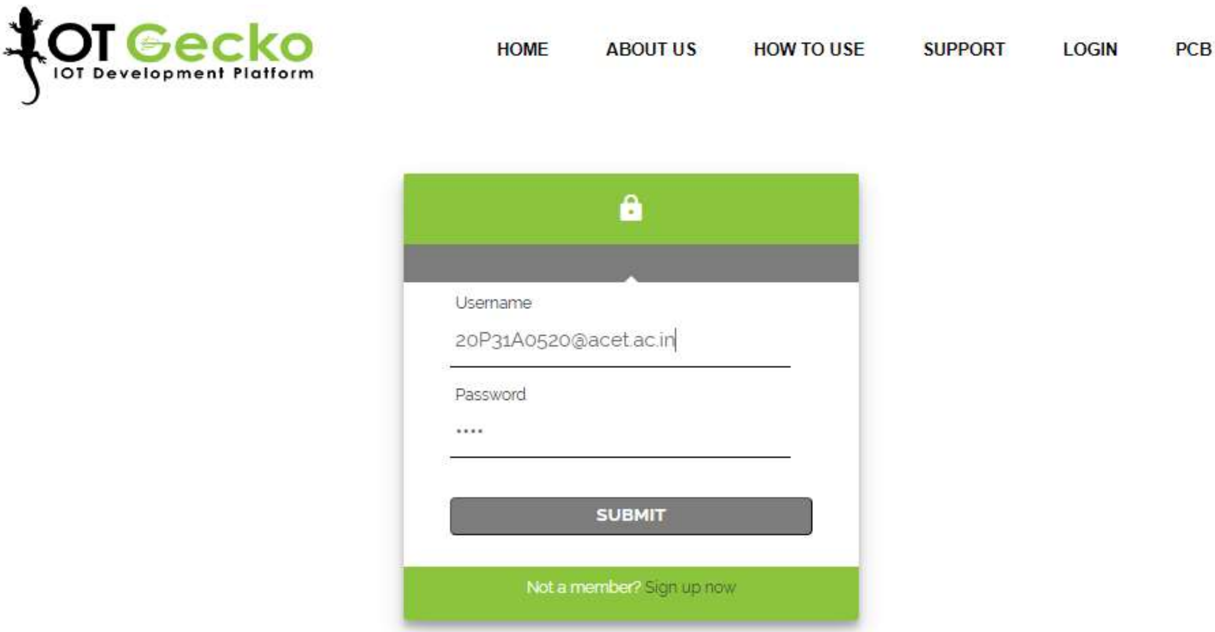
a. Review the embedded C code.

b. Compile the code using the Arduino IDE compiler.

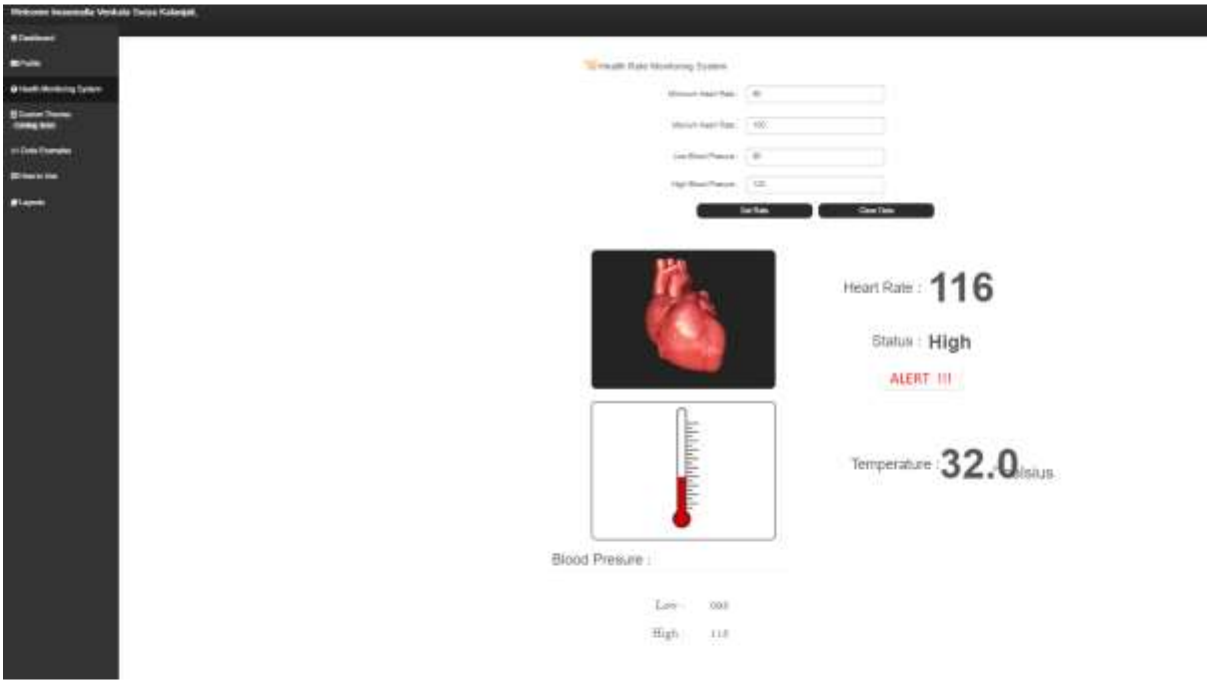
c. Upload the code to the Arduino Nano board.

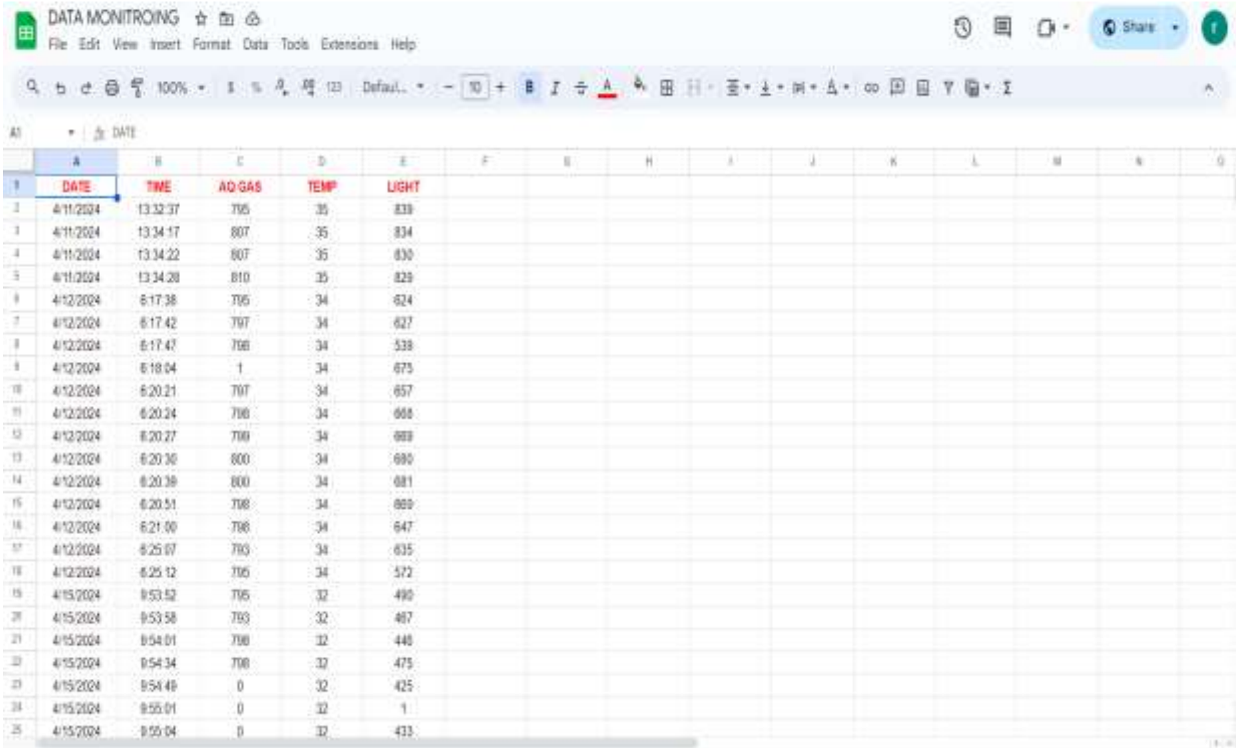
- d. Monitor the serial output for any error messages or unexpected behaviour.
- Expected Result: Embedded C code logic is correctly implemented. Compilation and upload are successful without any errors.
- Pass/Fail Criteria: Pass: Code logic is correct, and upload is successful. Fail: Code contains logic errors, fails to compile/upload, or exhibits unexpected behaviour during testing.



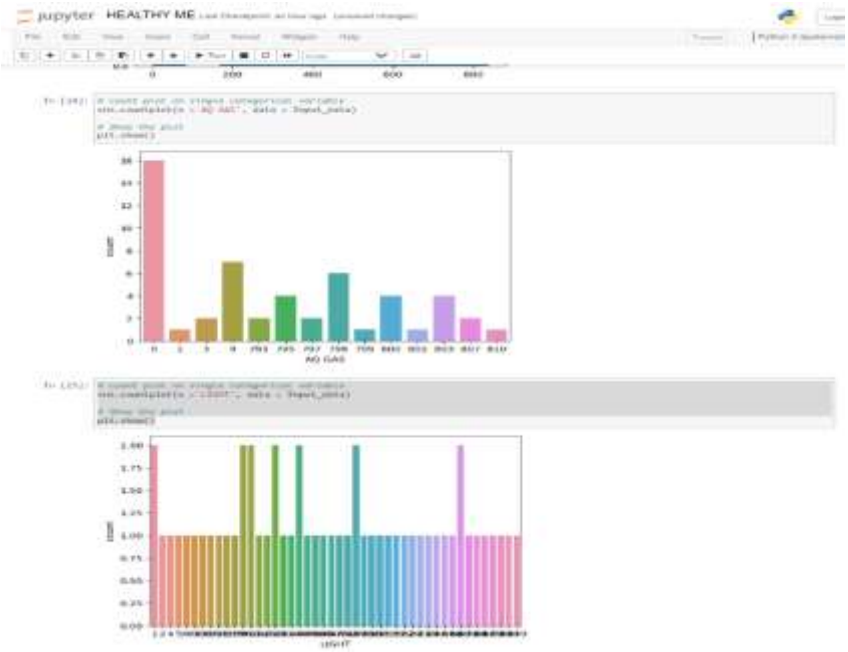

IOT Gecko
IOT Development Platform
HOME
ABOUT US
HOW TO USE
SUPPORT
LOGIN
PCB
Username
20P31A0520@acet.ac.in
Password
SUBMIT
Not a member? Sign up now

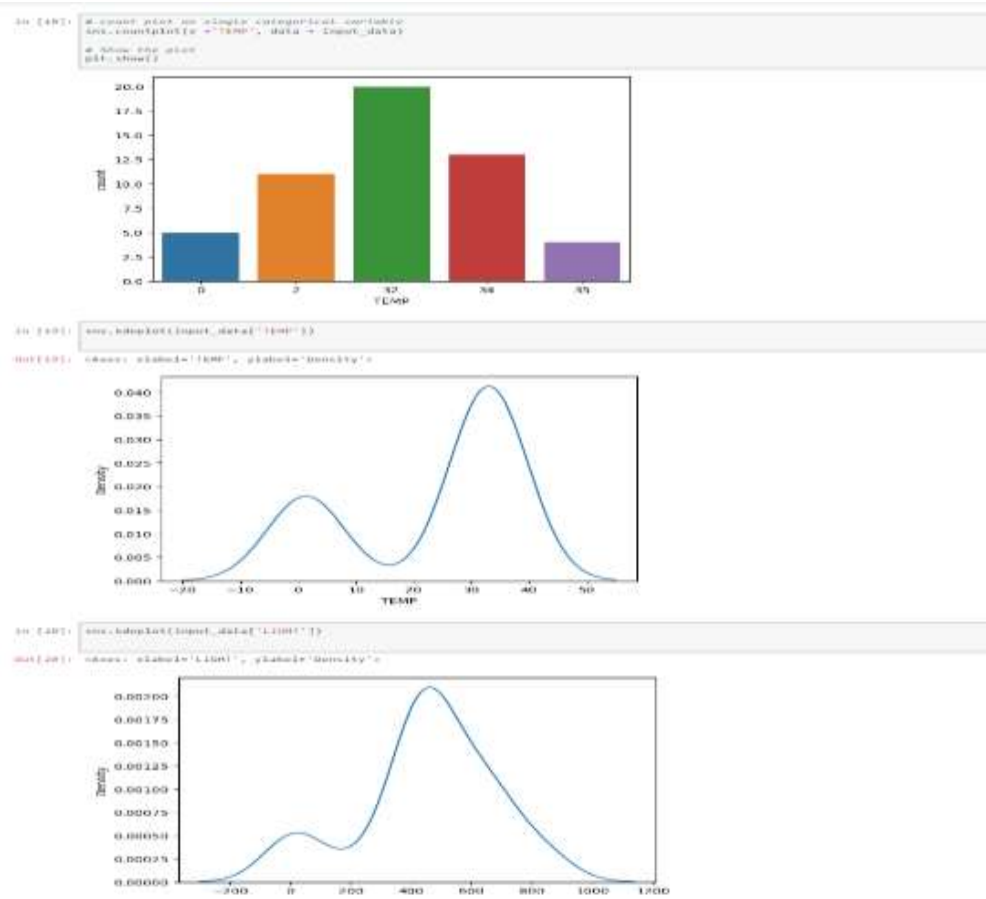

Dashboard
0
23
65

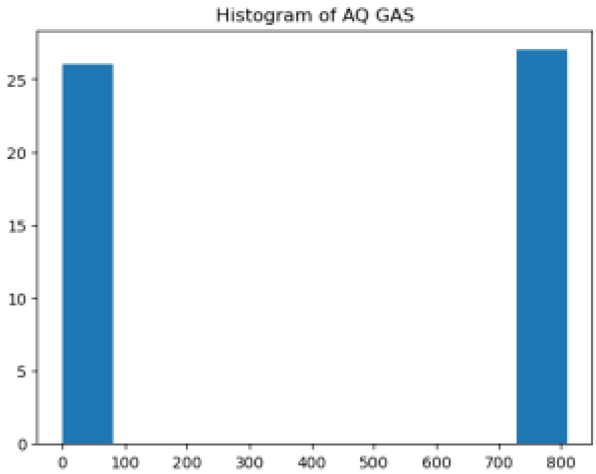


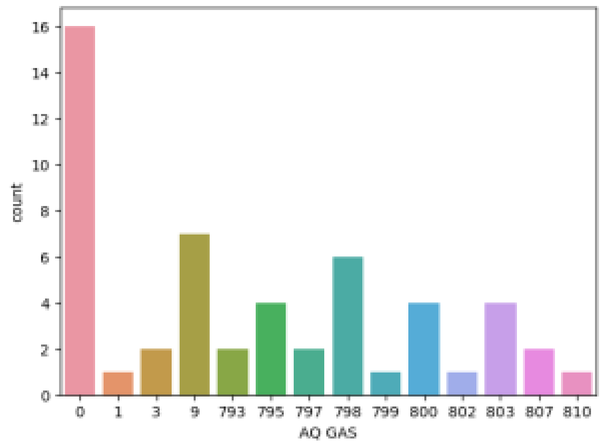


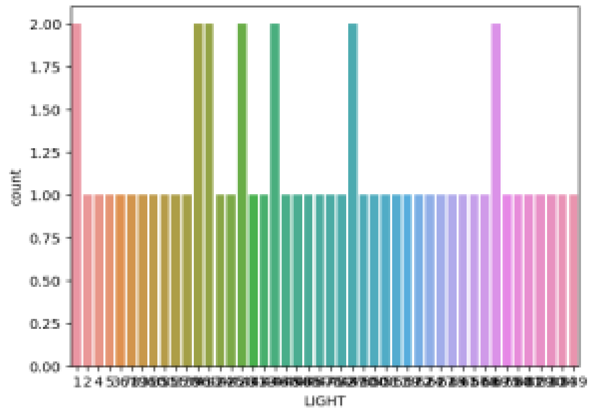
DATE	TIME	AQ GAS	TEMP	LIGHT
4/11/2024	13:32:37	795	35	839
4/11/2024	13:34:17	807	35	834
4/11/2024	13:34:22	807	35	830
4/11/2024	13:34:28	810	35	829
4/12/2024	6:17:38	795	34	624
4/12/2024	6:17:42	797	34	627
4/12/2024	6:17:47	798	34	538
4/12/2024	6:18:04	1	34	675
4/12/2024	6:20:21	797	34	657
4/12/2024	6:20:24	798	34	668
4/12/2024	6:20:27	799	34	669
4/12/2024	6:20:30	800	34	680
4/12/2024	6:20:39	800	34	681
4/12/2024	6:20:51	798	34	669
4/12/2024	6:21:00	798	34	647
4/12/2024	6:25:07	793	34	635
4/12/2024	6:25:12	795	34	572
4/15/2024	9:53:52	795	32	490
4/15/2024	9:53:58	793	32	467
4/15/2024	9:54:01	798	32	448
4/15/2024	9:54:34	798	32	475
4/15/2024	9:54:49	0	32	425
4/15/2024	9:55:01	0	32	1
4/15/2024	9:55:04	0	32	433

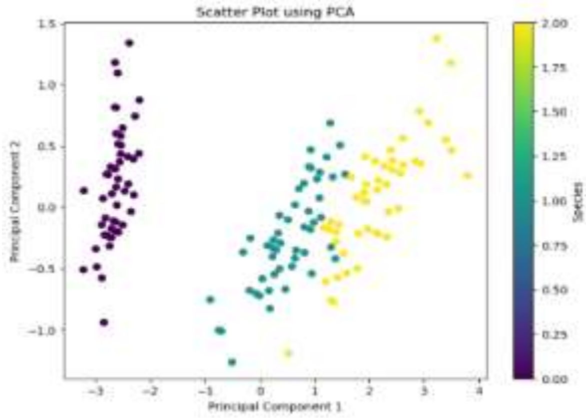

jupyter HEALTHY ME
count
AQ GAS
count

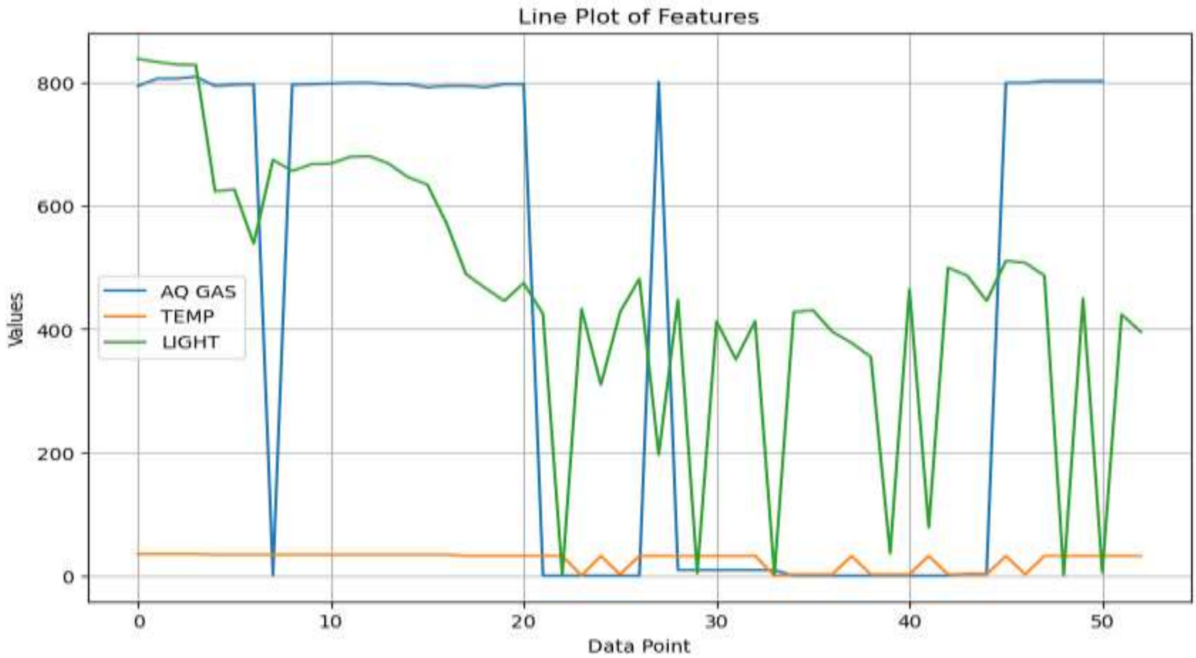

count
TEMP
Density
TEMP
Density

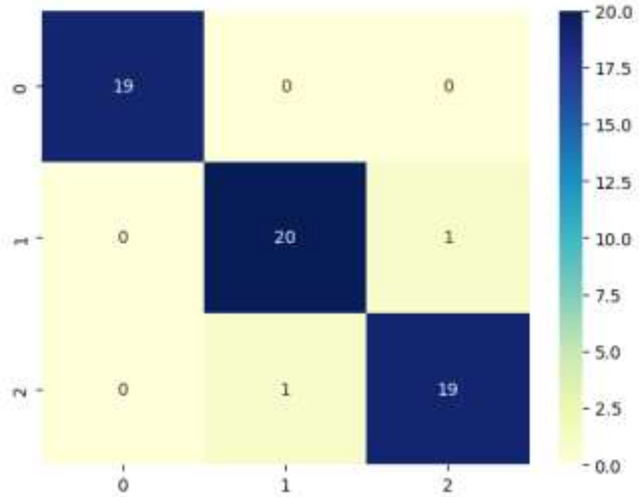

Histogram of AQ GAS
25
20
15
10
5
0
0 100 200 300 400 500 600 700 800

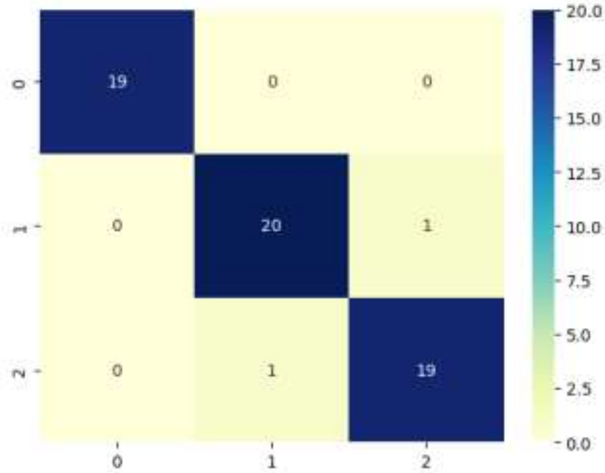

count
16
14
12
10
8
6
4
2
0
0 1 3 9 793 795 797 798 799 800 802 803 807 810
AQ GAS

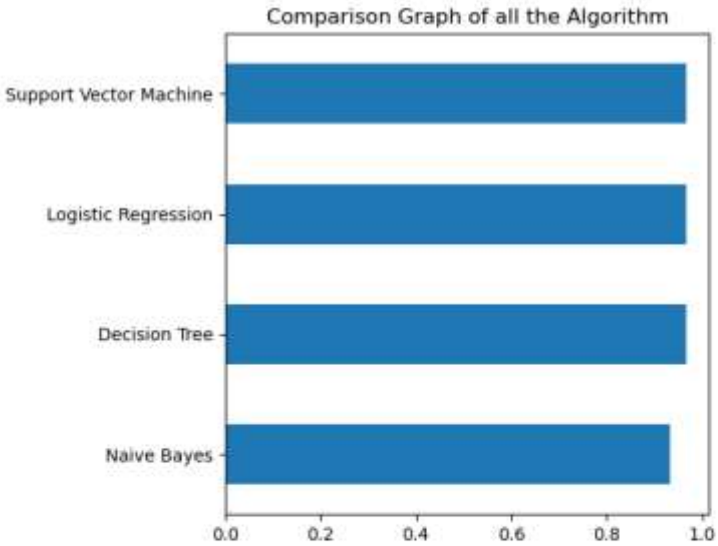

count
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
LIGHT

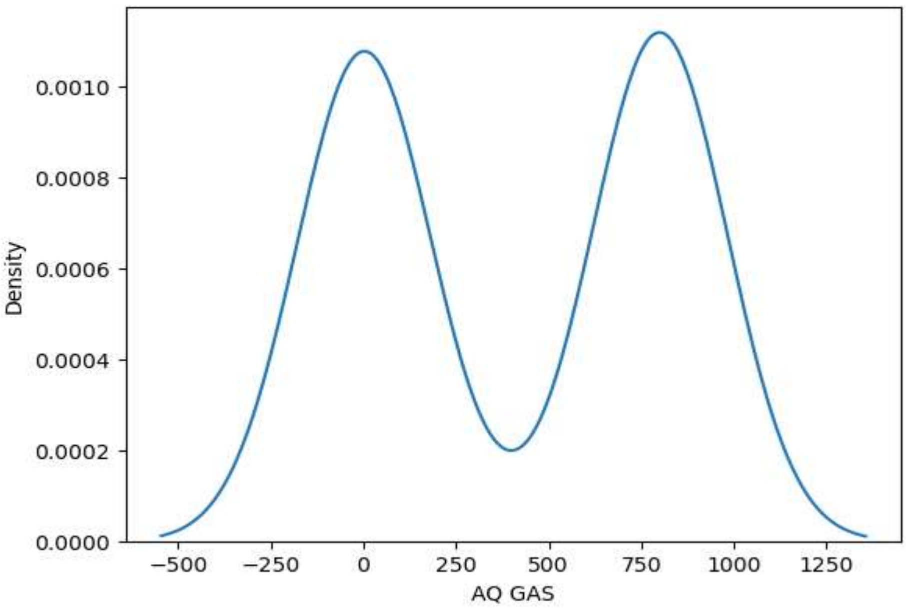

Scatter Plot using PCA
Principal Component 1
Principal Component 2
Species


Line Plot of Features
AQ GAS
TEMP
LIGHT
Values
800
600
400
200
0
0 10 20 30 40 50
Data Point


19
0
0
0
20
1
0
1
19
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0


19
0
0
0
20
1
0
1
19
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0


Comparison Graph of all the Algorithm
Support Vector Machine
Logistic Regression
Decision Tree
Naive Bayes
0.0
0.2
0.4
0.6
0.8
1.0


Density
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
−500
−250
0
250
500
750
1000
1250
AQ GAS

CONCLUSION

In conclusion,an Healthy Me(Helath Monitoring System Using IOT and ML) offers a highly efficient , automated solutionfor human beings to monitor their health condition based on Blood Pressures,Heartrate,Temperature and humidity and etc……..By Combining the Health monitoring machines with IOT capabilities,this system enhances productivity, accuracy, and scalability , making it as a valuable for the human beings in real life by elobarating this one.

Reference

https://www.researchgate.net/publication/343896361_Smart_Health_Monitoring_System_using_I OT_and_Machine_Learning_Techniques

https://ieeexplore.ieee.org/document/9132937

https://www.researchgate.net/publication/341747187_Machine_Learning_based_Health_Monitoring_System

https://ieeexplore.ieee.org/document/10027145

Wikipedia, the free encyclopedia

https://www.academia.edu/44254020/IoT_BASED_HEALTH_MONITORING_SYSTEM

https://www.researchgate.net/publication/378605383_IoT_Based_HEALTH_Patient_Monitoring_System

https://ncbi.nlm.nih.gov/pmc/articles/PMC9601552/

IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.