IoT Enabled Home Security and Safety Solution: Door Automation

Using Grovator platform & this working model, students would be able to develop new innovative cloud based smart home security and safety solutions

1. Expanding the system to include video surveillance and motion detection.
2. Implementing advanced access control features like biometric authentication, fingerprint, facial recognition.
3. Enhance garage door security with remote monitoring and control.
4. Develop a system that integrates video-enabled smart doorbells with package detection, remote access for secure delivery, and instant alerts for arrivals.

Students' learning outcome

5. Using the Grovator® platform, students can develop new innovative solutions for advanced smart home security and safety technology.
6. Understanding of interfacing hardware components (IR, servo motor) with an IoT Raspberry Pi gateway.
7. Proficiency in programming languages like Python for hardware control and data processing.
8. Implementing MQTT protocol for secure and efficient data exchange between the Raspberry Pi gateway and AWS IoT Core, facilitating real-time notifications and control within the home security system

Functional working

The IoT-enabled Smart Door Security Solution utilizes an IR sensor for access control and presence detection. When a person approaches the door, the IR sensor detects their presence, triggering the Raspberry Pi to activate the servo motor, thereby unlocking the door. Simultaneously, the system transmits real-time notifications of presence and door status to the homeowner's mobile application via the Raspberry Pi-based IoT gateway and the AWS cloud. This ensures immediate awareness of all access events and nearby activity, enhancing overall security and providing timely alerts.

Technology component used

Raspberry Pi - IoT Gateway, IR Sensor , Servomotor

Communication protocol used

9. **GPIO:** Direct hardware interaction for IR Sensor (Digital Pins) & Servo Motor (PWM Pins)
10. **MQTT:** Secure, real-time data exchange between Raspberry Pi and AWS IoT Core for notifications and remote commands.

SYSTEM CONFIGURATION

1. **Raspberry Pi (IoT Gateway):** Acts as the central processing unit, processing sensor data , controlling actuators , and enabling internet connectivity via Wi-Fi or Ethernet.
2. **IR Sensor:** Connected to a GPIO input pin of the Raspberry Pi for presence detection.
3. **Servo Motor:** Connected to Raspberry Pi GPIO pins (PWM) to control the door locking mechanism.

SOFTWARE CONFIGURATION

4. The Raspberry Pi-based IoT gateway powers up, initializing GPIO pins for IR sensor input and servo motor control.
5. The IoT Gateway establishes a Wi-Fi connection and uses MQTT to transmit door status, presence detection logs, and receive remote unlock commands via the AWS IoT cloud.
6. The IoT Gateway continuously monitors the IR sensor input via GPIO communication, detecting presence near the door.
7. When the IR sensor detects presence, the IoT gateway triggers the servo motor to unlock the door.
8. When presence is detected or the door is remotely unlocked, the IoT gateway triggers logging of the event.
9. When remote unlock commands are received via MQTT, the IoT gateway activates the servo motor to unlock the door.

VISUAL DIAGRAM

IoT Enabled Home Security and Safety Solution: Digital Lock

Using Grovator platform & this working model, students would be able to develop new innovative cloud based smart home security and safety solutions

10. Expanding the system to include video surveillance and motion detection.
11. Implementing advanced access control features like biometric authentication, fingerprint, facial recognition.
12. Develop a system that integrates video-enabled smart doorbells with package detection, remote access for secure delivery, and instant alerts for arrivals.

Students' learning outcome

13. Using the Grovator® platform, students can develop new innovative solutions for advanced smart home security and safety technology.
14. Understanding of interfacing hardware components (keypad, servo motor, buzzer) with a IoT Raspberry Pi gateway.
15. Implementing basic security measures like password authentication and real-time alerts.
16. Proficiency in programming languages like Python for hardware control and data processing.
17. Implementing MQTT protocol for secure and efficient data exchange between the Raspberry Pi gateway and AWS IoT Core, facilitating real-time notifications and control within the home security system

Functional working

The IoT-enabled Home Security & Safety Solution provides secure door access through a keypad interface. When a user enters the correct password on the keypad, the system validates it, and the servo motor activates, unlocking the door. If an incorrect password is entered, the servo motor remains locked, and the buzzer sounds an alarm. Simultaneously, regardless of success or failure, the system transmits real-time notifications to the homeowner's mobile application via an IoT gateway (Raspberry Pi) and the AWS cloud, informing them of the password attempt and door status. This ensures immediate awareness of any security events.

Technology component used

Raspberry Pi - IoT Gateway, Keypad , Servomotor, buzzer

Communication protocol used

18. **GPIO:** Direct hardware interaction for keypad and buzzer control (Digital Pins) & Servo Motor (PWM Pins)
19. **MQTT:** Secure, real-time data exchange between Raspberry Pi and AWS IoT Core for notifications and remote commands.

SYSTEM CONFIGURATION

- **Raspberry Pi (IoT Gateway):** Acts as the central processing unit, processing sensor data , controlling actuators , and enabling internet connectivity via Wi-Fi or Ethernet.
- **Keypad:** Connected to GPIO pins of the Raspberry Pi for password input.
- **Servo Motor:** Connected to Raspberry Pi GPIO pins (PWM) to control the door locking mechanism.
- **Buzzer:** Connected to a GPIO output pin of the Raspberry Pi for audible alerts.

SOFTWARE CONFIGURATION

- The Raspberry Pi based IoT gateway powers up, GPIO pins for keypad input, buzzer output, and servo motor control.
- The IoT Gateway establishes a Wi-Fi connection and uses MQTT to transmit lock status, access logs, and receive remote unlock commands via AWS IoT cloud.
- The IoT Gateway continuously monitors keypad input via GPIO communication, collecting password entries.
- When a password entry is acquired, the IoT gateway executes an algorithm to verify the password against stored authorized passwords.
- When calculated, the Raspberry Pi controls the servo motor to unlock the door upon successful verification, and triggers the buzzer for incorrect password entries.
- When password entries are verified or failed, the IoT gateway triggers logging of the event.
- When remote unlock commands are received via MQTT, the IoT gateway activates the servo motor to unlock the door.

VISUAL DIAGRAM

IoT Enabled Home Security Solution: Security Solution using IR Sensor, Camera & Buzzer

Using Grovator platform & this working model, students would be able to develop new innovative cloud based smart home security and safety solutions

20. Expanding the system to include video surveillance and motion detection.
21. Monitoring elderly people living alone, and providing alerts based on falls, or unusual activity.
22. Enhance garage door security with remote monitoring and control.
23. Monitor pet activity and prevent them from escaping.
24. Enhance garage door security with remote monitoring and control.

Students' learning outcome

25. Using the Grovator® platform, students can develop new innovative solutions for advanced smart home security and safety technology.
26. Understanding of interfacing hardware components (IR, servo motor, USB Camera) with a IoT Raspberry Pi gateway.
27. Implementing basic security measures like presence detection, image capture, and real-time alerts.
28. Proficiency in programming languages like Python for hardware control and data processing.
29. Implementing MQTT protocol for secure and efficient data exchange between the Raspberry Pi gateway and AWS IoT Core, facilitating real-time notifications and control within the home security system

Functional working

The smart door access control system utilizes an IR sensor for basic presence detection and a camera for capturing intruder images. When a person approaches the door, the IR sensor detects their presence, triggering the Raspberry Pi. The Raspberry Pi then controls the servo motor to unlock the door. If the IR sensor detects presence for an extended period, or if the door is opened without authorization, the camera captures an image of the person. This image and a notification are sent to the homeowner's mobile application via the Raspberry Pi-based IoT gateway and the AWS cloud. The system also activates a buzzer during unauthorized access. The servo motor is used to lock and unlock the door.

Technology component used

Raspberry Pi - IoT Gateway, USB Camera , Servomotor, buzzer, IR sensor

Communication protocol used

30. **GPIO:** Direct hardware interaction for IR Sensor, buzzer (Digital Pins) & Servo Motor (PWM Pins)
31. **MQTT:** Secure, real-time data exchange between Raspberry Pi and AWS IoT Core for notifications and remote commands.

SYSTEM CONFIGURATION

32. **Raspberry Pi (IoT Gateway):** Central microprocessor managing system operations, sensor data, image capture, and cloud connectivity via Wi-Fi/Ethernet**.**
33. **IR Sensor:** Connected to a GPIO input pin of the Raspberry Pi for presence detection.
34. **Camera:** Captures images of intruders, connected via USB.
35. **Servo Motor:** Controls door locking, connected to Raspberry Pi GPIO (PWM).
36. **Buzzer:** Emits audible alerts for unauthorized access, connected to Raspberry Pi GPIO.
37. **Mobile App:** Sends access notifications and intruder images, and potentially controls the system remotely via cloud communication.

SOFTWARE CONFIGURATION

- The IoT Gateway powers up, configures its GPIO pins, and establishes a connection with the cloud via Wi-Fi or Ethernet to handle system operations.
- The IR sensor continuously monitors for presence, triggering the servo motor to unlock the door when someone approaches.
- If the IR sensor detects prolonged presence or unauthorized door opening, the Raspberry Pi captures an image using the camera.
- The Raspberry Pi sends the captured image and alert notifications to the mobile app via MQTT.
- The Raspberry Pi triggers the buzzer during unauthorized access.
- The mobile app connects to the cloud via WebSocket to receive real-time data updates, including intruder images, and sends push notifications when unauthorized access is detected, allowing authorized users to monitor and control the system remotely.
- Remote unlock commands via MQTT activate the servo motor.

VISUAL DIAGRAM

IoT-enabled Smart Energy Management using smart lighting System: industry working model

Using Grovator platform & this working model, students would be able to develop new innovative cloud based smart energy management solutions

38. Design a system that Balances energy supply and demand using real-time data from smart meters.
39. Develop a system which Controls window blinds based on sunlight and time of day for energy efficiency.
40. Design a system that Optimizes building temperature control based on zone occupancy and usage.

Students' learning outcome

1. Using the Grovator® platform, students can develop new innovative solutions for advanced smart energy management.
2. Understanding of interfacing hardware components (IR sensor, LDR, LED) with an IoT Raspberry Pi gateway.
3. Implementing energy-saving algorithms based on sensor data.
4. Proficiency in programming languages like Python for hardware control and data processing.
5. Implementing MQTT protocol for secure and efficient data exchange between the Raspberry Pi gateway and AWS IoT Core, facilitating real-time monitoring and control.

Functional working

The smart lighting system utilizes an IR sensor for occupancy detection and an LDR for ambient light level detection. When the IR sensor detects presence in a room, or when the LDR detects low ambient light, the Raspberry Pi activates the LED light. The LDR can also be used to detect motion if the light level changes rapidly, such as the shadow of a person. The system adjusts the LED light's brightness based on the LDR's readings to maintain optimal lighting while minimizing energy consumption. LED indicators provide visual feedback on the system's status. Real-time data and control commands are transmitted via MQTT to a mobile app for remote monitoring and adjustments.

Technology component used

Raspberry Pi - IoT Gateway, IR Sensor ,LDR (Light Dependent Resistor) ,LED Light.

Communication protocol used

41. GPIO: IR uses digital input, LDR requires an ADC for analog input, and LED control uses digital output or PWM.
42. IoT Gateway publishes sensor data to MQTT topics on AWS IoT Core

System configuration

6. Raspberry Pi (IoT Gateway): Central microprocessor managing sensor data, lighting control, and cloud connectivity via Wi-Fi/Ethernet.
7. IR Sensor: Connected to a GPIO input pin of the Raspberry Pi for occupancy detection.
8. LDR: Connected to an analog input pin of the Raspberry Pi for ambient light level detection.
9. LED Light: Connected to a GPIO output pin (PWM) of the Raspberry Pi for brightness control.
10. LED Indicators: Connected to GPIO output pins of the Raspberry Pi for visual feedback.
11. Mobile App: Sends monitoring data and control commands, and displays system status..

Software configuration

12. The IoT Gateway powers up, configures its GPIO pins, and establishes a connection with the cloud via Wi-Fi or Ethernet.
13. The LED indicators display system status (e.g., occupancy, light level, energy consumption).
14. The Raspberry Pi sends sensor data and system status to the mobile app via MQTT.
15. The mobile app connects to the cloud via WebSocket to receive real-time data and send control commands.
16. Remote control commands via MQTT adjust LED light brightness or turn the LED light on/off.
17. The LDR is used to detect motion, and the system logs motion events.

Visual diagram

IoT-ENABLED SMART IRRIGATION SOLUTION: INDUSTRY WORKING MODEL

USING GROVATOR® PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SUSTAINABLE FARMING SOLUTIONS

1. Develop a system which Precisely irrigates distinct garden zones based on their individual soil moisture requirements.

2. Develop a system using flow and pressure sensors to identify and mitigate water leaks, conserving resources.

STUDENTS' LEARNING OUTCOME

3. By using Grovator®, students can create innovative new solutions for Smart agriculture.

4. Understanding of interfacing hardware components (moisture sensor, solenoid valve) with an IoT Raspberry Pi gateway.

5. Implementing water-saving algorithms based on soil moisture data.

6. Proficiency in programming languages like Python for hardware control and data processing.

7. Implementing MQTT protocol for secure and efficient data exchange between the Raspberry Pi gateway and AWS IoT Core, facilitating real-time monitoring and control.

FUNCTIONAL WORKING

The smart irrigation system utilizes a moisture sensor to monitor soil moisture levels. When the moisture sensor detects that the soil moisture falls below a predefined threshold, the Raspberry Pi activates the solenoid valve, allowing water to flow. The system adjusts the irrigation duration based on the moisture sensor's readings to maintain optimal soil moisture while minimizing water consumption. Real-time data and control commands are transmitted via MQTT to a mobile app for remote monitoring and adjustments.

TECHNOLOGY COMPONENTS USED

Raspberry Pi - IoT Gateway , Soil Moisture Sensor , Solenoid Valve

COMMUNICATION PROTOCOL USED

1. GPIO: Direct hardware interaction for moisture sensor (Analog Input) and solenoid valve control (Digital Output).
2. MQTT: Secure, real-time data exchange between Raspberry Pi and AWS IoT Core for notifications and remote commands.

SYSTEM CONFIGURATION

- **Raspberry Pi (IoT Gateway):** Central microprocessor managing sensor data, irrigation control, and cloud connectivity via Wi-Fi/Ethernet.
- **Moisture Sensor:** Connected to an analog input pin of the Raspberry Pi for soil moisture level detection.
- **Solenoid Valve:** Connected to a GPIO output pin of the Raspberry Pi for water flow control.
- **Mobile App:** Sends monitoring data and control commands, and displays system status..

SOFTWARE CONFIGURATION

3. The IoT Gateway powers up, configures its GPIO pins, and establishes a connection with the cloud via Wi-Fi or Ethernet.
4. The moisture sensor continuously monitors soil moisture levels.
5. The Raspberry Pi activates the solenoid valve when soil moisture falls below the threshold.
6. The Raspberry Pi adjusts the irrigation duration based on moisture sensor readings.
7. The Raspberry Pi sends sensor data and system status (e.g., moisture levels, irrigation status) to the mobile app via MQTT.
8. The mobile app connects to the cloud via WebSocket to receive real-time data and send control commands.
9. Remote control commands via MQTT adjust irrigation duration or turn irrigation on/off.
10. The system logs irrigation events and moisture data.

IoT-ENABLED SMART WATER QUALITY TESTING : INDUSTRY WORKING MODEL

USING GROVATOR® PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SUSTAINABLE FARMING SOLUTIONS

11. Automated hydroponic systems using the nutrient film technique for space-efficient, hygienic, water-saving, and uniform crop production in indoor and rooftop farming.
12. Smart lighting control systems to enhance energy efficiency and plant growth, paired with organic farming solutions for pesticide-free, sustainable hydroponic cultivation.
13. Smart hydroponic systems designed for space missions, addressing challenges like limited space and resources.
14. Design smart compact hydroponic systems for both decorative and festival purposes.
15. Smart Aquaponics Systems by combining IoT-based Hydroponics with fish farming to optimize resource use

STUDENTS' LEARNING OUTCOME

16. By using Grovator®, students can create innovative new solutions for Smart agriculture.
17. Learn to connect and utilize analog sensors (pH, TDS) including calibration techniques for accurate data acquisition.
18. Learning how to securely connect different IoT Gateways to the cloud using MQTT & CoAP.
19. Learn to store data in the cloud database & monitor different parameters for plant growth.

FUNCTIONAL WORKING

The hydroponics system operates with minimal water consumption and leverages IoT connectivity through various sensors. A TDS sensor monitors the mineral content in the water and determines the required levels for optimal plant growth. A pH sensor assesses whether the water is acidic or basic, ensuring it is suitable for the plants. A relay system controls the timely activation of the water pump and LED lights to maintain an optimal growing environment. All data, including sensor readings and plant growth updates, are transmitted to a mobile device via an internet cloud for real-time tracking and management.

TECHNOLOGY COMPONENTS USED

PH sensor, TDS Sensor

COMMUNICATION PROTOCOL USED

20. Analog Communication for TDS and pH sensor
21. MQTT for communication with the cloud
22. Websocket for real time data exchange

SYSTEM CONFIGURATION

1. The pH sensor connects to Analog-1 to monitor solution acidity for nutrient absorption.
2. The TDS sensor connects to Analog-2 to ensure proper nutrient concentration.
3. The onboard RTC uses I2C communication to automate lighting (ON at 6:00 AM, OFF at 6:00 PM).
4. The IoT gateway collects sensor data and sends it to the cloud for remote access.
5. Edge computing program is running on the IoT Gateway.
6. The mobile app displays pH, TDS.

SOFTWARE CONFIGURATION

7. **IoT Gateway** powers up, configures certain **GPIO pins** which are used for input (for reading sensor data), **I2C Bus** (for communication with sensors), **RTC** (real-time clock).
8. It establishes a wired connection with the temperature sensor, TDS (Total Dissolved Solids) sensor, pH sensor and camera.
9. The IoT Gateway continuously monitors the data from the sensors **(temperature, humidity, time, TDS, pH)**, which is then used for the real-time sensor data transmission for effective farming of crops.
10. It sends messages to the mobile app, i.e., "pH value," "TDS value," in real-time.

Address: Grok Learning Pvt. Ltd., 3rd Floor, A-Wing, Amba Bhavan, Plot No. 29, Next to Bharat petrol pump, Sion Circle (West), Mumbai-400022, India | **Phone:** +91 93245 02127 | **Email:** sales@grokstem.com

IoT-ENABLED SMART HEALTHCARE APPLICATION FOR PULSE RATE & OXYGEN MONITORING: INDUSTRY WORKING MODEL

USING GROVATOR PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SMART MEDICAL DEVICES & SOLUTIONS

11. Students can use GSR sensors to build healthcare monitoring and research systems like post traumatic stress disorder and depression monitoring systems.
12. Smart healthcare devices to monitor the temperature of patients by using thermal sensors.
13. Smart thermal devices to monitor medical equipment temperature and ensure the proper storage of temperature-sensitive medications.

STUDENTS' LEARNING OUTCOME

14. Using the Grovator® platform, students can develop new innovative smart medical devices & services for healthcare industries.
15. Learn how to connect and manage multiple healthcare sensors like , Pulse Oximeter simultaneously using different communication protocols.
16. Learn how to calibrate different sensors to improve data accuracy.
17. Learn how to securely connect different IoT Gateways to the cloud using MQTT & CoAP.
18. Learn to store data in the cloud database, apply AI-data models to provide forecasting and create smart health data monitoring dashboard.

FUNCTIONAL WORKING

The system focuses on monitoring heart rate and blood oxygen levels. The individual places their forefinger on a pulse oximeter, which measures and displays SpO2 levels and heartbeat readings on both the LCD screen and a mobile app

TECHNOLOGY COMPONENTS USED

IoT Gateway, 1 LCD Display, 1 LED Module, 1 Pulseoximeter sensor, ,3D Printed Parts.

COMMUNICATION PROTOCOL USED

19. I2C for interfacing with the LCD display
20. I2C for getting heartbeat and SPo2 data from pulse-oximeter.
21. Digital (HIGH/LOW) communication for LED Module and peristaltic pump.
22. MQTT for communication with the cloud
23. Websocket for real time data exchange
24. Analog input communication for GSR sensor data.

SYSTEM CONFIGURATION

25. The system is designed around a microprocessor-based IoT Gateway, which connects to the IoT cloud via Wi-Fi or Ethernet to manage all system operations seamlessly. Edge computing program is running on the IoT Gateway.
26. **Heart rate and Spo2 Monitoring :** A pulse oximeter is used to measure heart rate and SpO2 levels. The sensors and motors are connected to the gateway's I2C port.
27. **Displays:**
    1. An LCD display at the top shows all readings and provides step-by-step instructions for an improved user experience.
    2. An LED display indicates whether data collection by the sensors is in progress or completed.

SOFTWARE CONFIGURATION

28. The **IoT Gateway** powers up, configures its **GPIO pins,** and displays a prompt on the LCD to sanitize hands.
29. Next, the LCD prompts the user to place their finger on the **pulse oximeter sensor.**

30. When the finger is placed, the **pulse oximeter** collects data through the **I2C port** and transmits it to the **IoT Gateway.**
31. The gateway processes the data and displays it on the LCD screen.
32. The IoT Gateway processes the data, and the final readings are displayed on the **LCD** screen using the **I2C port.**

IoT-ENABLED SMART STRESS MONITORING APPLICATION: INDUSTRY WORKING MODEL

USING GROVATOR PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SMART MEDICAL DEVICES & SOLUTIONS

33. Cloud-connected system enabling remote healthcare professionals to monitor and intervene in patient stress levels.
34. Aggregates employee stress data for HR to improve workplace wellness with personalized resources.
35. Real-time data alerts educators to students in distress, supported by stress management education.

STUDENTS' LEARNING OUTCOME

36. Using the Grovator® platform, students can develop new innovative smart medical devices & services for healthcare industries.
37. Learn how to connect and manage multiple healthcare sensors like GSR etc. simultaneously using different communication protocols.
38. Learn how to calibrate different sensors to improve data accuracy.
39. Learn how to securely connect different IoT Gateways to the cloud using MQTT & CoAP.
40. Learn to store data in the cloud database, apply AI-data models to provide forecasting and create smart health data monitoring dashboard.

FUNCTIONAL WORKING

The Smart Healthcare Station begins by allowing individuals to sanitize their hands using an automatic dispenser, with messages displayed on an LCD screen to guide the process. Following this, the individual places their fingers on a GSR sensor, which analyzes and displays their stress level as low, medium, or high. Once measurement is completed, the readings are displayed on both the mobile app & the LCD screen for easy monitoring.

TECHNOLOGY COMPONENTS USED

IoT Gateway, 1 LCD Display, 1 LED Module, 1 GSR sensor, 1 Pulseoximeter sensor,3D Printed Parts.

COMMUNICATION PROTOCOL USED

41. I2C for interfacing with the LCD display
42. Digital (HIGH/LOW) communication for IR sensor, LED Module and peristaltic pump.
43. MQTT for communication with the cloud
44. Websocket for real time data exchange
45. Analog input communication for GSR sensor data.

SYSTEM CONFIGURATION

46. The system is designed around a microprocessor-based IoT Gateway, which connects to the IoT cloud via Wi-Fi or Ethernet to manage all system operations seamlessly. Edge computing program is running on the IoT Gateway.
47. **Stress Monitoring :** A GSR sensor is incorporated to monitor stress levels. The GSR sensor is connected to the gateway's analog port.
48. **Displays:**
    1. An LCD display at the top shows all readings and provides step-by-step instructions for an improved user experience.
    2. An LED display indicates whether data collection by the sensors is in progress or completed.

SOFTWARE CONFIGURATION

49. The **IoT Gateway** powers up, configures its **GPIO pins,** and displays a prompt on the LCD to sanitize hands.
50. Once the hands are sanitized, the **IoT Gateway** updates the LCD display with the message to place a hand on the GSR sensor.
51. When the user places their hand on the sensor, it begins collecting data and sends it to the IoT Gateway via the **Analog input port.**
52. The **IoT Gateway** processes the received data, calculates the average, and displays the final count on the LCD.

53. The IoT Gateway processes the data, and the final readings are displayed on the **LCD** screen using the **I2C port.**

IoT-ENABLED SMART AIR QUALITY MONITORING SYSTEM: INDUSTRY WORKING MODEL

USING GROVATOR PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SMART AIR QUALITY MONITORING SOLUTIONS

54. Design systems to monitor specific pollution sources, such as industrial emissions, construction sites, and traffic hotspots.
55. Develop a distributed network of air quality sensors across a city or region, creating a real-time air quality map.
56. Develop systems that automatically adjust ventilation in buildings and public spaces based on real-time air quality data.

STUDENTS' LEARNING OUTCOME

57. Using the Grovator® platform, students can develop and implement IoT systems for real-time air quality monitoring, contributing to sustainable smart city initiatives.
58. learn to connect and manage particulate matter (PM) sensors, and control LED indicators, simultaneously using various communication protocols like Serial and GPIO.
59. learn how to process sensor data to determine air quality categories and translate this information into visual feedback using LED indicators.
60. learn to securely connect IoT gateways to the cloud using MQTT or HTTP for real-time data transmission and remote monitoring.
61. gain experience in creating systems that provide real-time air quality data and generate alerts based on predefined thresholds, enhancing public awareness and safety.

FUNCTIONAL WORKING

The PM sensor continuously monitors air quality, sending data to the IoT Gateway. The Gateway processes this data, determines the air quality category (safe, moderate, hazardous), and controls the LED indicators accordingly (green, yellow, red). Simultaneously, the Gateway transmits the data to the cloud, enabling remote monitoring and alerts through a mobile app.

TECHNOLOGY COMPONENTS USED

IoT Gateway, Particulate Matter Sensor, LED

COMMUNICATION PROTOCOL USED

62. **Wi-Fi/Ethernet**: Facilitates communication between the IoT Gateway and the IoT cloud, enabling seamless data transmission and remote management.
63. **GPIO**: Local communication protocol for interfacing the LED indicators with the IoT Gateway, enabling the control of the LED colors based on air quality
64. **Serial Communication (UART):** Local communication protocol for interfacing the particulate matter (PM) sensor with the IoT Gateway, allowing for the transfer of PM data.
65. **MQTT or HTTP**: Used for transmitting data from the IoT Gateway to the cloud for real-time monitoring and status updates.

SYSTEM CONFIGURATION

1. **IoT Gateway and Cloud Integration**: The system is powered by a microprocessor-based IoT Gateway that connects to the cloud via Wi-Fi or Ethernet. Edge computing program is running on

the IoT Gateway. It collects and transmits real-time data about particulate matter (PM2.5 and PM10) levels to the cloud for remote monitoring.

2. **Air Quality Monitoring**: A particulate matter (PM) sensor continuously measures the PM levels in the ambient air. Based on the PM readings, the sensor determines the air quality category (safe, moderate, hazardous).
3. **Visual Air Quality Indication**: The system uses RGB LED indicators to visually represent the air quality category. If the air quality is safe, the LED shows green. If moderate, it shows yellow. If hazardous, it shows red.

4. **Real-Time Data Communication**: The IoT Gateway communicates with the cloud using protocols like MQTT or HTTP, sending data regarding PM levels and air quality category. This allows for remote monitoring, alerts, and data analysis.

SOFTWARE CONFIGURATION

1. The IoT Gateway powers up, initializes its GPIO pins, and begins reading particulate matter (PM) data from the PM sensor, determining the real-time air quality category.
2. It establishes a wireless connection to the IoT cloud via Wi-Fi or Ethernet, enabling real-time communication to monitor air quality status and manage system operations efficiently.
3. The IoT Gateway continuously checks the PM levels via the PM sensor, and the real-time air quality data is transmitted to the cloud for analysis and management.
4. The IoT Gateway processes the PM data to determine the air quality category (green, yellow, or red) and sets the RGB LED indicators accordingly.
5. When the PM levels exceed predefined threshold values, the system triggers alerts to the cloud and, optionally, to the mobile application
6. The IoT Gateway communicates continuously with the cloud to update the air quality status, ensuring real-time monitoring and management.

IoT- ENABLED SMART SALINE MONITORING: INDUSTRY WORKING MODEL

USING GROVATOR PLATFORM & THIS WORKING MODEL, STUDENTS WOULD BE ABLE TO DEVELOP MANY INNOVATIVE CLOUD BASED SMART HEALTHCARE SERVICES

7. Smart healthcare dispensers which can automatically dispense saline, medicine as required by the patients.
8. Smart healthcare monitoring systems which can automatically trigger alerts to nurses and doctors based on the real-time data of patients' health conditions.
9. Smart digital infant weighing scales for measurement and monitoring infants body weight in pediatric department
10. Smart fluid intake and output monitoring system track bags and urine containers

STUDENTS' LEARNING OUTCOME

11. Using the Grovator® platform, students can develop new innovative smart medical devices & services for healthcare industries.
12. Learn to connect and manage multiple healthcare sensors, such as load cell sensors, simultaneously using various communication protocols.
13. Learn to calibrate various sensors to enhance data accuracy.
14. Learn how to securely connect different IoT Gateways to the cloud using MQTT & CoAP.
15. Learn to store data in a cloud database, apply AI-driven models for forecasting, and develop a smart health data monitoring and alert triggering dashboard.

FUNCTIONAL WORKING

Smart saline monitoring uses a load sensor or load cell to measure the saline level in the bottle accurately. When the saline level is sufficient, a green LED indicator is activated. Upon pressing the drain switch on the panel, the saline level begins to drop. Once the level falls below threshold, the LED indicator turns red, and an alert buzzer sounds. The live saline percentage is displayed on an LCD screen and simultaneously notified to the nurse via a mobile phone, ensuring timely intervention and efficient patient care.

TECHNOLOGY COMPONENTS USED

IoT Gateway, 2 DC motor pump, 2 SPST switch, 1 LCD Display, 1 LED Module, 1 Load cell, 3D Printed Parts.

COMMUNICATION PROTOCOL USED

16. I2C for interfacing with the LCD display
17. I2C protocol for receiving data from thermal sensors.
18. Digital (HIGH/LOW) communication for Load cell, LED Module and peristaltic pump.
19. MQTT for communication with the cloud
20. Websocket for real time data exchange

SYSTEM CONFIGURATION

21. The system is designed around a microprocessor-based IoT Gateway, which connects to the IoT cloud via Wi-Fi or Ethernet to manage all system operations seamlessly. Edge computing program is running on the IoT Gateway.
22. **Saline monitoring**: The system includes a load cell sensor connected to the GPIO port for continuous monitoring of the saline level.
23. **Drain Switch :** An SPST drain switch is used to transfer saline water from one bottle to another, causing a change in the saline level. This switch is connected to the gateway's relay port..

24. **Fill Switch :** An SPST fill switch is used to refill the saline level back to 100% once it is fully drained. This switch is also connected to the gateway's relay port.
25. **Buzzer Indicator :**A buzzer, connected to the GPIO port, is activated when the saline level drops below a specified threshold.
26. **Displays:**
    1. An LCD display shows the live percentage of the saline level.
    2. An LED display indicates the saline status as sufficient, medium, or below the threshold.

SOFTWARE CONFIGURATION

The IoT Gateway powers up, configures its GPIO pins, and displays the current saline level percentage on the LCD screen.

27. If the saline bottle is full, the IoT Gateway uses LCD screen to display the percentage is above 60 using I2C protocol.
28. Once the drain switch is turned on, the saline level begins to drop.
29. The IoT Gateway uses I2C communication to update the real-time percentage on the display
30. When the saline level drops below 30%, the IoT Gateway sends a signal to the LED and buzzer module via the GPIO pin, indicating a low saline level.