Amrita Vishwa Vidyapeetham

Amrita School of Computing, Coimbatore

Department of Computer Science and Engineering

19CSE446 – Internet of Things

Tutorial (Design Based)

Date: 19 May 2024

Group No: 13

SNo	Roll No	Name
1	CB.EN.U4CSE21003	Adhvaith Sankar S S
2	CB.EN.U4CSE21004	Ajay Raj M
3	CB.EN.U4CSE21048	Rohith M

Smart Waste Management System

Table Of Contents

1. Objective

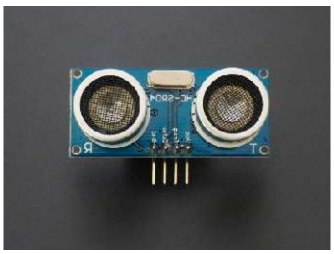
The objective of this project is to design and implement a smart waste management system using IoT technology. This system is intended to address common inefficiencies in traditional waste collection methods by leveraging real-time data and advanced communication technologies. The main goals are to enhance the efficiency of waste collection, reduce operational costs, promote environmental sustainability, and utilize data analytics for continuous improvement in waste management practices. The specific objectives are as follows:

1. **Developing a System for Real-time Monitoring**: Create a reliable and accurate system to continuously monitor the fill levels of waste bins. This involves using sensors to detect and report the current waste level, ensuring timely data collection for effective decision-making.
2. **Implementing Automatic Status Updates**: Develop a mechanism to automatically update the status of waste bins on a digital map. This includes color-coded indicators reflecting the fill levels, providing an intuitive and quick reference for waste collection personnel and management.
3. **Optimizing Waste Collection Routes**: Utilize the collected data to design and implement optimized waste collection routes. This aims to reduce fuel consumption, minimize travel time, and improve overall efficiency in waste collection operations.
4. **Analysing Collected Data**: Gather and analyse data to identify patterns and trends in waste generation. This information can be used to predict future waste levels, adjust collection schedules, and make informed decisions about waste management strategies.
5. **Creating a User-friendly Interface**: Develop an intuitive and accessible interface for waste collection personnel. This interface will display real-time data on bin status, suggested collection routes, and historical data trends, aiding in efficient and effective waste management.

2. <u>Hardware Selection and Design</u>

Sensors:

- **Ultrasonic Sensor:**
  - **Function:** Measures the fill level of waste bins by emitting sound waves and detecting the time it takes for the echo to return, determining the bin's fill level.
  - **Specifications:** Chosen for accuracy, range, and reliability in various environmental conditions. Positioned inside the waste bin at the top to measure the distance to the waste surface.



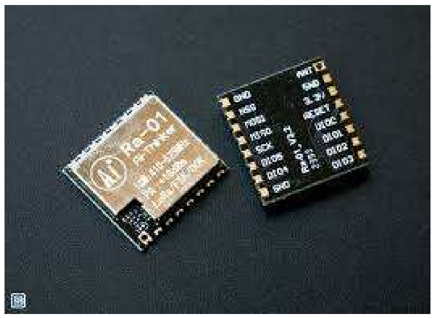
Controller:

- **ESP32 Microcontrollers:**
  - **Inside the Waste Bin:** Connected to the ultrasonic sensor, collects sensor data, and transmits it using the LoRa communication module.
  - **Gateway ESP32:** Equipped with both LoRa and Wi-Fi modules. Receives data from the bin ESP32 via LoRa and uploads it to the cloud through Wi-Fi.

Communications Units:

- **LoRa Module (Communications Unit):**

- **Function:** Enables low-power, long-range wireless communication between the ESP32 inside the bin and the gateway ESP32 over considerable distances without direct line of sight.
- **Specifications:** Operates using radio frequency signals, typically at a frequency of 868 MHz (or 915 MHz depending on the region). Selected for reliable communication in urban environments with obstacles.



- **Wi-Fi Module (Communications Unit):**
  - **Integrated with ESP32:** The ESP32 microcontroller comes with an inbuilt Wi-Fi module, allowing the gateway ESP32 to connect to local Wi-Fi networks.
  - **Function:** After receiving data via LoRa, the Wi-Fi module uploads the data to the cloud server, where it can be processed and accessed remotely.

Software Integration:

MQTT Protocol:

- The code integrates the MQTT protocol for communication with HiveMQ Cloud, a fully managed cloud MQTT platform.
- It uses the MQTTClient library to establish a connection to the HiveMQ Cloud broker and handle MQTT communication.
- The MQTT broker (HiveMQ Cloud) allows the ESP32 devices to publish and subscribe to MQTT topics for data exchange.

LoRa Protocol:

- LoRa communication is used for local communication between ESP32 devices.

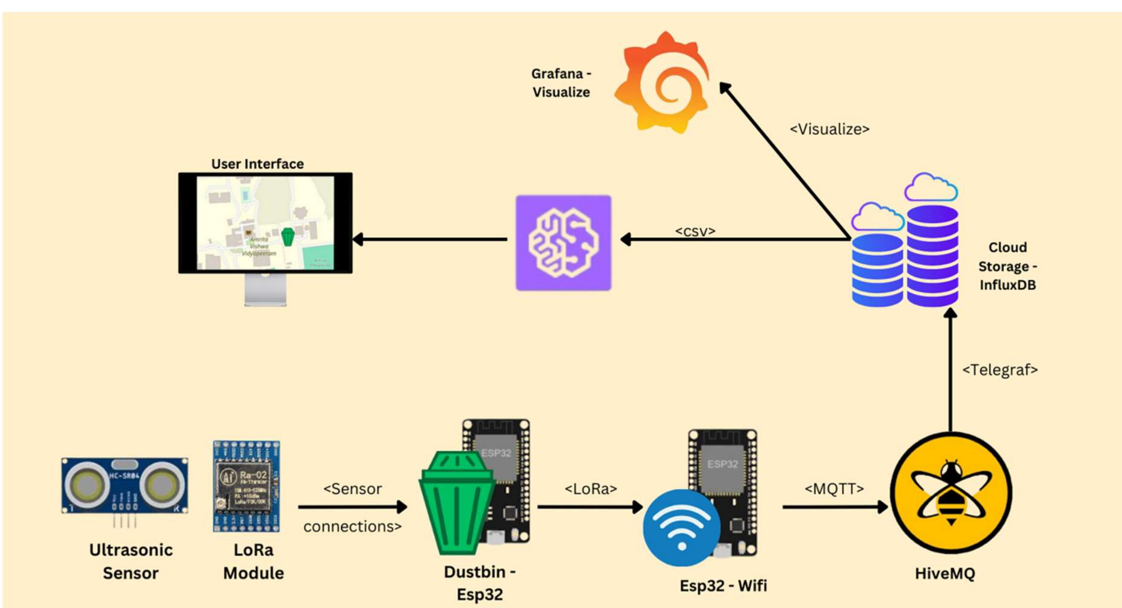
- It enables wireless communication between the ESP32 inside the waste bin and the gateway ESP32.
- LoRa communication is managed using the LoRa library, which handles sending and receiving LoRa packets.

Cloud Integration:

- **HiveMQ Cloud MQTT Broker Connection:**
    - The code connects to HiveMQ Cloud, a fully managed cloud MQTT platform, for MQTT communication.
    - It uses the WiFiClientSecure library for secure Wi-Fi communication.
    - The MQTT broker address, port, username, and password are provided in the code for connecting to the HiveMQ Cloud broker.
- **Data Transmission to HiveMQ Cloud:**
    - Data collected from the waste bin (received via LoRa) is published to specific MQTT topics.
    - The gateway ESP32 subscribes to these topics and publishes the received data to the HiveMQ Cloud MQTT broker for cloud storage and processing.

3. Software Design

Flowchart



Waste Bin ESP32 Pseudo Code:

1. Initialize LoRa communication:

   a. Set up LoRa module pins and parameters.

   b. Initialize LoRa communication.

2. Connect to LoRa network:

   a. Configure LoRa module with network settings (frequency, spreading factor, bandwidth, etc.).

   b. Join the LoRa network.

3. Loop:

   a. Read ultrasonic sensor data:

      i. Trigger the ultrasonic sensor to send a pulse.

      ii. Measure the time taken for the pulse to return (echo).

      iii. Convert the time into distance (in centimetres or inches) based on the speed of sound.

   b. Convert sensor data to desired format (cm):

      i. Convert the distance measurement obtained from the ultrasonic sensor into centimetres.

   c. Prepare LoRa packet with sensor data:

      i. Create a data structure or format for the LoRa packet, including fields for distance in inches and centimetres.

      ii. Populate the packet with the converted sensor data.

   d. Send LoRa packet to destination ESP32 (gateway):

      i. Transmit the prepared LoRa packet containing the sensor data to the destination address (gateway ESP32).

      ii. Handle any error or acknowledgement mechanism to ensure reliable delivery if necessary.

   e. Wait for a short delay before repeating the loop:

      i. Pause execution for a predefined interval to control the frequency of sensor readings and LoRa transmissions.

Gateway ESP32 Pseudo Code:

1. Initialize LoRa communication:

a. Set up LoRa module pins and parameters.

b. Initialize LoRa communication.

2. Connect to LoRa network:

a. Configure LoRa module with network settings (frequency, spreading factor, bandwidth, etc.).

b. Join the LoRa network.

3. Connect to MQTT broker (Cloud):

a. Initialize Wi-Fi connection.

b. Connect to the local Wi-Fi network.

c. Establish a secure connection to the MQTT broker hosted on the cloud server (HiveMQ Cloud).

4. Loop:

a. Listen for incoming LoRa packets:

i. Check if any LoRa packets have been received.

ii. If a packet is received, proceed to process it; otherwise, continue looping.

b. If a packet is received:

i. Extract sensor data from the packet:

- Read and parse the incoming LoRa packet to extract sensor data (distance measurements).

- Identify the format and structure of the incoming data.

ii. Measure distance from the sensor data (cm):

iii. Publish the calculated distance to MQTT broker (Cloud):

- Establish a connection to the MQTT broker.

- Format the calculated distance data into a message payload suitable for MQTT.

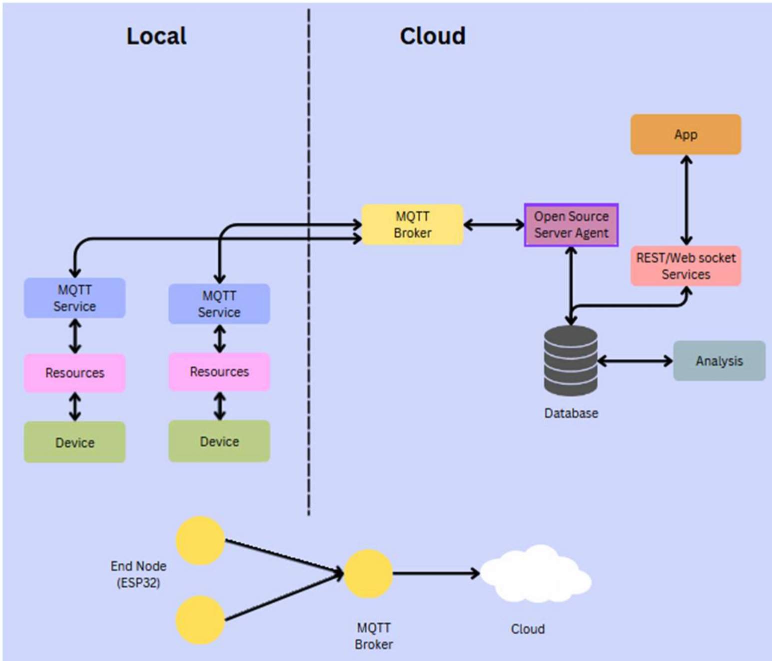
- Publish the distance data to a specific MQTT topic on the cloud broker for storage and processing.

c. Wait for a short delay before repeating the loop:

- Introduce a delay to control the frequency of LoRa packet reception and MQTT publishing.

- Ensure the loop executes at an appropriate interval to balance real-time monitoring with network efficiency.

4. IoT System Design (level 5)



5. Integration of End, Edge and Cloud

1. End Devices (Waste Bin ESP32)

Role:

- **Data Collection:** End devices are responsible for collecting waste level data using ultrasonic sensors.
- **Data Transmission:** They transmit the collected data to the edge devices using LoRa communication.

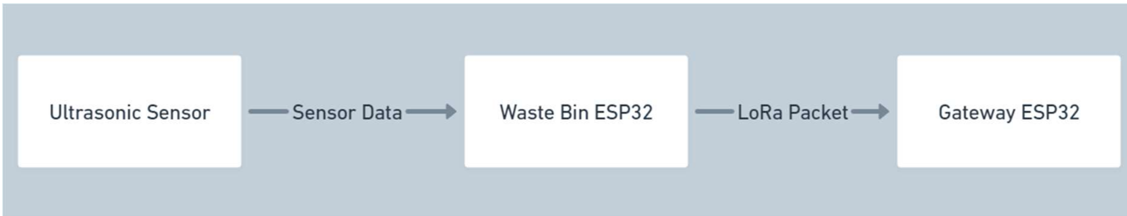
Key Components:

- **Ultrasonic Sensors:** Measure the fill level of waste bins.
- **ESP32 Microcontrollers:** Interface with the sensors and handle data transmission via LoRa.

Process:

1. **Data Collection:**
   - The ultrasonic sensor measures the distance to the waste surface.
2. **Data Formatting:**
   - The measured distance is converted into readable formats (inches/cm).
3. **Data Transmission:**
   - The formatted data is transmitted to the gateway ESP32 via LoRa.

Diagram:



2. Edge Devices (Gateway ESP32)

Role:

- **Data Aggregation and Processing:** Edge device receive data from waste bins, process it, and prepare it for cloud transmission.
- **Data Transmission to Cloud:** They send the processed data to the cloud using Wi-Fi and MQTT.

Key Components:

- **LoRa Module:** Receives data packets from waste bin ESP32 devices.
- **Wi-Fi Module:** Connects to the local network and transmits data to the cloud.
- **MQTT Client:** Manages the communication with the HiveMQ Cloud broker.

Process:

1. **Data Reception:**
    - The gateway ESP32 listens for incoming LoRa packets from the waste bins.
2. **Data Processing:**
    - Extracts sensor data from the received packets and calculates distances (cm).
3. **Data Transmission to Cloud:**
    - Publishes the processed data to the HiveMQ Cloud via MQTT.

Diagram:

3. Cloud Infrastructure (HiveMQ Cloud)

Role:

- **Data Storage and Analysis:** The cloud infrastructure stores the incoming data, processes it for analysis, and provides tools for visualization.

- **User Interface:** It offers a dashboard for real-time monitoring and historical data analysis.

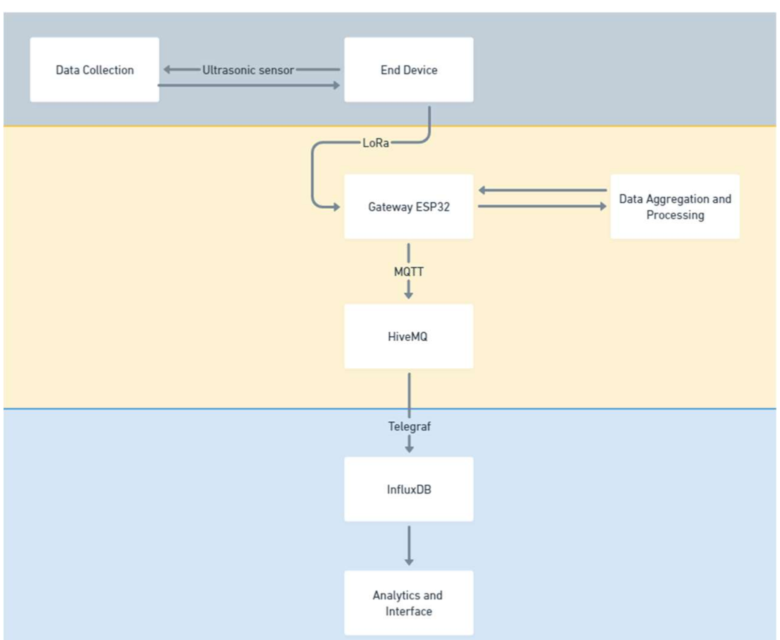
Key Components:

- **MQTT Broker (HiveMQ Cloud):** Manages the communication between the gateway ESP32 and cloud services.
- **Database (InfluxDB):** Stores the received data for historical analysis.
- **Analytics Engine:** Processes the data to provide insights and predictive analytics.
- **Dashboard Interface:** Displays real-time data, historical trends, and suggested waste collection routes.

Process:

1. **Data Reception:**
    - The MQTT broker receives data from the gateway ESP32.
2. **Data Storage:**
    - The data is stored in a database for historical and trend analysis.
3. **Data Analysis:**
    - Analytics engine (SageMaker, Collab) processes the data to identify patterns and optimize waste collection routes.
4. **Data Visualization:**
    - A dashboard interface provides real-time monitoring and visualization of waste levels.

Diagram:



6. Data Accumulation and Processing

The data accumulation and processing are critical components that ensure efficient monitoring and actionable insights. Here's how it works:

Role:

- **Data Aggregation**: Collecting data from end devices and preparing it for analysis.
- **Data Processing**: Cleaning, transforming, and analysing the data to extract meaningful insights and predictions.

Key Components:

- **Edge Device (Gateway ESP32)**: Receives and preprocesses data from end devices (waste bin ESP32).
- **Cloud Infrastructure (HiveMQ Cloud, InfluxDB, Analytics Engine)**: Stores, processes, and analyses the data.
- **Dashboard Interface**: Visualizes processed data for end users.

Process:

1. **Data Reception**:
    - Edge device (Gateway ESP32) listen for incoming data from waste bin ESP32 devices using LoRa communication.
    - Data from multiple bins is aggregated at the gateway.
2. **Data Transmission to Cloud**:
    - The gateway ESP32 uses Wi-Fi and MQTT protocol to transmit the aggregated data to the cloud infrastructure.
    - Data is published to the MQTT broker (HiveMQ Cloud).
3. **Data Storage**:
    - In the cloud, the data is stored in a time-series database like InfluxDB.
    - This database allows efficient storage and retrieval of time-stamped data, facilitating historical analysis.
4. **Data Processing and Analysis**:
    - The analytics engine processes the incoming data to clean and transform it.
    - Advanced algorithms and models (e.g.,LSTM, RandomForest) are applied to the data to predict future waste levels and optimize collection routes.

time	host	topic	value
2024-04-28 15:13:59.796648906+00:00	LAPTOP-8866L08J	LoRa/ab/cm	49
2024-04-28 15:14:07.679438800+00:00	LAPTOP-8866L08J	LoRa/ab/cm	49
2024-04-28 15:14:09.832700532+00:00	LAPTOP-8866L08J	LoRa/ab/cm	49
2024-04-28 15:14:09.832700561+00:00	LAPTOP-8866L08J	LoRa/ab/cm	49
2024-04-28 15:14:10.872860813+00:00	LAPTOP-8866L08J	LoRa/ab/cm	49
...	...	...	...
2024-05-02 09:40:27.947178318+00:00	LAPTOP-8866L08J	LoRa/ab/cm	14
2024-05-02 09:40:34.404146115+00:00	LAPTOP-8866L08J	LoRa/ab/cm	10
2024-05-02 09:40:34.438229501+00:00	LAPTOP-8866L08J	LoRa/ab/cm	10
2024-05-02 09:40:35.490694885+00:00	LAPTOP-8866L08J	LoRa/ab/cm	14
2024-05-02 09:40:35.490707844+00:00	LAPTOP-8866L08J	LoRa/ab/cm	14

Date	height	category
11-05-2024 00:00	79	empty
11-05-2024 00:30	77	empty
11-05-2024 01:00	76	empty
11-05-2024 01:30	76	empty
11-05-2024 02:00	74	empty
...	...	...
24-05-2024 21:30	20	full
24-05-2024 22:00	20	full
24-05-2024 22:30	20	full
24-05-2024 23:00	18	half
24-05-2024 23:30	15	half

	date&time	readings	percentage
0	2024-04-16 00:00	80	0.00
1	2024-04-16 00:30	80	0.00
2	2024-04-16 01:00	79	1.25
3	2024-04-16 01:30	80	0.00
4	2024-04-16 02:00	78	2.50
...	...	...	...
668	2024-04-29 22:00	53	33.75
669	2024-04-29 22:30	52	35.00
670	2024-04-29 23:00	50	37.50
671	2024-04-29 23:30	47	41.25
672	2024-04-30 00:00	45	43.75

7. IoT Analytics

In our smart waste management system, IoT analytics play a critical role in transforming raw sensor data into actionable insights. We utilized cloud-based platforms and advanced machine learning models to analyse and predict waste levels effectively.

Platforms Used:

- **Amazon SageMaker and Google Colab**: For training, deploying, and managing machine learning models.

Machine Learning Models:

- **Random Forest**: Used to predict future waste levels by understanding patterns and trends in waste level data.
- **LSTM (Long Short-Term Memory)**: Employed for time-series forecasting to predict future waste levels.

Implementation Details:

Data Preparation:

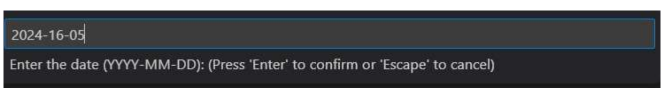
- **Data Collection**: Sensor data from waste bins (via ESP32 microcontrollers) was collected and transmitted using LoRa to edge devices and then to the cloud.
- **Data Aggregation**: Data was aggregated in a time-series database (InfluxDB) for efficient retrieval and analysis.
- **Preprocessing**: Data cleaning and formatting were performed to ensure quality inputs for the machine learning models.

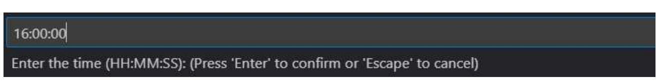
Model Training and Analysis:

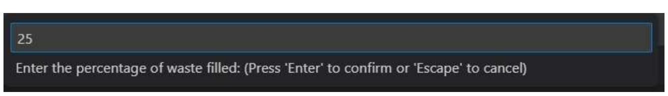
- **Amazon SageMaker**:
  - **Random Forest Model**: Trained to predict future waste levels based on historical data.
  - **Performance Analysis**: Model accuracy and performance metrics were evaluated to ensure reliability.
- **Google Colab**:
  - **LSTM Model**: Trained to predict future waste levels using the sequential nature of the time-series data.

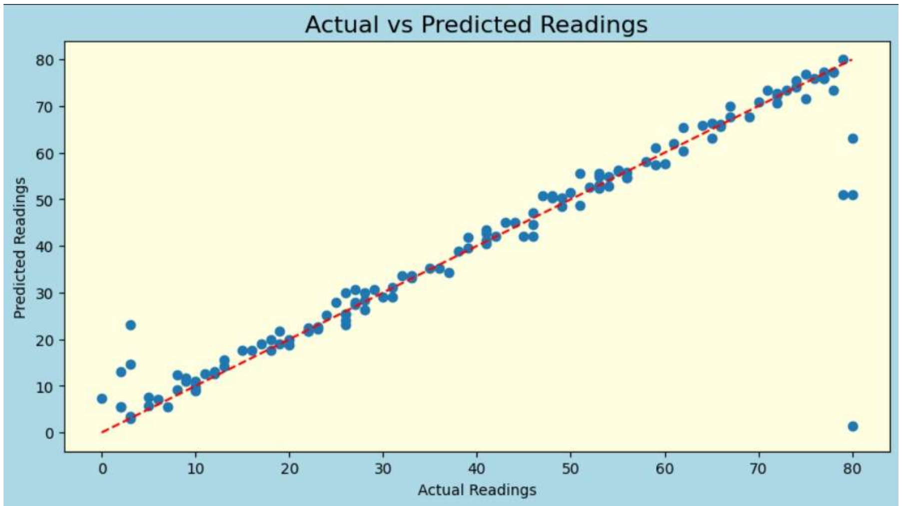
Prediction and Visualization:

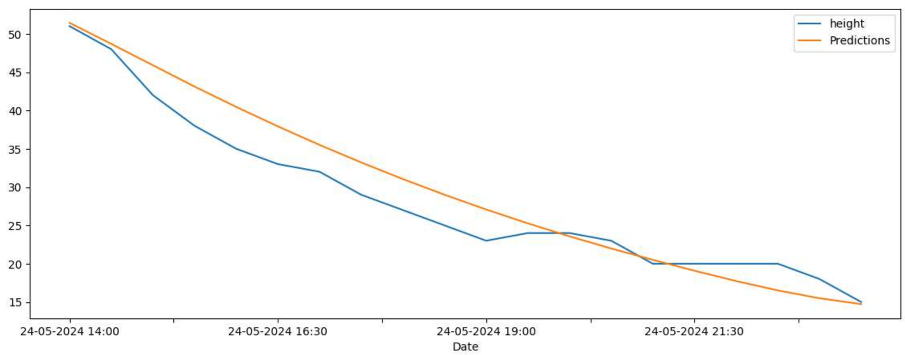
- **Prediction**: Both Random Forest and LSTM models were used to make predictions on new data.
- **Visualization**: Predicted data was visualized using tools like Matplotlib to provide insights into waste trends and future levels.









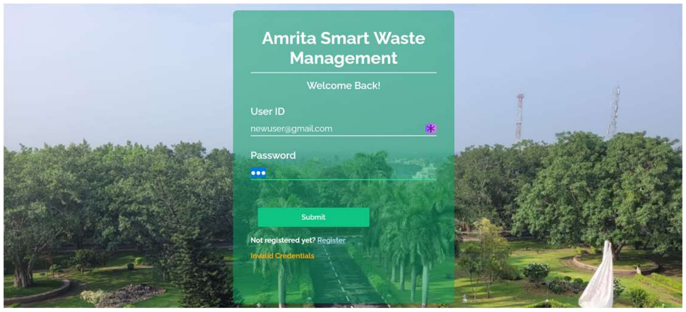


8. <u>Sample Test cases and Reports</u>

Test Case 1: Validation for User Registration

Objective: To verify that the user registration process functions correctly.

- **Expected Result:** The system should accept valid user data and store it in the database, while rejecting invalid data with appropriate error messages.
- **Actual Result:** Confirm that the system accepts valid user data and stores it in the database, and rejects invalid data with appropriate error messages.



Test Case 2: User Login Verification

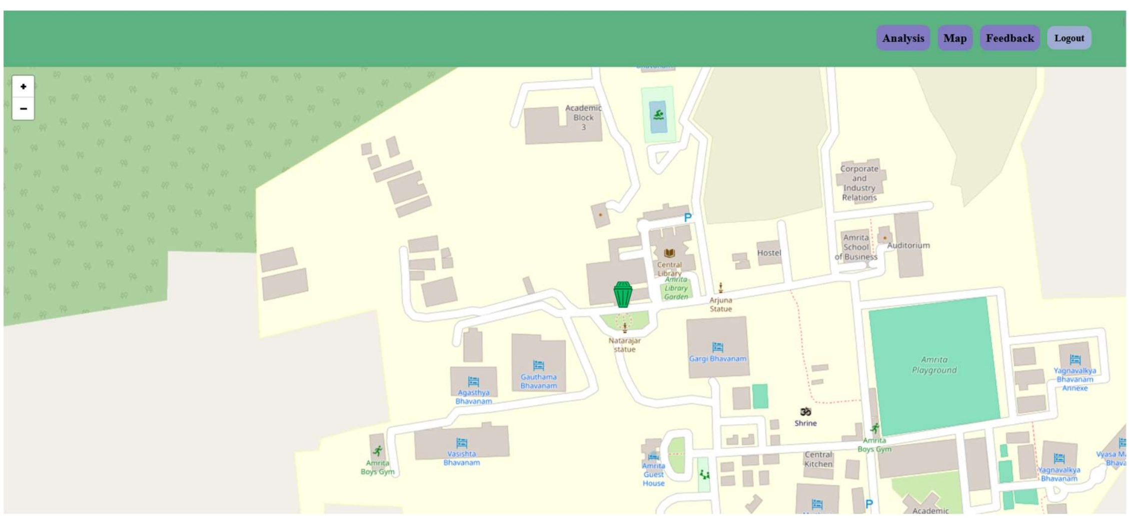
Objective: To verify that the login functionality works correctly.

- **Expected Result:** The system should authenticate users with correct credentials and allow access, while denying access to users with incorrect credentials.
- **Actual Result:** Confirm that the system authenticates users with correct credentials and allows access, and denies access to users with incorrect credentials.

Test Case 3: Check if the Bin is Empty

Objective: To verify that the system accurately detects and reports when a waste bin is empty.

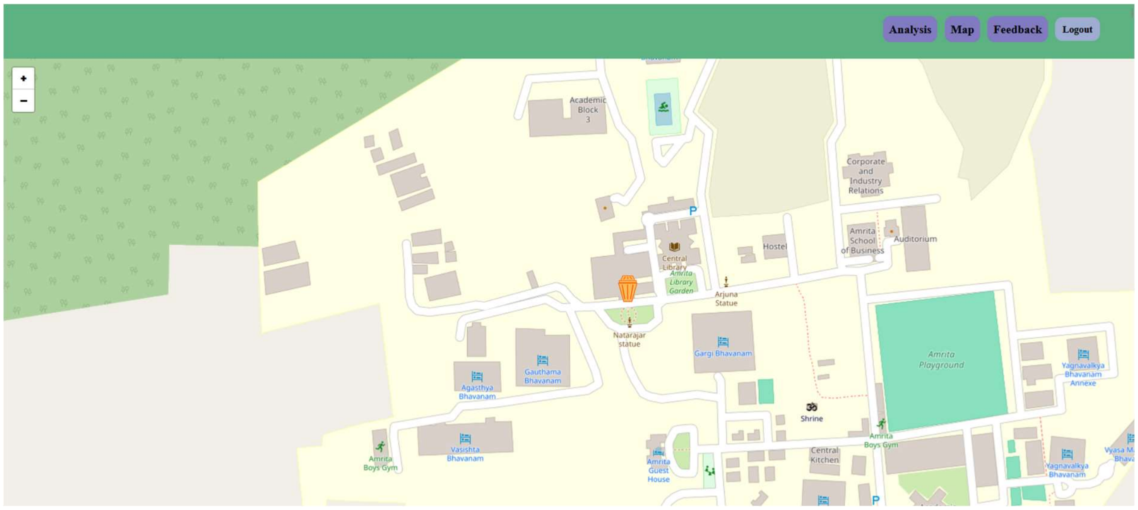
- **Expected Result:** The system should report the waste bin as empty when no waste is detected.
- **Actual Result:** Confirm that the system accurately reports the waste bin as empty when no waste is detected



Test Case 4: Check if the Dustbin is Partially Full

Objective: To verify that the system accurately detects and reports when a waste bin is partially full.

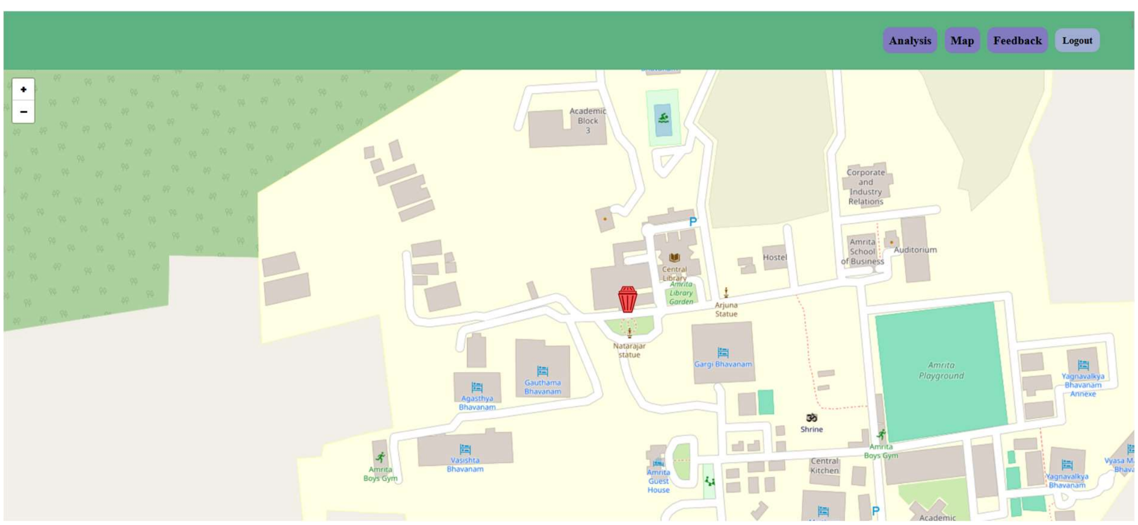
- **Expected Result:** The system should report the waste bin as partially full when it is not completely empty or full.
- **Actual Result:** Confirm that the system accurately reports the waste bin as partially full when it is not completely empty or full.



Test Case 5: Check if the Bin is Critically Full

Objective: To verify that the system accurately detects and reports when a waste bin is critically full.

- **Expected Result:** The system should report the waste bin as critically full when it reaches a predefined high level of waste.
- **Actual Result:** Confirm that the system accurately reports the waste bin as critically full when it reaches a predefined high level of waste.



9. Outcomes

1. **Enhanced Efficiency in Waste Collection**:
    - **Optimized Routes**: By predicting waste levels, collection routes were optimized, reducing unnecessary trips and ensuring timely collection.
    - **Reduced Operational Costs**: Efficient route planning and reduced fuel consumption led to significant cost savings.
2. **Real-Time Monitoring**:
    - **Continuous Data Collection**: The system provided real-time data on waste bin levels, enabling immediate response to fill levels and preventing overflows.
    - **Proactive Management**: With continuous monitoring, waste management teams could proactively address issues before they became critical.
3. **Environmental Sustainability**:
    - **Reduced Carbon Footprint**: Optimized collection routes and fewer trips reduced the carbon footprint of waste collection vehicles.
    - **Improved Waste Sorting**: Insights from data analytics helped improve waste sorting and recycling efforts, contributing to environmental sustainability.
4. **Data-Driven Insights**:
    - **Historical Data Analysis**: Analysing historical data helped understand waste generation patterns and trends, facilitating better planning and resource allocation.
    - **Forecasting and Planning**: The use of machine learning models like ARIMA, Random Forest and LSTM provided accurate forecasts of waste levels, aiding in better planning and decision-making.
5. **Scalability and Adaptability**:
    - **Scalable Solution**: The system was designed to be scalable, allowing easy integration of more waste bins and expansion to larger areas.
    - **Adaptable Framework**: The use of cloud-based platforms and IoT technology ensured that the system could be adapted and improved over time with new data and evolving needs.
6. **Community Benefits**:
    - **Cleaner Public Spaces**: Timely waste collection ensured cleaner and more hygienic public spaces.
    - **Increased Public Awareness**: The visibility of smart waste bins and their benefits raised public awareness about waste management and environmental responsibility.

Overall, the project demonstrated the potential of IoT and data analytics in revolutionizing traditional waste management methods, making them more efficient, cost-effective, and environmentally friendly.

10. Conclusion and Future Scope

In conclusion, the integration of end devices, edge devices, and cloud infrastructure in our smart waste management project has enabled a comprehensive and efficient solution for monitoring, analysing, and optimizing waste collection processes. The project successfully leverages IoT technology, real-time data processing, and cloud-based analytics to address the challenges of traditional waste management methods. By bringing together various components, including sensors, microcontrollers, communication modules, MQTT brokers, databases, analytics engines, and dashboard interfaces, we have created a cohesive system capable of delivering actionable insights and improving waste management practices.

The integration of end devices, represented by Waste Bin ESP32, ensures accurate and timely data collection from waste bins using ultrasonic sensor. These devices play a crucial role in gathering essential information about fill levels and transmitting it to the edge devices, or Gateway ESP32s, using LoRa communication. The edge devices receive, process, and transmit this data to the cloud infrastructure, where it is stored, analysed, and visualized in real-time. This seamless integration enables continuous monitoring of waste levels, efficient route planning, and proactive decision-making to optimize waste collection processes.

Looking ahead, there are several avenues for future development and enhancement of the smart waste management system. These include:

1. **Enhanced Sensor Capabilities:** Integrate advanced sensor technologies, such as image recognition and weight sensors, to provide more comprehensive and accurate data about waste composition and fill levels.

2. **Mobile Applications:** Develop mobile applications and engagement platforms to empower residents to report waste-related issues, participate in recycling initiatives, and contribute to community-wide waste management efforts.

3. **IoT Network Expansion:** Expand the IoT network to cover a wider geographic area and incorporate additional waste collection points, public spaces, and recycling facilities into the system.

4. **Energy Efficiency and Sustainability:** Integrate energy-efficient components, renewable energy sources, and sustainable materials into the system's hardware design to minimize environmental impact and reduce operational costs.

5. **Data Sharing and Collaboration:** Foster collaboration with local municipalities, waste management companies, and environmental organizations to share data, best practices, and resources for collective problem-solving and innovation.

6. **Regulatory Compliance and Standards:** Ensure compliance with regulatory requirements and industry standards related to waste management, data privacy, and environmental sustainability, and actively participate in the development of new regulations and guidelines.

Reference

- https://www.hivemq.com/blog/integrating-esp32-lorawan-hivemq-mqtt-broker-advanced-iot/
- https://airb.in/