International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

AI-Driven Smart Waste Collection with Dynamic Route Optimization and Predictive Analytics

Tanuku N V Santhoshi Bhavani*[1], CH. Laxmi Tirupathamma[2], Thirumalapudi Roop Rahul[3]

1.Assistant Professor, Department of CSE (AIML), Malla Reddy college of Engineering, Hyderabad, India
E mail: santhoshibhavani92@gmail.com.
2. Executive Assistant for CLO and Content Associate, Hyderabad ,India. E maill: chlakshmi2025@gmail.com.
3.Assistant Professor, Department of CSE (AIML), Malla Reddy college of Engineering, Hyderabad, India.
E mail: rooprahul99@gmail.com.

Abstract—Municipal waste collection is often organized around fixed schedules, even though waste accumulates at different rates across locations. This mismatch can lead to unnecessary truck visits to partially empty bins, overflowing receptacles in high-demand areas, inefficient use of fuel, and limited operational visibility. This article presents SmartWaste, an Internet of Things (IoT)-enabled waste monitoring and route-planning prototype described in the accompanying project report. The design combines an HC-SR04 ultrasonic distance sensor, an ESP32 microcontroller, Wi-Fi telemetry, a MySQL relational database normalized to Third Normal Form (3NF), a priority-classification module, a heuristic route planner based on the Traveling Salesperson Problem (TSP), and web dashboards for municipal supervisors and collection workers. Sensor readings are transmitted as structured JSON messages at ten-minute intervals. The system classifies bins using configurable fill thresholds and prioritizes bins requiring collection; a nearest-neighbor route is then improved using 2-opt exchanges. The report describes a 15-bin test bed and reports a reduction in route distance from 42.8 km to 29.4 km, a reduction in travel time from 4.8 hours to 2.9 hours, a 28.9% reduction in diesel consumption, and a decrease in weekly overflow complaints from 14 to 1. It also reports a 4.2 ms database update latency under its stated benchmark conditions. These figures are presented as results reported in the project documentation; the article does not independently reproduce the experiments. The work illustrates how low-cost sensing, relational data integrity, explainable prioritization, and dynamic routing can be integrated into a practical waste-collection workflow. Limitations include the small test bed, the need for longer field trials, and the distinction between geographic-distance heuristics and road-network travel costs.

Keywords—Smart waste management; Internet of Things; ESP32; ultrasonic sensing; MySQL; database normalization; route optimization; Traveling Salesperson Problem; municipal logistics

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

System Overview

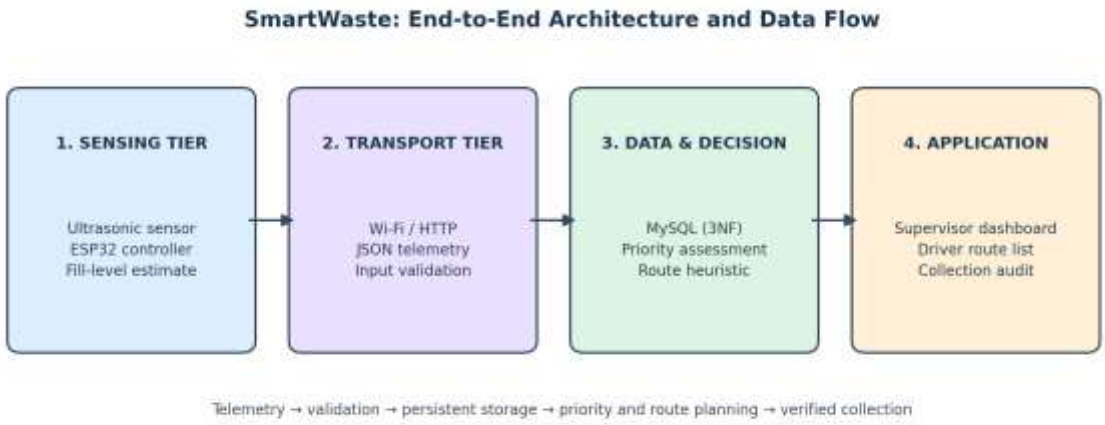


Figure 1. Four-tier SmartWaste architecture and telemetry flow.

1. Introduction

Urban growth and changing consumption patterns place increasing pressure on municipal solid-waste services. Collection systems must serve residential streets, markets, transport hubs, hospitals, campuses, and other locations whose waste-generation rates differ significantly. A fixed collection schedule is straightforward to administer, but it does not directly reflect the current state of each bin. A truck may visit a bin that is almost empty while another bin overflows before its next scheduled visit. These mismatches consume time and fuel and can create hygiene, environmental, and public-space problems[1][2].

Internet of Things (IoT) sensing offers a way to make collection decisions from current observations rather than schedules alone. A sensor-equipped bin can estimate the distance between its lid and the waste surface, while a microcontroller packages the reading and sends it to a server. When these readings are stored alongside bin locations, worker assignments, and collection events, the information can support prioritization and route planning. A web interface can then provide supervisors with an overview and drivers with an actionable stop list[3][4].

The project documented here, Smart Waste, integrates these elements into an end-to-end prototype. Its emphasis is not solely on a sensor or a routing algorithm: it connects sensing, telemetry, database design, decision logic, route planning, dashboard presentation, and collection verification. The project report identifies MySQL 8.0 as the central data store and uses a normalized relational design to maintain relationships among bins, sensor readings, workers, route plans, and collection history[5].

The main contribution is an integrated workflow for condition-based waste collection using relatively inexpensive hardware and open-source software components. The system uses an interpretable set of fill-level thresholds and a heuristic TSP-style route planner[6].

Its dashboard and collection log close the operational loop by allowing workers to confirm a completed stop and record the resulting state. The reported prototype outcomes are promising, but they should be interpreted in light of the limited 15-bin test bed and the evaluation details available in the project report[7].

2. Problem Statement and Objectives

The baseline problem is the inefficiency of collecting every bin in a fixed sequence without considering fill level. This creates three linked challenges: unnecessary visits to bins that do not require service, delayed collection at bins that fill quickly, and limited real-time information for supervisors. A practical solution must therefore detect bin state, communicate observations reliably, prioritize service needs, and convert the selected bins into a route that a driver can follow[8][9].

The source project specifies five engineering objectives. First, construct a low-cost sensing node using an HC-SR04 ultrasonic sensor and ESP32. Second, maintain a MySQL database in Third Normal Form, with linked records and query performance targets. Third, classify bin urgency through explicit thresholds and a priority score. Fourth, plan a shorter route across bins that require service using a TSP-inspired heuristic. Fifth, provide a responsive dashboard for live monitoring, dispatch, configuration, and collection-history review[10][11].

The system uses three operational states: Normal for fill levels below 65%, Warning for levels from 65% to below 80%, and Urgent for levels at or above 80%, as described across the project report. The exact threshold boundary should be implemented consistently in software and documentation. A richer Urgency Priority Index (UPI) is also proposed, combining current fill level, recent filling speed, time since last collection, and location importance. The use of explicit factors makes the prioritization easier to explain to municipal operators than a black-box prediction model[12].

3. Related Work and Research Gap

The project report reviews prior approaches involving ultrasonic sensors, cloud-connected bins, SMS alerts, geospatial dashboards, waste-fill prediction, wireless sensor networks, RFID, low-power wide-area networking, and vehicle-routing heuristics. Collectively, these approaches show that sensing and connectivity can improve visibility, while routing methods can reduce unnecessary travel. However, a system that solves only one part of the problem may leave operational gaps: a map may display full

bins without recommending a collection sequence; a sensor prototype may lack a durable data model; and a route algorithm may be evaluated only in simulation rather than integrated with live collection records[13][14].

The report identifies eight gaps relevant to the proposed design: disconnected hardware and software; weak database engineering; single-threshold decisions that ignore filling rate; lack of practical route guidance; limited worker-facing interfaces; high deployment or cloud costs; noisy readings caused by irregular waste surfaces; and lack of closed-loop verification after collection. Smart Waste addresses these concerns through a single pipeline, median filtering of sensor readings, a normalized relational schema, configurable thresholds, map-based monitoring, route dispatch, and collection confirmation[15].

The article uses the literature descriptions and bibliographic entries provided in the project documentation. Those references have not been independently checked for bibliographic accuracy in preparing this article. For formal journal submission, each reference and DOI should be verified against the publisher or scholarly index, and claims attributed to individual studies should be checked against the original papers16][17].

Table 1. Summary of approaches discussed in the project report

Approach described	Potential contribution	Gap noted in the source report
Ultrasonic sensing with local storage	Low-cost bin-level measurement	Limited integration with route planning and durable multi-user data storage
Cloud-connected monitoring	Remote telemetry and dashboard visibility	Recurring service costs and dependence on internet access
SMS threshold alerts	Simple notification when a bin is full	No unified map or operational route sequence
GIS / map visualization	Spatial visibility of bin state	Map display alone does not optimize collection
Fill-level prediction	Anticipation of future waste accumulation	Needs adequate historical data and deployment validation
Routing heuristics	Potential reduction in travel distance	Must be connected to live bin status and road-network costs
SmartWaste approach	Sensing, relational storage, routing, and collection audit in one workflow	Needs larger and independently reproducible field evaluation

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

4. System Architecture

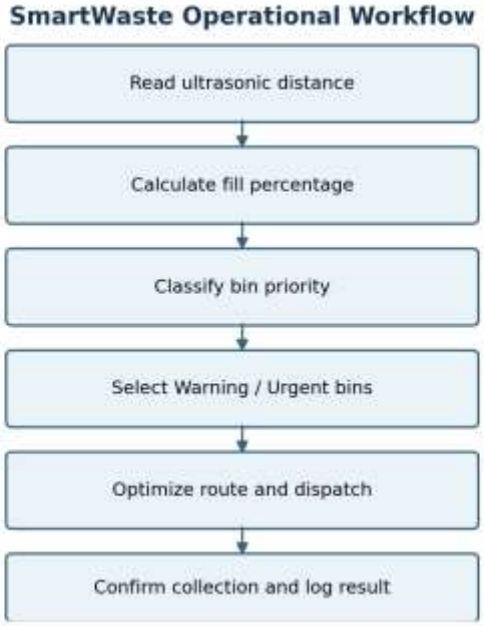


Figure 2. Operational workflow from sensing to verified collection.

Smart Waste is organized as four connected tiers. The sensing tier consists of the physical bin, ultrasonic sensor, ESP32 microcontroller, and battery. The transport tier uses Wi-Fi and HTTP/REST-style communication to transmit a JSON payload. The processing tier receives and validates telemetry, stores records in MySQL, evaluates urgency, and generates route plans. The presentation tier provides a supervisor dashboard and a mobile-friendly driver interface[18].

A typical telemetry record contains a bin identifier, measured distance, calculated fill percentage, battery voltage, and timestamp. The ESP32 is described as waking approximately every ten minutes, taking multiple distance readings, selecting a median value to reduce the effect of outliers, sending the data, and returning to a low-power state. The source report describes a weather-resistant enclosure and rechargeable 18650 cells, with a logic-level shifter used to protect the ESP32 input from the HC-SR04 output voltage.

The backend validates incoming readings before database insertion. The current bin record is updated, while the historical sensor reading is retained for trend analysis. The decision engine filters bins that need collection and passes their locations to the route planner. The resulting ordered stops are stored as a route plan and presented to the driver. When collection is confirmed, the system records the event and updates the bin's current state. This separation of sensing, storage, decision-making, and presentation allows each module to be tested independently.[19]

5. Hardware and Software Design

The hardware node uses the HC-SR04 ultrasonic sensor to estimate the air gap between the sensor mounted beneath the lid and the top of the waste. The sensor measures echo travel time; distance is estimated from the speed of sound and round-trip duration. If the bin has internal height H, measured air gap d, and an effective sensor margin m, the project uses a fill estimate of the general form: Fill (%) = 100 × (H − d)/(H − m), clipped to the interval 0–100%. Calibration is necessary because bin dimensions, sensor placement, and irregular waste surfaces affect the reading.

To reduce measurement noise, the firmware collects multiple readings and uses the median rather than relying on a single echo. The source report stresses central mounting and a protective collar to reduce false echoes from bin walls. The report also describes a sleep/wake cycle intended to reduce energy consumption. These are sensible design choices, but battery-life claims should be verified under the actual sensor power draw, Wi-Fi connection time, battery capacity, weather, and sampling interval.

The software stack listed in the project report includes C++ firmware using an Arduino/ESP-IDF environment, a Python Flask backend, MySQL 8.0 with InnoDB, HTML/CSS/JavaScript for the browser interface, and Leaflet with OpenStreetMap for map display. These technologies provide a relatively accessible development path and reduce reliance on proprietary mapping services[20].

In a production deployment, API authentication, encrypted transport, device identity, rate limiting, access control, and secure secret storage would also be required.

6. Database Design and Data Integrity

The database is a core part of the design because the platform must preserve both the latest state of a bin and a history of observations and collections. The source report identifies five principal entities: BINS, SENSOR_DATA, WORKERS, COLLECTION_HISTORY, and ROUTE_PLANS. BINS stores stable attributes such as asset code, location, capacity, fill level, status, and battery health. SENSOR_DATA stores time-stamped measurements and links each reading to a bin. WORKERS stores staff and shift information. COLLECTION_HISTORY records collection events, including the bin, worker, vehicle, time, before-and-after fill values, and collected weight where available. ROUTE_PLANS stores generated route details and their assigned operational context.

The schema is intended to follow Third Normal Form (3NF). Stable bin information is stored once rather than repeated in every sensor record. Measurements are held in a separate one-to-many table, while worker information is separated from collection events. Primary keys uniquely identify rows, and foreign keys preserve valid relationships. Indexes on fields frequently used for filtering—such as

status and timestamp—can improve dashboard and history-query performance. The report describes benchmarking with 100,000 synthetic sensor records and states that indexing substantially reduced query times.

Collection confirmation should be implemented as a transaction. The system inserts a collection-history record and updates the current bin state together. If either action fails, the transaction should roll back so the database does not show a bin as empty without an audit record, or show an audit record without the corresponding state update. Database constraints, validation rules, and transaction handling are therefore not merely implementation details; they are central to operational accountability.

7. Priority Assessment and Route Optimization

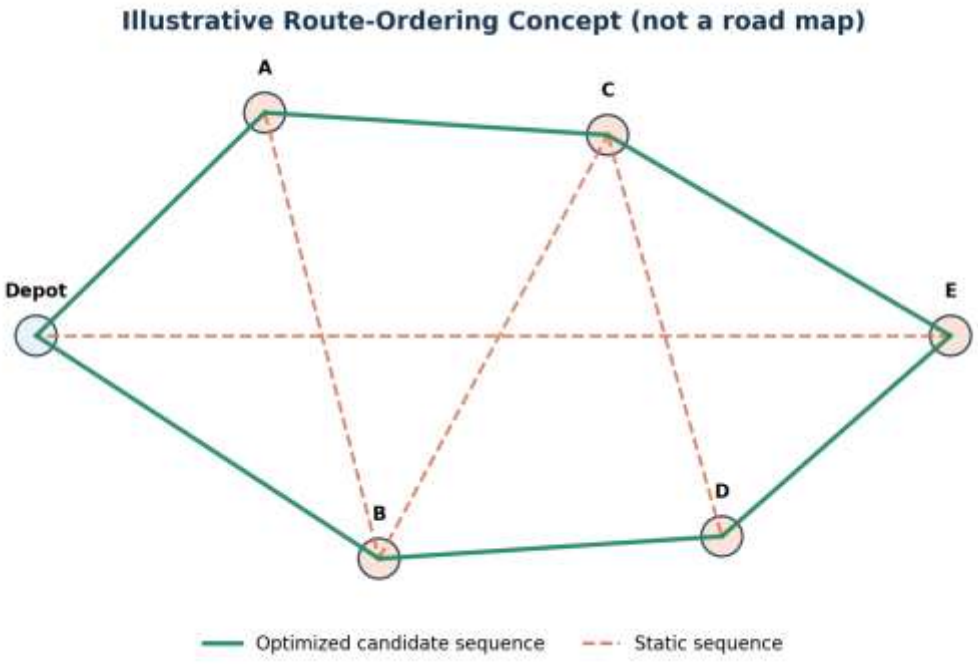


Figure 3. Illustrative comparison of static and optimized route ordering; schematic only.

The system filters bins based on service need before planning the route. In the threshold-based version, bins below 65% are normally excluded, bins between 65% and 80% are marked Warning, and bins at or above 80% are marked Urgent. The report additionally defines a proposed Urgency Priority Index using four factors: current fill level (55% weight), filling speed over the previous three hours (20%), time since last collection (15%), and location importance (10%). Location multipliers are described for hospitals, markets, and parks. For reproducibility, an implementation should

normalize each factor to a common scale before applying weights and document how missing historical readings are handled.

Route ordering is modeled as a Traveling Salesperson Problem (TSP)-style tour. Exact TSP optimization becomes computationally expensive as the number of stops grows, so the project uses a heuristic approach. First, it selects bins requiring service. Next, a nearest-neighbor procedure starts at the depot and repeatedly selects the nearest unvisited candidate. Finally, a 2-opt improvement step tests whether reversing portions of the route reduces total distance. The resulting sequence is saved and displayed to the driver.

The source describes Haversine distance for calculating distances between geographic coordinates. Haversine distance estimates straight-line distance over the Earth's surface; it is not the same as road-network distance or actual driving time. Therefore, the reported route-distance improvements should be treated as prototype-level comparisons unless the route engine uses a street graph or routing service to account for one-way roads, turn restrictions, traffic, and road connectivity. A practical next version should use road-network travel costs while retaining the fast heuristic for stop ordering.

Table 2. Priority factors specified in the project report

Factor	Weight in reported UPI	Operational interpretation
Current fill level	55%	Reflects immediate bin capacity pressure
Filling speed over previous 3 hours	20%	Captures how quickly the bin is approaching overflow
Time since last collection	15%	Accounts for time elapsed since servicing
Location importance	10%	Raises priority for designated critical locations

8. Implementation and Experimental Method

The project report describes a 15-bin test bed and a web application with bin monitoring, route planning, worker dispatch, collection history, settings, and event-stream views. The operational flow begins with sensor measurements, proceeds through wireless transmission and database storage, and ends with a recommended collection route. During collection, the driver confirms each serviced bin, and the platform records the action in the audit history.

The report also describes a database benchmark based on 100,000 synthetic sensor readings and 1,000 collection records, together with 10 unit tests and 5 integration tests. It reports that all 15 tests passed. For a repeatable research evaluation, the test protocol should state the hardware configuration, server specifications, network conditions, number of repeated runs, route-distance calculation method, fuel-measurement method, sensor ground-truth procedure, and confidence intervals or variation across trials. Those details are not fully specified in the material summarized here, so the numerical results below are reported as documented rather than independently validated.

The baseline is described as a static route that visits all 15 bins, whereas the SmartWaste route visits the seven bins identified as needing collection. A fair comparison requires the same depot, comparable operating conditions, equivalent vehicle capacity assumptions, and a consistent distance measurement method. Fuel consumption and overflow complaints should ideally be collected over multiple shifts and weeks, with clear definitions of a complaint and a collection event.

9. Results and Discussion

According to the project report, the static route covers 42.8 km and requires 4.8 hours, while the dynamic route covers 29.4 km and requires 2.9 hours. The absolute distance reduction is 13.4 km per shift. Relative to the 42.8 km baseline, this is a reduction of approximately 31.3%. The reported travel-time reduction is 1.9 hours, or approximately 39.6% relative to the 4.8-hour baseline. The report attributes these improvements to skipping eight bins that were below the selected collection thresholds and visiting the seven bins requiring service.

The report states that diesel consumption decreased by 28.9%, corresponding to 4.1 litres saved per shift. It also reports a decrease in overflow complaints from 14 per week to 1 per week, equivalent to a 92.9% reduction, and an increase in collection productivity from 3.1 to 5.2 bins per hour.

These values suggest that condition-based collection may improve the use of fleet time while helping focus service on locations with greater need. However, complaint counts can be affected by reporting behavior, seasonality, and local waste-generation patterns; they should be evaluated over a longer period before generalizing to city-wide operations.

For data processing, the source reports a 4.2 ms database update time and an end-to-end response-time breakdown totaling approximately 160.2 ms, including sensor ping, median filtering, Wi-Fi message transmission, and database-trigger processing. These measurements refer to different stages and should not be conflated: database update latency is not the same as end-to-end sensor-to-dashboard latency. The report further claims that indexing improved query performance by more than 96% on the stated synthetic dataset. Publishing the raw timings, query definitions, index definitions, and benchmark environment would allow independent replication.

The project estimates hardware component cost at approximately INR 1,780 per bin and calculates fuel savings using a diesel price of INR 95 per litre. At 4.1 litres saved per shift, the arithmetic fuel saving is INR 389.50 per shift. The report estimates payback within about 4.8 months for the stated 15-bin setup. This estimate depends on the number of operating days, the number of trucks and bins equipped, installation and maintenance costs, communication costs, battery replacement, and whether

the reported fuel reduction persists in field operation. It should therefore be presented as a preliminary estimate, not a guaranteed financial outcome.

Table 3. Reported comparison between static and dynamic collection

Measure	Static route / baseline	SmartWaste route / result	Reported change
Route distance	42.8 km	29.4 km	13.4 km less; 31.3% reduction
Travel time	4.8 hours	2.9 hours	1.9 hours less; 39.6% reduction
Diesel consumption	Baseline not separately stated	4.1 L saved per shift	28.9% reduction reported
Overflow complaints	14 per week	1 per week	92.9% reduction
Collection productivity	3.1 bins/hour	5.2 bins/hour	Increase reported by project
Database update latency	—	4.2 ms	Reported under stated benchmark conditions
Sensor-to-server response	—	160.2 ms total	Reported stage-wise latency total

10. Limitations and Future Work

The main limitation is scale: a 15-bin test bed is useful for demonstrating integration but cannot represent all variation in a large municipality. Bin shape, waste composition, sensor mounting, weather, network coverage, traffic, and route restrictions can affect performance. The report provides headline results but limited information on repetitions, uncertainty, and external validation. The system also depends on Wi-Fi, which may not be available at every roadside location. Sensor readings can be distorted by irregular waste surfaces, condensation, dirt, or a poorly aligned transducer.

A second limitation concerns routing. A nearest-neighbor plus 2-opt heuristic can generate useful tours quickly, but the best geometric route is not necessarily the best road route. The next implementation should use a road-network distance matrix and, where possible, time-dependent travel costs. Vehicle capacity, waste type, shift length, depot return requirements, and priority service windows can also be added to the optimization model.

The project report proposes LoRaWAN for longer-range, low-power communication, solar-assisted power, a dedicated driver mobile application, and automatic waste compaction as future directions. Additional research should include longer pilot deployments, ground-truth calibration of sensor readings, security testing, device authentication, encrypted telemetry, role-based access, fault recovery during network outages, and a formal cost-benefit assessment. Predictive models may later estimate fill rates, but they should be introduced only when enough reliable historical data is available and their benefit can be compared against the transparent threshold-based baseline.

11. Conclusion

This article presented Smart Waste, an integrated IoT-based approach to condition-driven municipal waste collection and route planning. The design combines ultrasonic sensing and ESP32 telemetry with a normalized MySQL database, interpretable urgency assessment, a TSP-inspired route heuristic, a browser dashboard, and transactional collection records. The project report describes a 15-bin prototype and reports reductions in distance, travel time, diesel use, and overflow complaints compared with a static collection route. The reported outcomes support the feasibility of connecting bin-level sensing with operational decisions in a single workflow.

The strongest contribution is the integration of data acquisition, reliable persistence, route generation, and collection verification rather than reliance on a sensor alert alone. Nevertheless, the reported performance figures require independent reproduction under clearly specified field conditions.

Future work should scale the pilot, use road-network routing, validate sensor accuracy over time, and strengthen security and reliability. With those improvements, SmartWaste could provide a practical foundation for more responsive and resource-efficient municipal waste collection.

References

[1] E. F. Codd, “A relational model of data for large shared data banks,” Communications of the ACM, vol. 13, no. 6, pp. 377–387, 1970, doi: 10.1145/362384.362685.

[2] D. Folianto, Y. S. Low, and W. L. Yeow, “SmartBin: Smart waste management system using wireless sensor mesh networks,” in Proc. IEEE World Forum on Internet of Things (WF-IoT), 2015, pp. 471–476.

[3] M. A. Hannan, M. Arebey, R. A. Begum, and H. Basri, “An automated waste collection system using RFID and intelligent optical sensors,” Waste Management, vol. 32, no. 12, pp. 2229–2238, 2018.

[4] T. Anagnostopoulos et al., “Challenges and solutions of surveillance web services for dynamic waste management,” Journal of Ambient Intelligence and Humanized Computing, vol. 8, no. 5, pp. 783–794, 2017.

[5] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Boston, MA, USA: Pearson, 2016.

[6] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY, USA: McGraw-Hill, 2020.

[7] C. J. Date, An Introduction to Database Systems, 8th ed. Boston, MA, USA: Addison-Wesley, 2004.

[8] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, The Traveling Salesman Problem: A Computational Study. Princeton, NJ, USA: Princeton University Press, 2006.

[9] G. Laporte, “The traveling salesman problem: An overview of exact and approximation algorithms,” European Journal of Operational Research, vol. 59, no. 2, pp. 231–247, 1992.

[10] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," Operations Research, vol. 21, no. 2, pp. 498–516, 1973.

[11] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," Journal of the ACM, vol. 7, no. 4, pp. 326–329, 1960.

[12] Espressif Systems, "ESP32 Series Datasheet and Hardware Technical Reference Manual," version 3.9, 2023.

[13] Oracle Corporation, "MySQL 8.0 Reference Manual: InnoDB Storage Engine Architecture and Optimization," 2024.

[14] Leaflet, "An open-source JavaScript library for mobile-friendly interactive maps," 2024. [Online]. Available: https://leafletjs.com/.

[15] OpenStreetMap Foundation, "OpenStreetMap." [Online]. Available: https://www.openstreetmap.org/.

[16] World Bank, What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Washington, DC, USA: World Bank, 2018.

[17] United Nations Environment Programme, Global Waste Management Outlook. Nairobi, Kenya: UNEP, 2021.

[18] M. M. Rahman, M. A. Hoque, and S. M. R. Islam, "IoT-based waste management system using ultrasonic sensors and local storage," IEEE Access, vol. 8, 2020. [Bibliographic details supplied in the project report; verify before submission.]

[19] S. Mahajan, P. Kokane, and A. D. Gawande, "Cloud-integrated smart waste management architecture utilizing Amazon Web Services," International Journal of Distributed Sensor Networks, vol. 17, no. 4, 2021. [Bibliographic details supplied in the project report; verify before submission.]

[20] K. Pardini et al., "IoT-based healthcare and solid waste monitoring architecture with geospatial mapping," IEEE Internet of Things Journal, vol. 7, no. 4, 2020. [Bibliographic details supplied in the project report; verify before submission.]