COLLEGE CODE: 3114

COLLEGE NAME: MEENAKSHI COLLEGE OF

ENGINEERING

DEPARTMENT: INFORMATION TECCHNOLOGY

STUDENT NM-ID: autit1071

ROLL NO: 311423205071

DATE: 16-05-2025

TECHNOLOGY-PROJECT NAME:

WASTE MANAGEMENT SYSTEM AND OPTIMIZATION

SUBMITTED BY:

VIJAYALAKSHMI S

SREENITHI A

VARALAKSHMI P

VIJAYAMADHUVANDHINI I

SUSHMITHA B

IoT-Based Smart Waste Management and Optimization System

Abstract

The IoT-Based Smart Waste Management and Optimization System aims to revolutionize municipal waste collection by using IoT sensors, cloud data analytics, and route optimization algorithms. The project enables real-time monitoring of waste bin levels, automates pickup scheduling, and uses AI-powered analytics for optimizing collection routes and resource allocation. This solution is scalable, environmentally impactful, and reduces operational costs. The system includes smart bins, a web-based dashboard for municipal authorities, and a mobile alert system for sanitation workers.

1. Project Demonstration

Overview:

This system demonstration showcases the end-to-end working of the smart waste management platform. Realtime data is gathered from IoT-enabled bins and presented on a dashboard. Route optimization is demonstrated using a Google Maps API simulation for waste collection trucks.

Demonstration Details:

- System Walkthrough: A live demo showing how full bins trigger alerts, how route suggestions aregenerated, and how the dashboard is updated.
- Sensor Data: Ultrasonic sensors measure waste levels and send data via Wi-Fi-enabled microcontrollerslike ESP32 to the cloud.
- Optimization Algorithms: Based on bin fullness and truck capacity, optimized pickup routes are suggestedusing AI-driven algorithms.
- Real-time Dashboard: Displays bin status, collection routes, and system alerts for maintenance.
- Security: Data is encrypted using TLS, and access is secured with role-based authentication foradministrators and workers.

Outcome:

The demonstration proves the efficiency of the system in optimizing waste collection and improving urban hygiene with minimal manpower.

2. Project Documentation

Overview:

This documentation provides complete insight into system architecture, hardware-software interaction, AI algorithms for route optimization, and user/operator interfaces.

Documentation Sections:

- System Architecture: Block diagrams and data flow charts showing sensor-cloud-dashboardcommunication.
- Hardware Design: Ultrasonic sensor, ESP32, power module, and bin prototype integration.
- Code Documentation: Sensor reading script (Arduino IDE), cloud integration (Firebase/MQTT), anddashboard backend (Python/Node.js).
- User Guide: Instructions for operators and municipal administrators to view and act on alerts.
- Admin Guide: Adding/removing bins, route planning, data management, and system maintenance.
- Testing Reports: Results from bin fill simulation, Wi-Fi range testing, and route recalculations underchanging scenarios.

Outcome:

All aspects of the system are clearly documented for future use, scaling, or modification.

3. Feedback and Final Adjustments

Overview:

Feedback is gathered during the live demo from faculty, end-users (workers), and tech mentors. Based on suggestions, UX/UI, sensor calibration, and map rendering speed are refined.

Steps:

- Feedback Collection: Through Google Forms and live observation.
- Refinement: Improved bin alert accuracy, responsive mobile UI for workers, and reduced data latency.
- Final Testing: Verifying real-time updates, route changes, and data backup functionalities.

Outcome:

The system is made field-ready and user-friendly, ensuring adaptability in real-world urban settings.

4. Final Project Report Submission

Overview:

The report includes the full journey from ideation to demonstration, highlighting challenges like network reliability and sensor calibration, with proposed solutions.

Report Sections:

- Executive Summary: Summarizes goals, achievements, and societal impact.
- Phase Breakdown: Research, prototype development, IoT integration, route optimization, and dashboardcreation.
- Challenges & Solutions: Issues like bin lid detection, faulty Wi-Fi coverage, and solution via edge datacaching.
- Outcomes: A reliable, scalable system for smart waste monitoring.

Outcome:

A complete, documented report for submission, useful for municipalities and future developers.

5. Project Handover and Future Works

Overview:

The system is handed over with manuals and plans for scaling to entire smart cities.

IoT-Based Smart Waste Management and Optimization System

Handover Details:

- Next Steps: Integration with municipal ERP systems, multilingual dashboard, adding features like recyclablewaste sorting using AI image classification.
- Hardware Extension: Solar charging bins, LoRa-based long-range communication, and mobile appdevelopment for public reporting.

Outcome:

The project is deployment-ready and opens doors for sustainable city management innovations.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
const char* ssid = "Your_SSID";
const char* password = "Your_PASSWORD";
const String serverUrl = "http://your-server-ip/upload";
const int trigPin = D1;
const int echoPin = D2;
void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi!");
}
long getDistanceCM() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  long distance = duration * 0.034 / 2;
  return distance;
}
void loop() {
  long distance = getDistanceCM();
  int fill_level = map(distance, 0, 30, 100, 0);
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/json");
    String payload = "{\"bin_id\": \"BIN_1\", \"fill_level\": " + String(fill_level) + "}";
    int httpResponseCode = http.POST(payload);
    Serial.print("POST Response code: ");
    Serial.println(httpResponseCode);
    http.end();
  } else {
    Serial.println("WiFi not connected!");
  }
delay(10000); // Send every 10 seconds
}
```

Smart Waste Bin IoT Code and Output

1. Arduino Code

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

const char* ssid = "Your_SSID";
const char* password = "Your_PASSWORD";
const String serverUrl = "http://your-server-ip/upload";

const int trigPin = D1;
const int echoPin = D2;

void setup() {
  Serial.begin(115200);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  WiFi.begin(ssid, password);
  Serial.print("Connecting to WiFi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nConnected to WiFi!");
}

long getDistanceCM() {
  digitalWrite(trigPin, LOW); delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  long duration = pulseIn(echoPin, HIGH);
  long distance = duration * 0.034 / 2;
  return distance;
```

```
}

void loop() {
  long distance = getDistanceCM();
  int fill_level = map(distance, 0, 30, 100, 0);
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    http.begin(serverUrl);
    http.addHeader("Content-Type", "application/json");
     String payload = "{\"bin_id\": \"BIN_1\", \"fill_level\": \"" + String(fill_level) +
"\"}";
    int httpResponseCode = http.POST(payload);
    Serial.print("POST Response code: ");
    Serial.println(httpResponseCode);
    http.end();
  } else {
    Serial.println("WiFi not connected!");
  }
  delay(10000); // Send every 10 seconds
}
```

2. Simulated Serial Monitor Output

```
Connecting to WiFi..........
Connected to WiFi!
Measured distance: 8 cm
Mapped fill level: 73%
POST payload: {"bin_id":"BIN_1","fill_level":"73"}
POST Response code: 200

Measured distance: 20 cm
Mapped fill level: 33%
```

Smart Waste Bin IoT Code and Output

```
POST payload: {"bin_id":"BIN_1","fill_level":"33"}
POST Response code: 200

WiFi not connected!
```