Mini project report on

University Fest Management System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE22CS351A – DBMS Project

Submitted by:

Viona Sequeira	**PES2UG22CS665**
Vijayalaxmi S.H	**PES2UG22CS663**

Under the guidance of

Dr. Geetha Dayalan

Associate Professor

PES University

AUG - DEC 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

University Fest Management System

is a bonafide work carried out by

Viona Sequeira	**PES2UG22CS665**
Vijayalaxmi S.H	**PES2UG22CS663**

In partial fulfilment for the completion of fifth semester DBMS Project (UE22CS351A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2024 – DEC. 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature
Dr. Geetha Dayalan
Associate Professor

DECLARATION

We hereby declare that the DBMS Project entitled **University Fest Management System** has been carried out by us under the guidance of **Prof. Geetha Dayalan, Associate Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2024.

Viona Sequeira	**PES2UG22CS665**
Vijayalaxmi S.H	**PES2UG22CS663**

ABSTRACT

The **Municipal Waste Management System (MWMS)** is a web based platform designed to revolutionize waste collection in urban areas. With a user friendly interface, MWMS encourages individuals to effortlessly request waste collection services tailored to specific waste types, including household, organic, and recyclable materials. By streamlining the request process, the platform promotes responsible waste management and encourages environmental sustainability. Addressing inefficiencies in traditional waste collection systems, MWMS contributes to cleaner cities and healthier communities, fostering modern approach to urban waste management.

TABLE OF CONTENTS

Introduction

Effective waste management is a critical component of maintaining clean, healthy, and sustainable urban environments. With increasing population density and the rising volume of waste generated in cities, traditional waste collection methods often struggle to keep up with the demand. Common challenges such as delayed collections and improper waste segregation lead to environmental pollution and resource wastage. Addressing these issues requires a modern, technology-driven approach to improve efficiency and promote eco-friendly practices.

The Municipal Waste Management System (MWMS) is a digital solution developed to meet these challenges head-on. Designed for ease of use and efficiency, the MWMS empowers individuals to coordinate waste collection more effectively. By leveraging real-time data processing, the platform reduces delays and improves overall resource utilization.

In addition to managing immediate collection needs, the system emphasizes the importance of proper waste segregation, encouraging users to sort their waste into categories such as household, organic etc. This not only simplifies collection processes but also supports recycling efforts and reduces landfill dependency.

The WMS goes beyond operational improvements by providing valuable insights through reporting and analytics. These features enable waste management organizations to monitor service performance, identify trends, and make data-driven decisions to enhance their operations. By integrating these advanced capabilities into a single platform, the Waste Management System represents a significant step toward smarter and more sustainable waste management in urban settings.

PROBLEM DEFINITIONS

With rapid urbanization, municipalities face significant challenges in efficiently managing waste collection and disposal. Traditional waste management systems often suffer from delayed collection times, improper waste segregation, and inefficient resource allocation, which contribute to environmental pollution and increased reliance on landfills. These issues underline the need for a digital solution to streamline waste collection processes, encourage better waste segregation, and provide analytics for data-driven decision-making.

The Municipal Waste Management System (MWMS) aims to address these issues by providing a platform that enables users to request waste collection, ensures waste is correctly segregated by type, and allows for real-time tracking of collection activities. Through data analytics and reporting, the system helps waste management authorities optimize collection schedules, monitor performance, and encourage sustainable waste practices.

USER REQUIREMENT SPECIFICATIONS

1. User Account Creation
    - Users should be able to create account specifying their name and address and phone number.

2. User and Admin Authentication
    - Users and admins must have secure login and authentication to access the MWMS. This should include role-based access controls to ensure that only authorized personnel can access specific functionalities.

3. Waste Collection Requests
    - Users can create requests for waste collection, specifying their area and type of waste (e.g., organic, recyclable, etc.).
    - The system should support tracking of requests and users should know when a vehicle has been assigned for requests

4. Waste Segregation Categories
    - The system should prompt users to categorize their waste, supporting types such as household, organic, recyclable, and hazardous.

   - Proper categorization should be enforced to facilitate efficient waste processing.

5. Admin Dashboard for Monitoring and Control
   - Admins should have access to a dashboard that displays all active collection requests, vehicle statuses, and waste types collected.
   - The dashboard should allow admins to assign vehicles based on request locations and waste types.
   - Admin should also be able to view all users under their centre

6. Vehicle and Route Management
   - Admins can assign specific vehicles to waste collection tasks, taking into account the area and waste type.

7. Analytics and Reporting
   - The system should provide reports on the volume and types of waste collected by area.
   - Analytical tools should be available to assess trends and make data-driven decisions for improving vehicle assignments and recycling efforts.

8. User-Friendly Interface
   - The platform should have a user-friendly design to ensure easy navigation for users and admins.
   - Users should be able to very easily view whether a vehicle has been assigned for their request or not.

9. Notification and Alerts System
   - Admin should receive a notification whenever user makes a request in the admin dashboard.
   - Only the admin of the centre that the user belongs to should get the notification.

LIST OF SOFTWARES/TOOLS/PROGRAMMING LANGUAGES USED

1. Programming Languages
    - Python: For backend development, API integration, and data processing.
    - JavaScript: For frontend development, user interactions, and dynamic web content.
    - SQL: For managing and querying the relational database.

2. Frameworks and Libraries
    - Flask: For building the backend and API endpoints.
    - React: For creating an interactive and user-friendly frontend interface.
    - Flowbite: For designing responsive and consistent UI components.

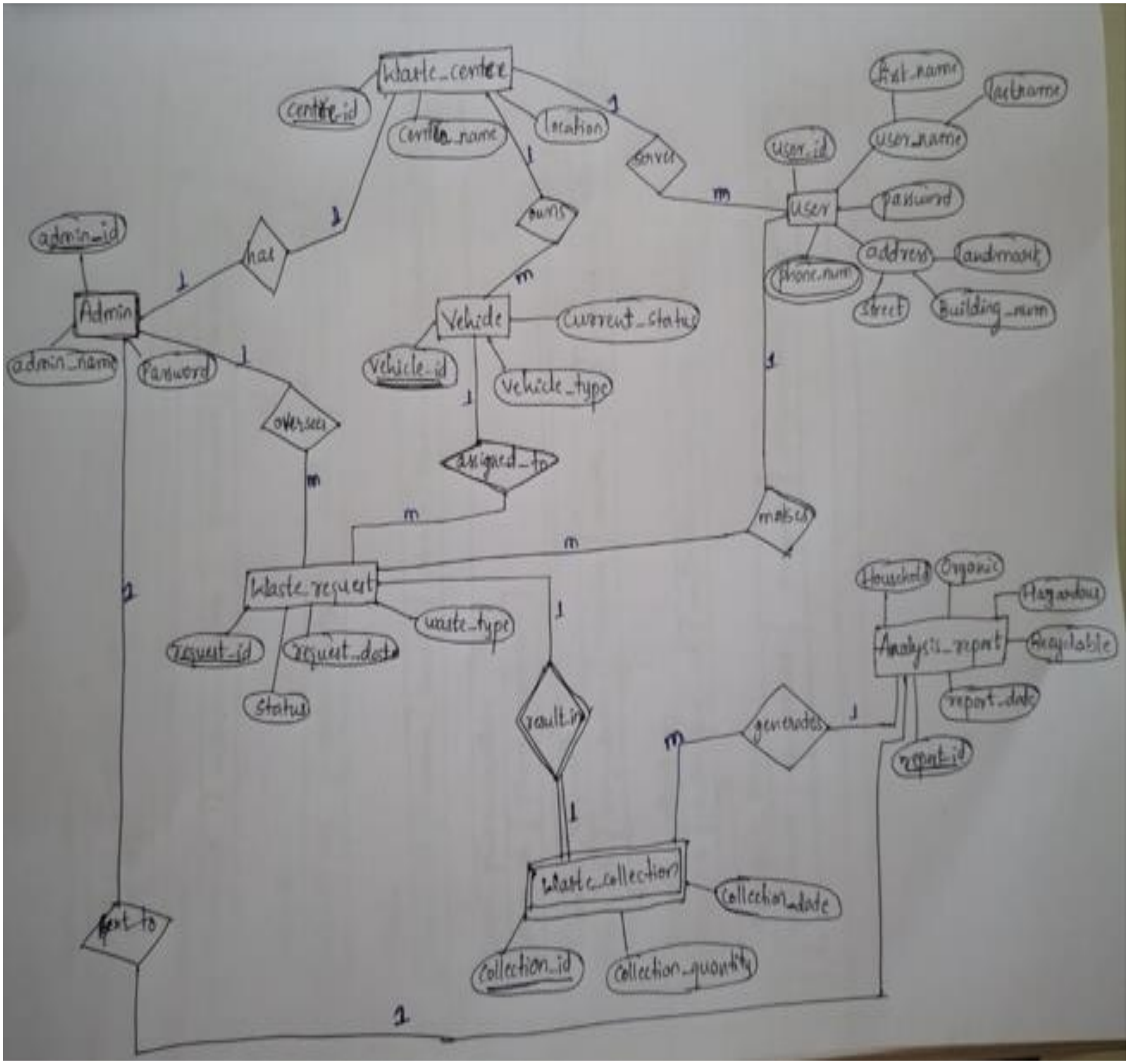
3. Database Management System
    - MySQL: For storing user data, admin data, waste requests, waste collections, vehicle assignments, and waste management data in a relational database.

4. Project Management and Planning Tools
    - Gantt Chart: For visualizing project schedules and dependencies.

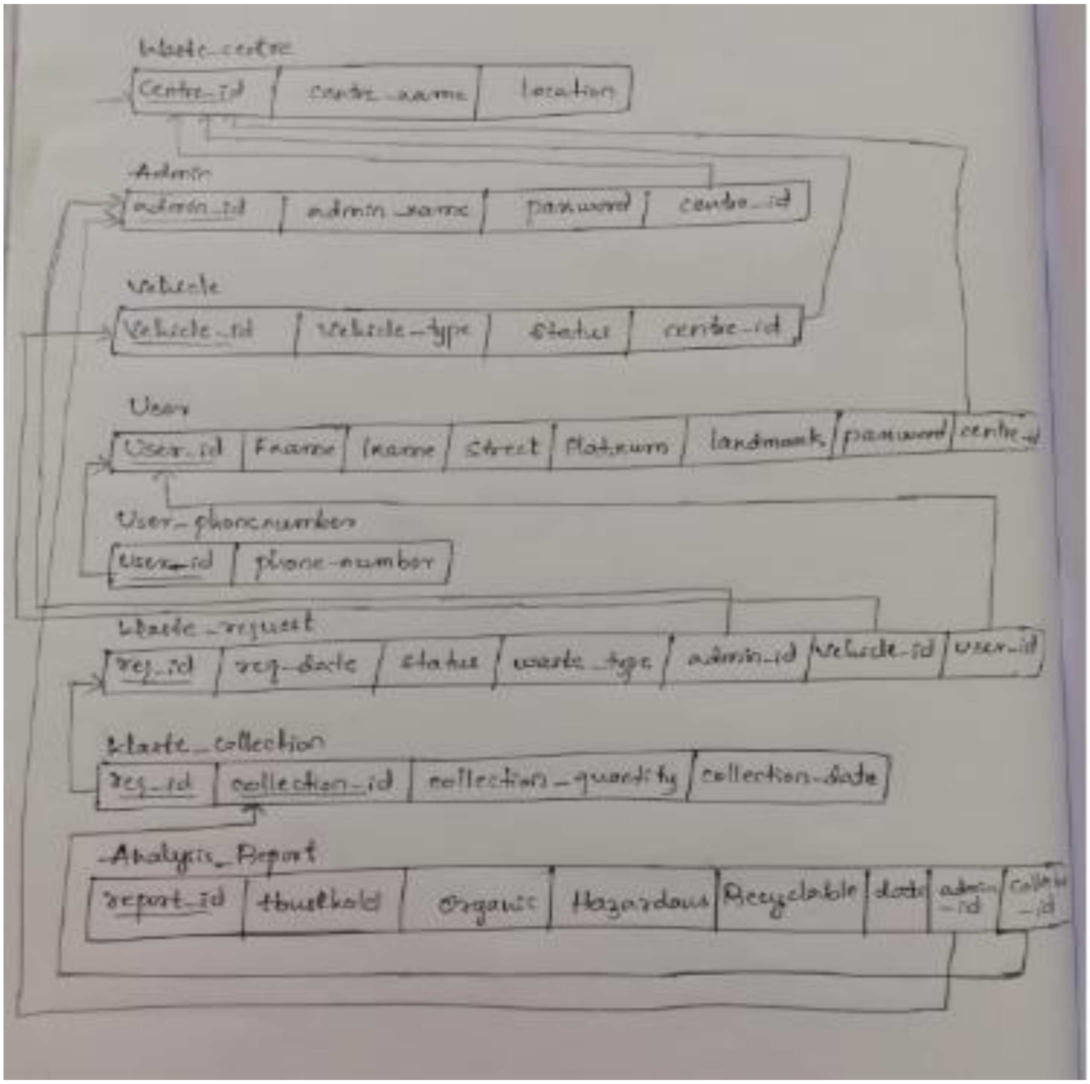
5. Version Control and Collaboration Tools
    - Git: For tracking code changes and version control.
    - GitHub: For hosting the repository, collaboration, and code reviews.

6. Bug Tracking and Testing Tools
    - Postman: For testing APIs and ensuring proper request-response handling in the backend.
7. Design and Prototyping Tools
    - Figma: For designing the user interface and creating interactive prototypes for user feedback.

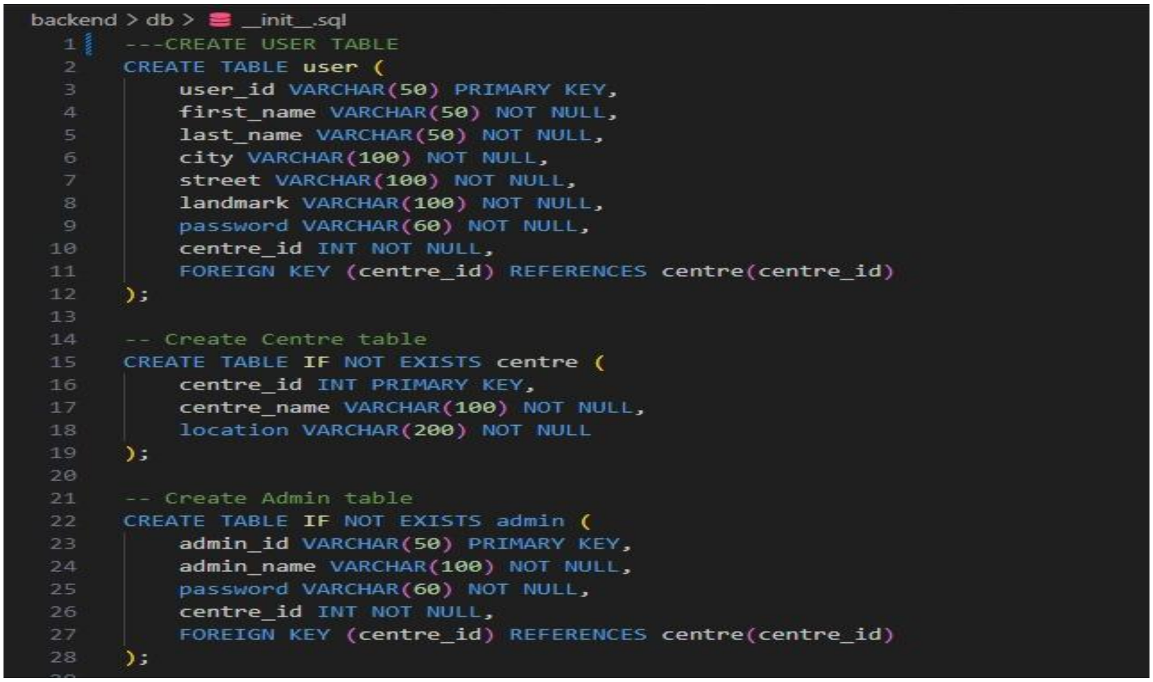
ER MODEL


Waste_centre
centre_id
centre_name
location
1
serves
m
has
1
1
runs
1
m
User
user_id
user_name
first_name
lastname
password
phone_num
address
landmark
street
Building_num
Admin
admin_id
admin_name
Password
1
oversees
m
Vehicle
Current_status
Vehicle_id
vehicle_type
1
assigned_to
m
1
makes
m
Waste_request
request_id
request_date
Status
waste_type
1
result_in
1
Waste_collection
Collection_date
Collection_id
Collection_quantity
m
generates
1
Analysis_report
Household
Organic
Hazardous
Recyclable
report_date
report_id
1
sent to
1

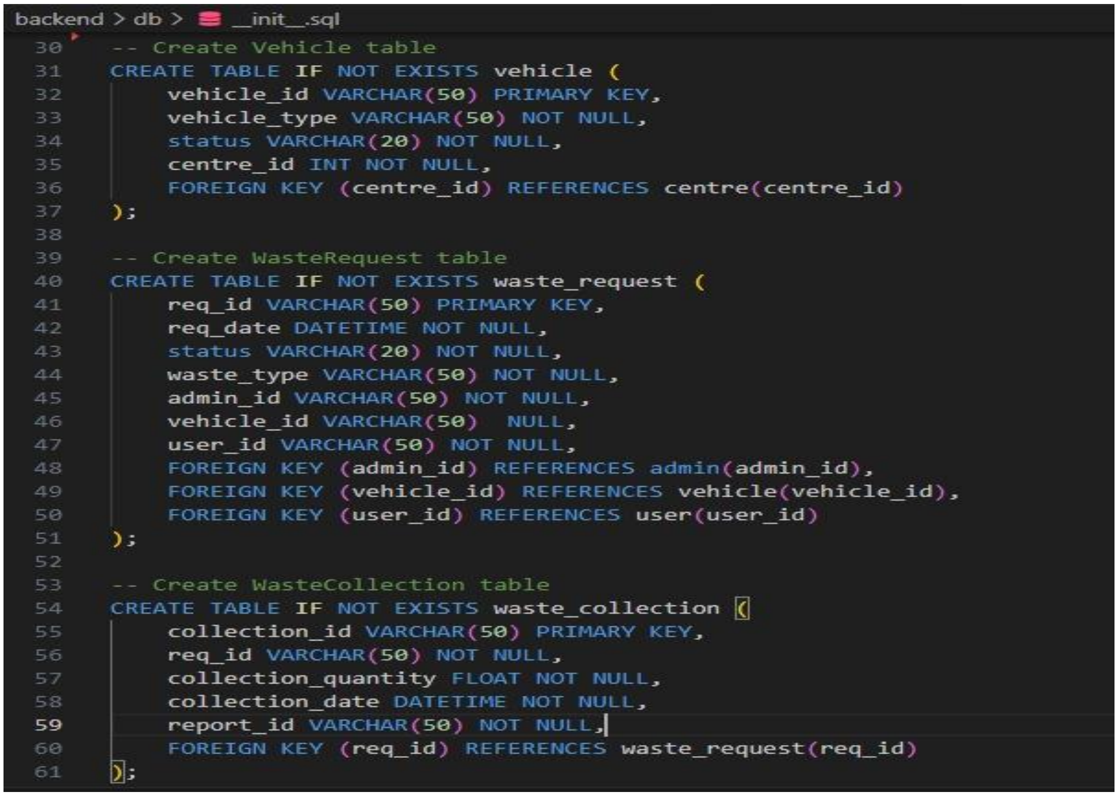
ER TO RELATIONAL MAPPING



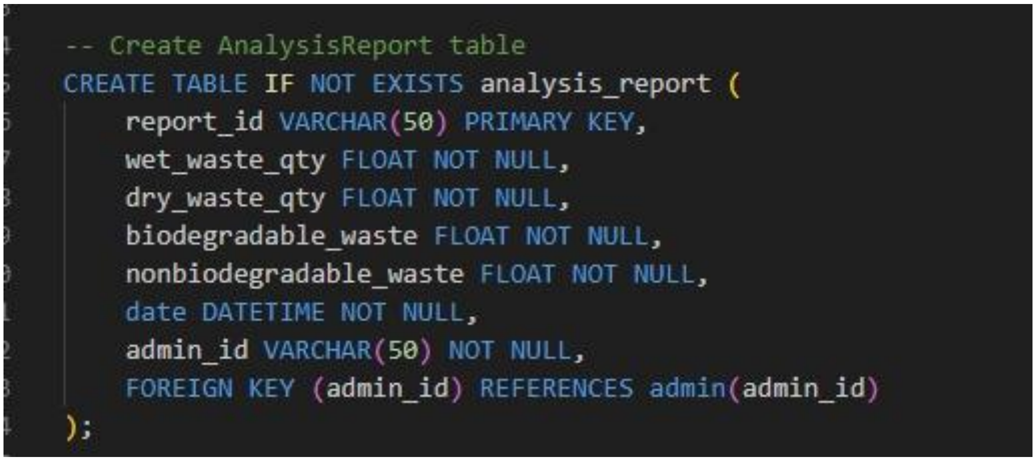
DDL STATEMENTS



```sql
---CREATE USER TABLE
CREATE TABLE user (
    user_id VARCHAR(50) PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    city VARCHAR(100) NOT NULL,
    street VARCHAR(100) NOT NULL,
    landmark VARCHAR(100) NOT NULL,
    password VARCHAR(60) NOT NULL,
    centre_id INT NOT NULL,
    FOREIGN KEY (centre_id) REFERENCES centre(centre_id)
);

-- Create Centre table
CREATE TABLE IF NOT EXISTS centre (
    centre_id INT PRIMARY KEY,
    centre_name VARCHAR(100) NOT NULL,
    location VARCHAR(200) NOT NULL
);

-- Create Admin table
CREATE TABLE IF NOT EXISTS admin (
    admin_id VARCHAR(50) PRIMARY KEY,
    admin_name VARCHAR(100) NOT NULL,
    password VARCHAR(60) NOT NULL,
    centre_id INT NOT NULL,
    FOREIGN KEY (centre_id) REFERENCES centre(centre_id)
);
```

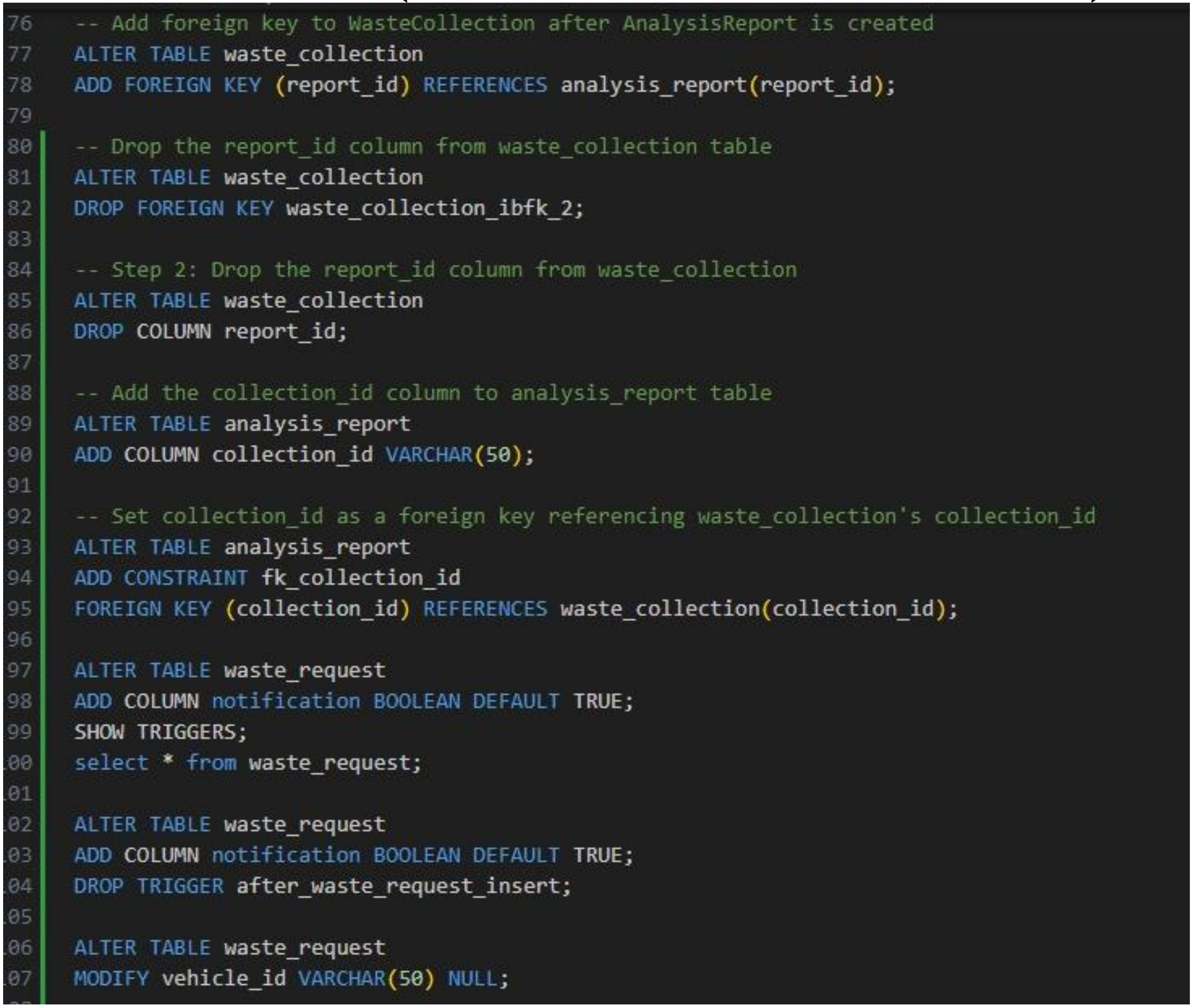


```sql
-- Create Vehicle table
CREATE TABLE IF NOT EXISTS vehicle (
    vehicle_id VARCHAR(50) PRIMARY KEY,
    vehicle_type VARCHAR(50) NOT NULL,
    status VARCHAR(20) NOT NULL,
    centre_id INT NOT NULL,
    FOREIGN KEY (centre_id) REFERENCES centre(centre_id)
);

-- Create WasteRequest table
CREATE TABLE IF NOT EXISTS waste_request (
    req_id VARCHAR(50) PRIMARY KEY,
    req_date DATETIME NOT NULL,
    status VARCHAR(20) NOT NULL,
    waste_type VARCHAR(50) NOT NULL,
    admin_id VARCHAR(50) NOT NULL,
    vehicle_id VARCHAR(50)  NULL,
    user_id VARCHAR(50) NOT NULL,
    FOREIGN KEY (admin_id) REFERENCES admin(admin_id),
    FOREIGN KEY (vehicle_id) REFERENCES vehicle(vehicle_id),
    FOREIGN KEY (user_id) REFERENCES user(user_id)
);

-- Create WasteCollection table
CREATE TABLE IF NOT EXISTS waste_collection (
    collection_id VARCHAR(50) PRIMARY KEY,
    req_id VARCHAR(50) NOT NULL,
    collection_quantity FLOAT NOT NULL,
    collection_date DATETIME NOT NULL,
    report_id VARCHAR(50) NOT NULL,
    FOREIGN KEY (req_id) REFERENCES waste_request(req_id)
);
```

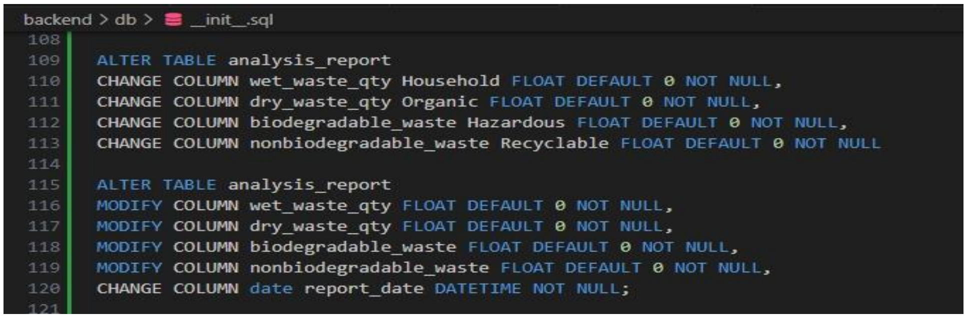


```sql
-- Create AnalysisReport table
CREATE TABLE IF NOT EXISTS analysis_report (
    report_id VARCHAR(50) PRIMARY KEY,
    wet_waste_qty FLOAT NOT NULL,
    dry_waste_qty FLOAT NOT NULL,
    biodegradable_waste FLOAT NOT NULL,
    nonbiodegradable_waste FLOAT NOT NULL,
    date DATETIME NOT NULL,
    admin_id VARCHAR(50) NOT NULL,
    FOREIGN KEY (admin_id) REFERENCES admin(admin_id)
);
```

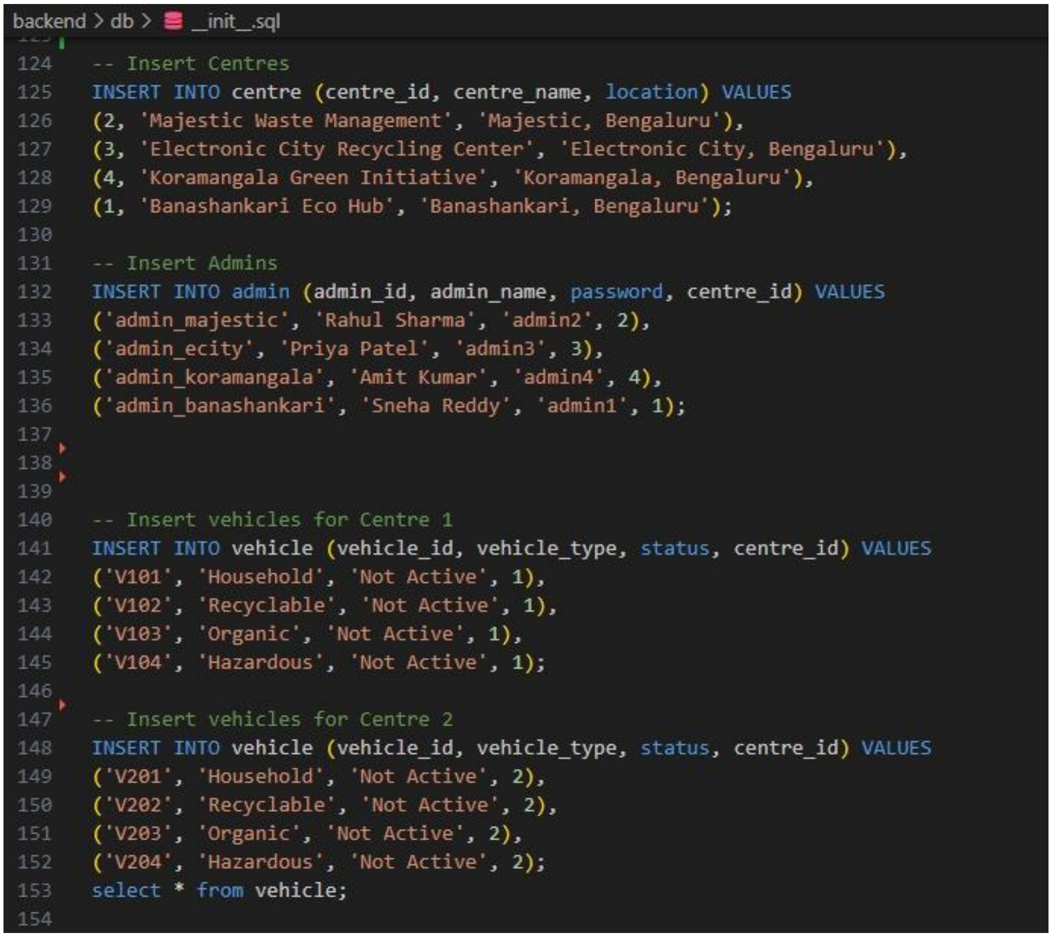
DML STATEMENTS (CRUD OPERATION SCREENSHOTS)



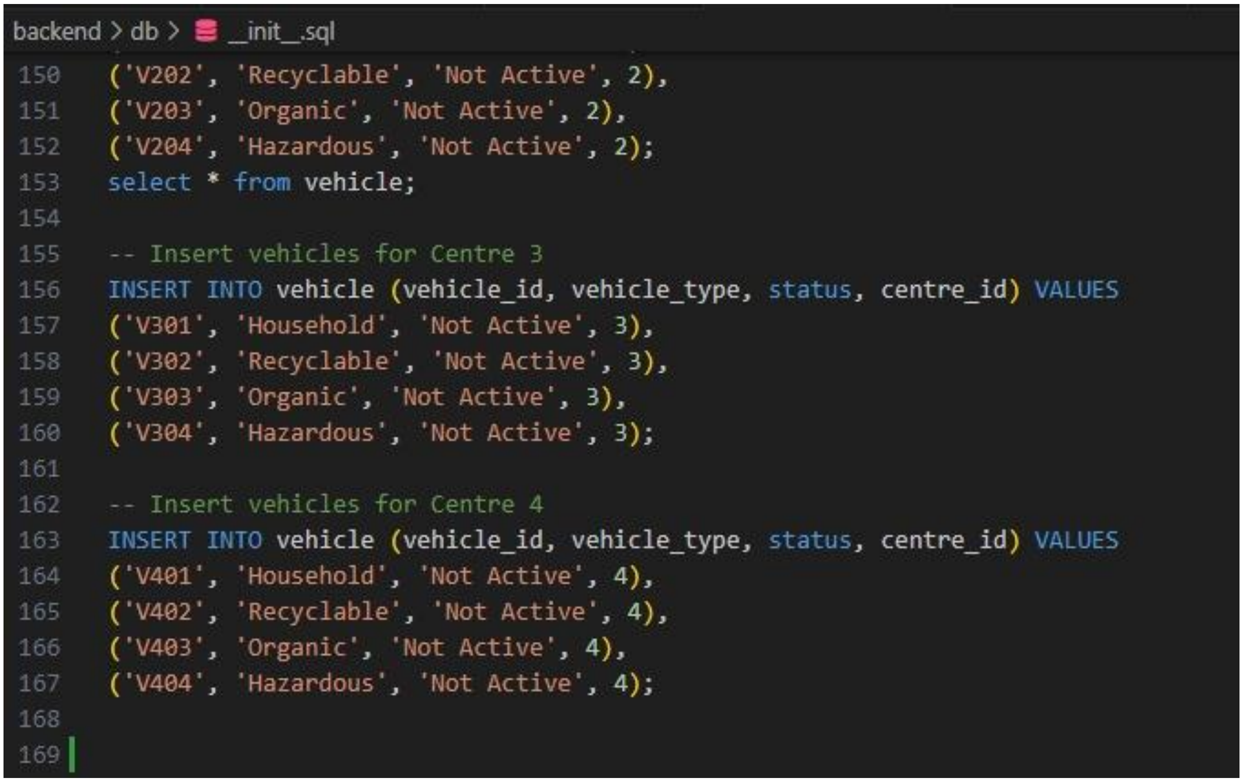
```sql
-- Add foreign key to WasteCollection after AnalysisReport is created
ALTER TABLE waste_collection
ADD FOREIGN KEY (report_id) REFERENCES analysis_report(report_id);

-- Drop the report_id column from waste_collection table
ALTER TABLE waste_collection
DROP FOREIGN KEY waste_collection_ibfk_2;

-- Step 2: Drop the report_id column from waste_collection
ALTER TABLE waste_collection
DROP COLUMN report_id;

-- Add the collection_id column to analysis_report table
ALTER TABLE analysis_report
ADD COLUMN collection_id VARCHAR(50);

-- Set collection_id as a foreign key referencing waste_collection's collection_id
ALTER TABLE analysis_report
ADD CONSTRAINT fk_collection_id
FOREIGN KEY (collection_id) REFERENCES waste_collection(collection_id);

ALTER TABLE waste_request
ADD COLUMN notification BOOLEAN DEFAULT TRUE;
SHOW TRIGGERS;
select * from waste_request;

ALTER TABLE waste_request
ADD COLUMN notification BOOLEAN DEFAULT TRUE;
DROP TRIGGER after_waste_request_insert;

ALTER TABLE waste_request
MODIFY vehicle_id VARCHAR(50) NULL;
```



```sql

ALTER TABLE analysis_report
CHANGE COLUMN wet_waste_qty Household FLOAT DEFAULT 0 NOT NULL,
CHANGE COLUMN dry_waste_qty Organic FLOAT DEFAULT 0 NOT NULL,
CHANGE COLUMN biodegradable_waste Hazardous FLOAT DEFAULT 0 NOT NULL,
CHANGE COLUMN nonbiodegradable_waste Recyclable FLOAT DEFAULT 0 NOT NULL

ALTER TABLE analysis_report
MODIFY COLUMN wet_waste_qty FLOAT DEFAULT 0 NOT NULL,
MODIFY COLUMN dry_waste_qty FLOAT DEFAULT 0 NOT NULL,
MODIFY COLUMN biodegradable_waste FLOAT DEFAULT 0 NOT NULL,
MODIFY COLUMN nonbiodegradable_waste FLOAT DEFAULT 0 NOT NULL,
CHANGE COLUMN date report_date DATETIME NOT NULL;
```



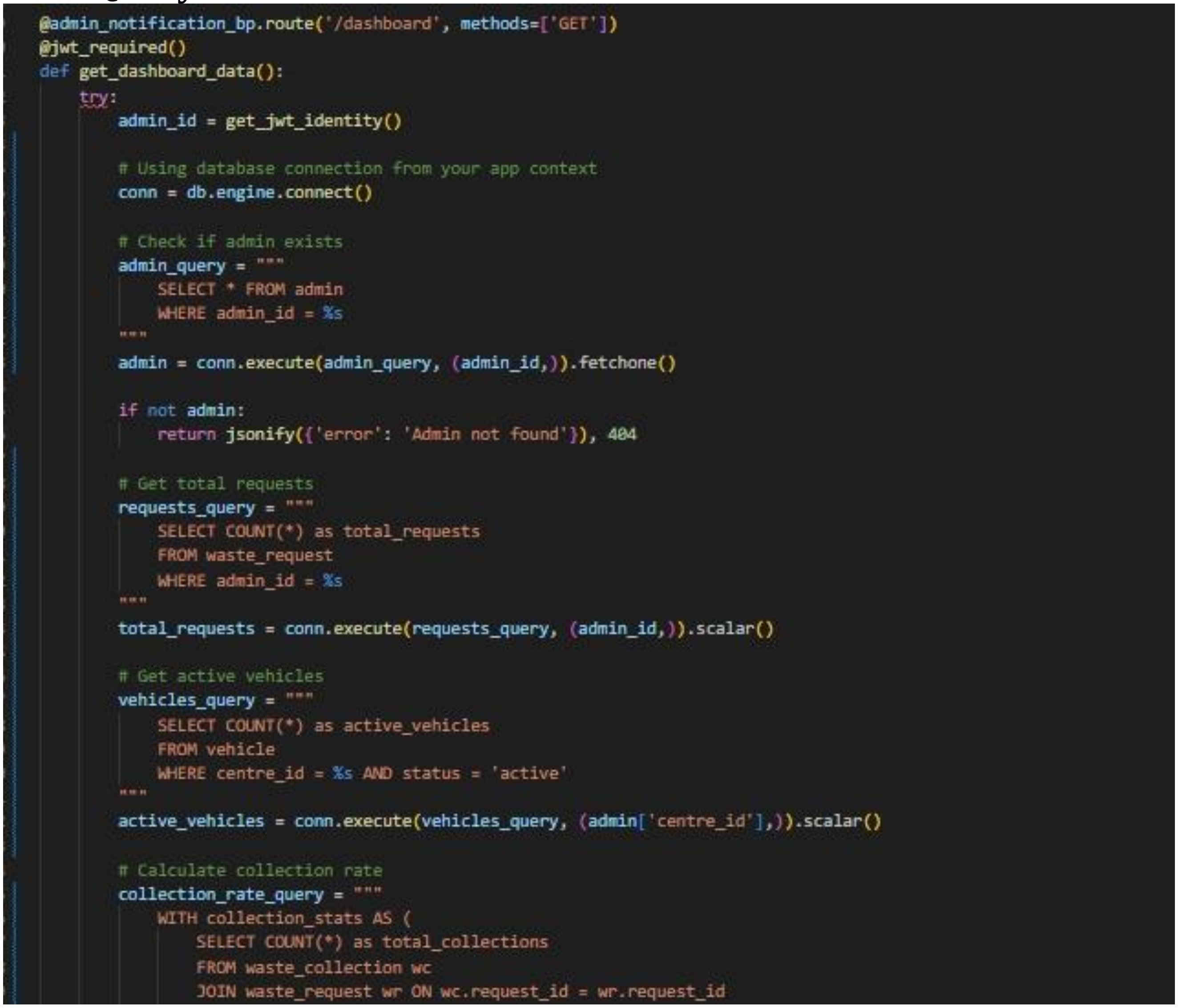
```sql
-- Insert Centres
INSERT INTO centre (centre_id, centre_name, location) VALUES
(2, 'Majestic Waste Management', 'Majestic, Bengaluru'),
(3, 'Electronic City Recycling Center', 'Electronic City, Bengaluru'),
(4, 'Koramangala Green Initiative', 'Koramangala, Bengaluru'),
(1, 'Banashankari Eco Hub', 'Banashankari, Bengaluru');

-- Insert Admins
INSERT INTO admin (admin_id, admin_name, password, centre_id) VALUES
('admin_majestic', 'Rahul Sharma', 'admin2', 2),
('admin_ecity', 'Priya Patel', 'admin3', 3),
('admin_koramangala', 'Amit Kumar', 'admin4', 4),
('admin_banashankari', 'Sneha Reddy', 'admin1', 1);



-- Insert vehicles for Centre 1
INSERT INTO vehicle (vehicle_id, vehicle_type, status, centre_id) VALUES
('V101', 'Household', 'Not Active', 1),
('V102', 'Recyclable', 'Not Active', 1),
('V103', 'Organic', 'Not Active', 1),
('V104', 'Hazardous', 'Not Active', 1);

-- Insert vehicles for Centre 2
INSERT INTO vehicle (vehicle_id, vehicle_type, status, centre_id) VALUES
('V201', 'Household', 'Not Active', 2),
('V202', 'Recyclable', 'Not Active', 2),
('V203', 'Organic', 'Not Active', 2),
('V204', 'Hazardous', 'Not Active', 2);
select * from vehicle;
```



```sql
('V202', 'Recyclable', 'Not Active', 2),
('V203', 'Organic', 'Not Active', 2),
('V204', 'Hazardous', 'Not Active', 2);
select * from vehicle;

-- Insert vehicles for Centre 3
INSERT INTO vehicle (vehicle_id, vehicle_type, status, centre_id) VALUES
('V301', 'Household', 'Not Active', 3),
('V302', 'Recyclable', 'Not Active', 3),
('V303', 'Organic', 'Not Active', 3),
('V304', 'Hazardous', 'Not Active', 3);

-- Insert vehicles for Centre 4
INSERT INTO vehicle (vehicle_id, vehicle_type, status, centre_id) VALUES
('V401', 'Household', 'Not Active', 4),
('V402', 'Recyclable', 'Not Active', 4),
('V403', 'Organic', 'Not Active', 4),
('V404', 'Hazardous', 'Not Active', 4);
```

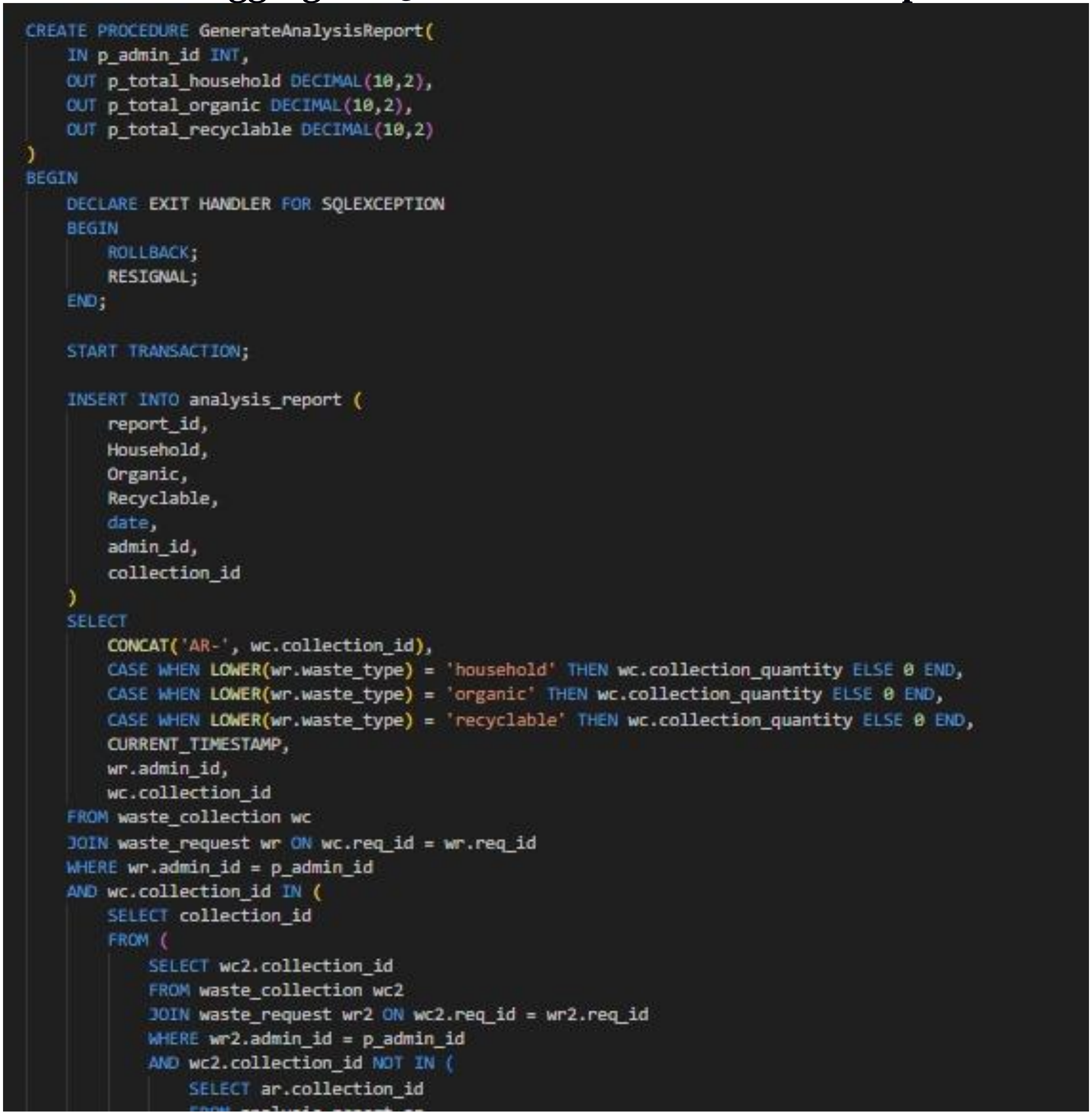
QUERIES (JOIN QUERY, AGGREGATE FUNCTION QUERIES AND NESTED QUERY)

Join Query



```
@admin_notification_bp.route('/dashboard', methods=['GET'])
@jwt_required()
def get_dashboard_data():
    try:
        admin_id = get_jwt_identity()

        # Using database connection from your app context
        conn = db.engine.connect()

        # Check if admin exists
        admin_query = """
            SELECT * FROM admin
            WHERE admin_id = %s
        """
        admin = conn.execute(admin_query, (admin_id,)).fetchone()

        if not admin:
            return jsonify({'error': 'Admin not found'}), 404

        # Get total requests
        requests_query = """
            SELECT COUNT(*) as total_requests
            FROM waste_request
            WHERE admin_id = %s
        """
        total_requests = conn.execute(requests_query, (admin_id,)).scalar()

        # Get active vehicles
        vehicles_query = """
            SELECT COUNT(*) as active_vehicles
            FROM vehicle
            WHERE centre_id = %s AND status = 'active'
        """
        active_vehicles = conn.execute(vehicles_query, (admin['centre_id'],)).scalar()

        # Calculate collection rate
        collection_rate_query = """
            WITH collection_stats AS (
                SELECT COUNT(*) as total_collections
                FROM waste_collection wc
                JOIN waste_request wr ON wc.request_id = wr.request_id
```

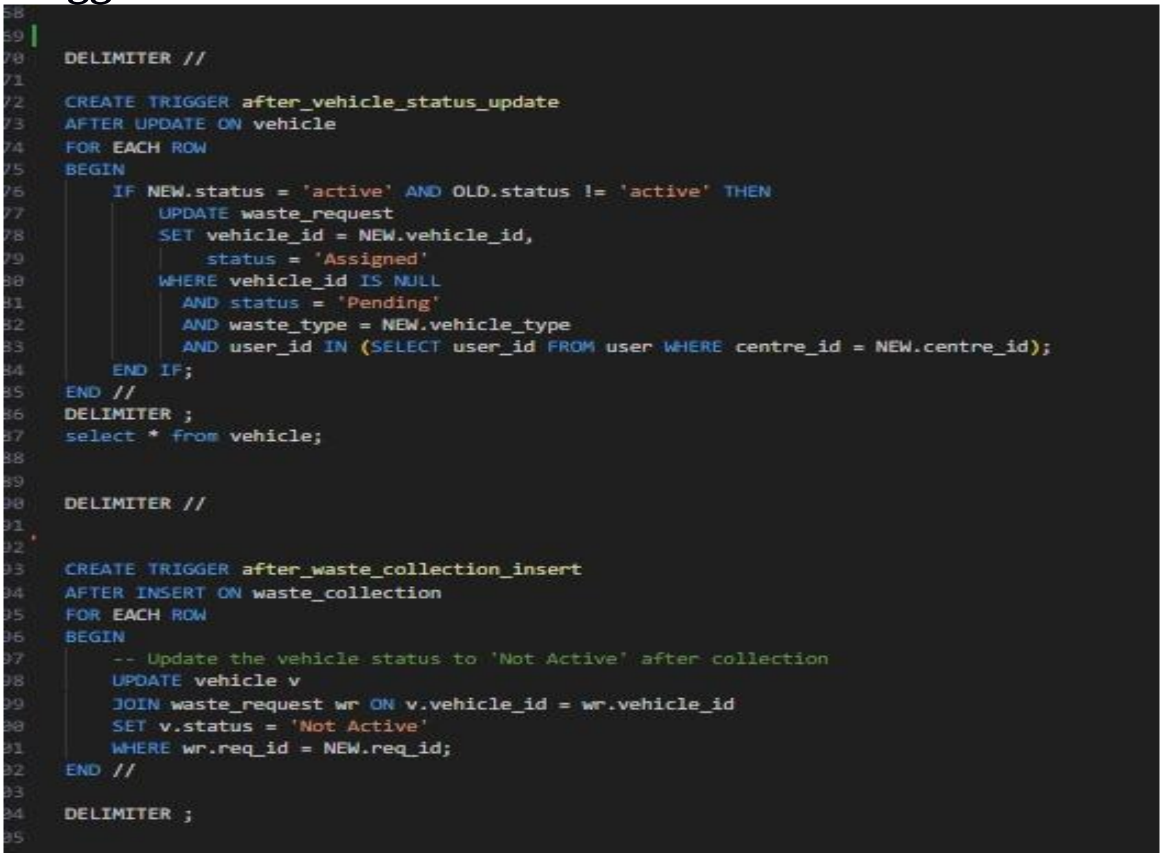
Nested and Aggregate Queries included in one stored procedure



```sql
CREATE PROCEDURE GenerateAnalysisReport(
    IN p_admin_id INT,
    OUT p_total_household DECIMAL(10,2),
    OUT p_total_organic DECIMAL(10,2),
    OUT p_total_recyclable DECIMAL(10,2)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    START TRANSACTION;

    INSERT INTO analysis_report (
        report_id,
        Household,
        Organic,
        Recyclable,
        date,
        admin_id,
        collection_id
    )
    SELECT
        CONCAT('AR-', wc.collection_id),
        CASE WHEN LOWER(wr.waste_type) = 'household' THEN wc.collection_quantity ELSE 0 END,
        CASE WHEN LOWER(wr.waste_type) = 'organic' THEN wc.collection_quantity ELSE 0 END,
        CASE WHEN LOWER(wr.waste_type) = 'recyclable' THEN wc.collection_quantity ELSE 0 END,
        CURRENT_TIMESTAMP,
        wr.admin_id,
        wc.collection_id
    FROM waste_collection wc
    JOIN waste_request wr ON wc.req_id = wr.req_id
    WHERE wr.admin_id = p_admin_id
    AND wc.collection_id IN (
        SELECT collection_id
        FROM (
            SELECT wc2.collection_id
            FROM waste_collection wc2
            JOIN waste_request wr2 ON wc2.req_id = wr2.req_id
            WHERE wr2.admin_id = p_admin_id
            AND wc2.collection_id NOT IN (
                SELECT ar.collection_id
                FROM analysis_report ar
```

STORED PROCEDURE, FUNCTIONS AND TRIGGERS

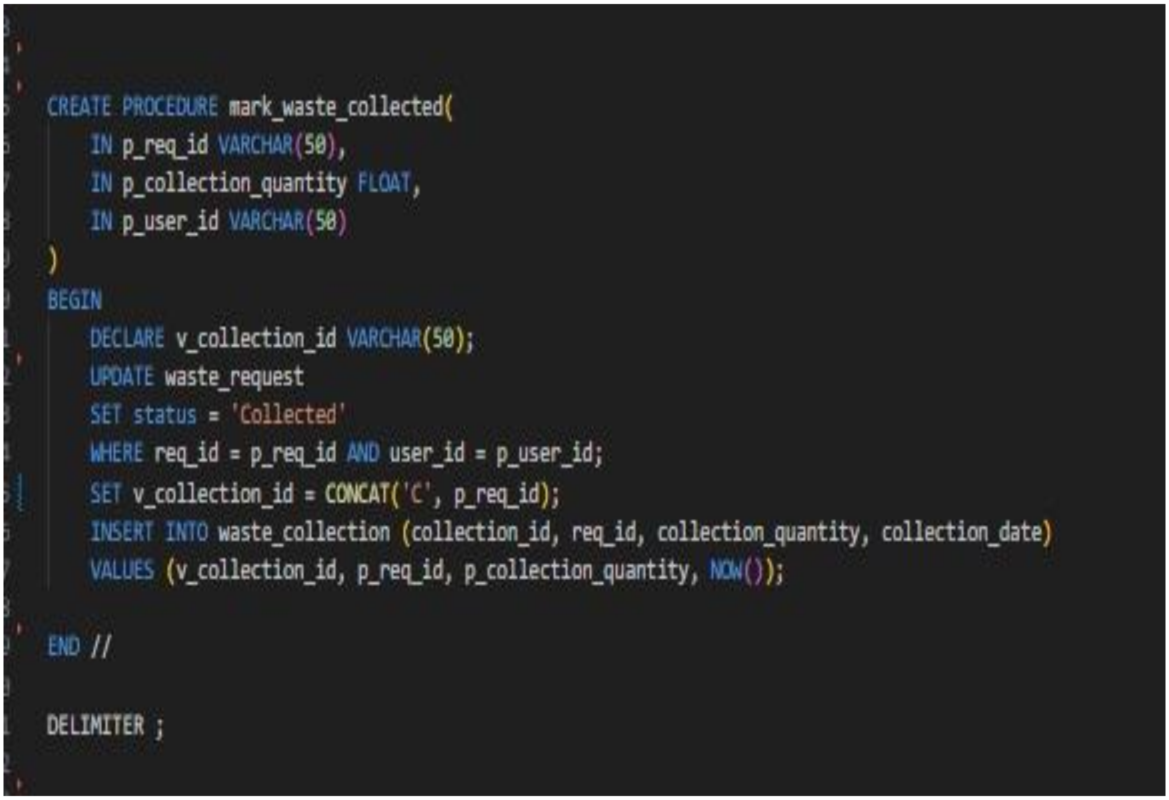
Triggers



```
DELIMITER //

CREATE TRIGGER after_vehicle_status_update
AFTER UPDATE ON vehicle
FOR EACH ROW
BEGIN
    IF NEW.status = 'active' AND OLD.status != 'active' THEN
        UPDATE waste_request
        SET vehicle_id = NEW.vehicle_id,
            status = 'Assigned'
        WHERE vehicle_id IS NULL
          AND status = 'Pending'
          AND waste_type = NEW.vehicle_type
          AND user_id IN (SELECT user_id FROM user WHERE centre_id = NEW.centre_id);
    END IF;
END //
DELIMITER ;
select * from vehicle;


DELIMITER //


CREATE TRIGGER after_waste_collection_insert
AFTER INSERT ON waste_collection
FOR EACH ROW
BEGIN
    -- Update the vehicle status to 'Not Active' after collection
    UPDATE vehicle v
    JOIN waste_request wr ON v.vehicle_id = wr.vehicle_id
    SET v.status = 'Not Active'
    WHERE wr.req_id = NEW.req_id;
END //

DELIMITER ;
```

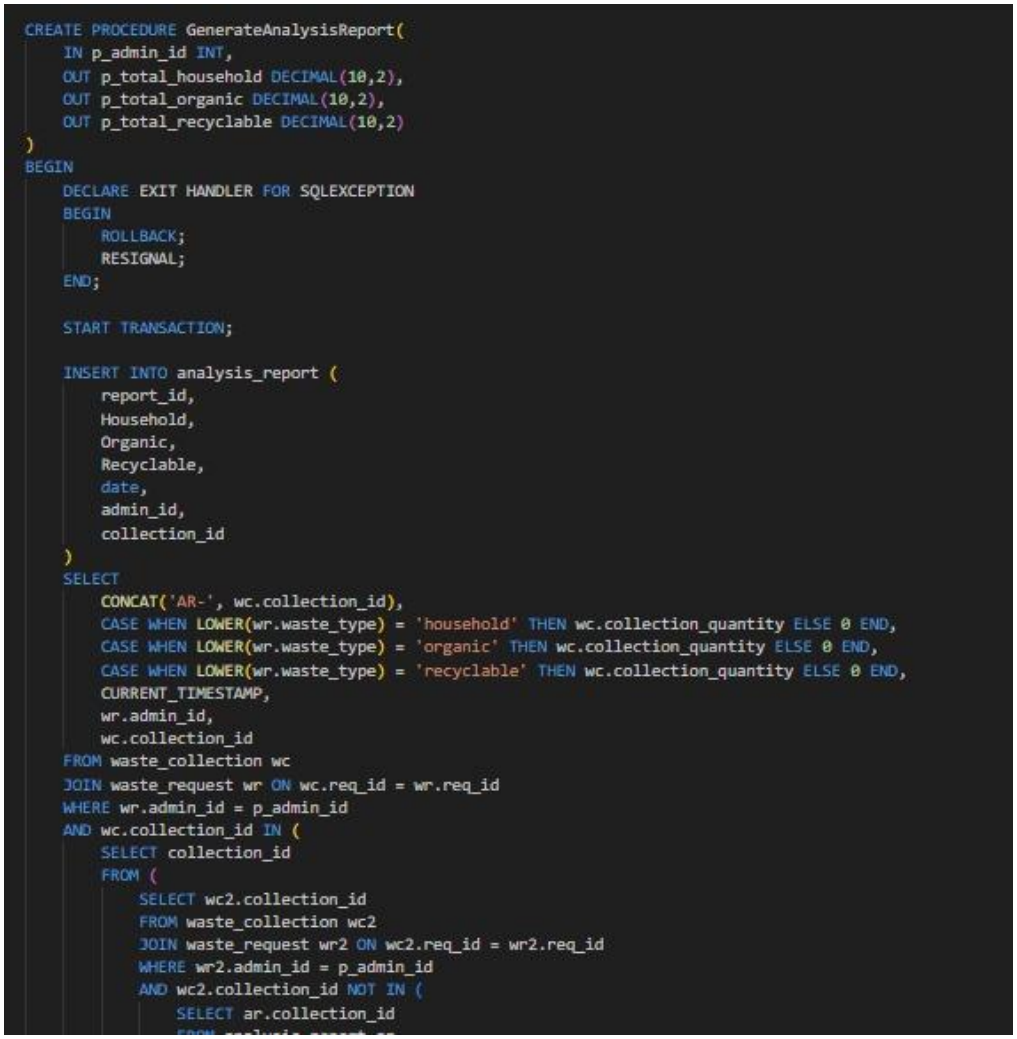
STORED PROCEDURE

Procedure 1

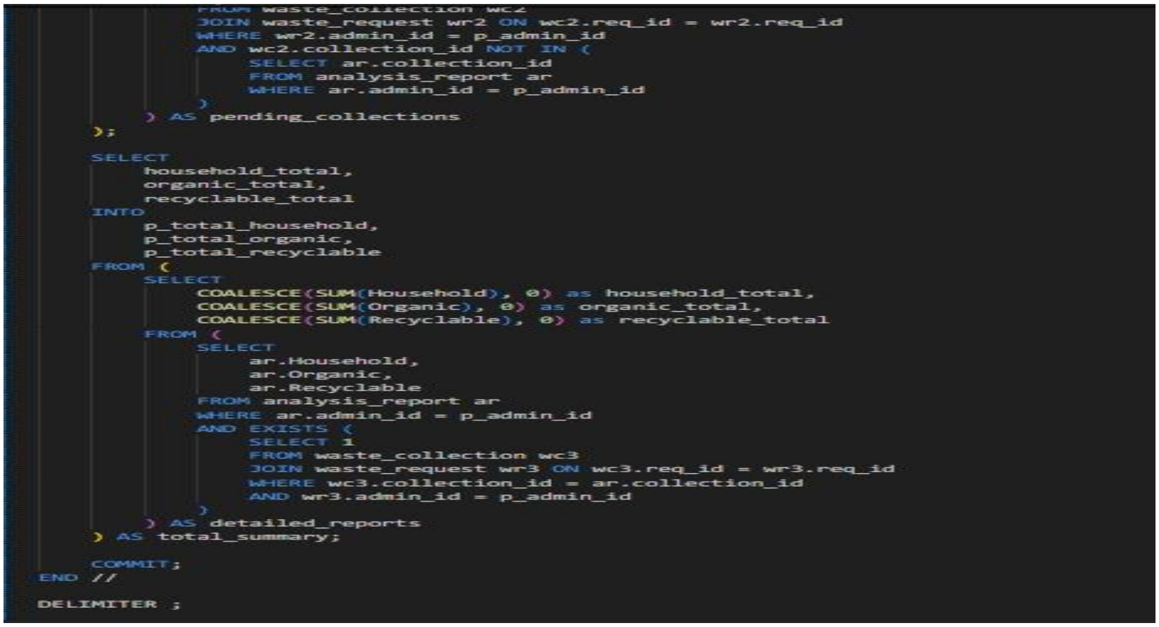


```
CREATE PROCEDURE mark_waste_collected(
    IN p_req_id VARCHAR(50),
    IN p_collection_quantity FLOAT,
    IN p_user_id VARCHAR(50)
)
BEGIN
    DECLARE v_collection_id VARCHAR(50);
    UPDATE waste_request
    SET status = 'Collected'
    WHERE req_id = p_req_id AND user_id = p_user_id;
    SET v_collection_id = CONCAT('C', p_req_id);
    INSERT INTO waste_collection (collection_id, req_id, collection_quantity, collection_date)
    VALUES (v_collection_id, p_req_id, p_collection_quantity, NOW());

END //

DELIMITER ;
```

Procedure 2



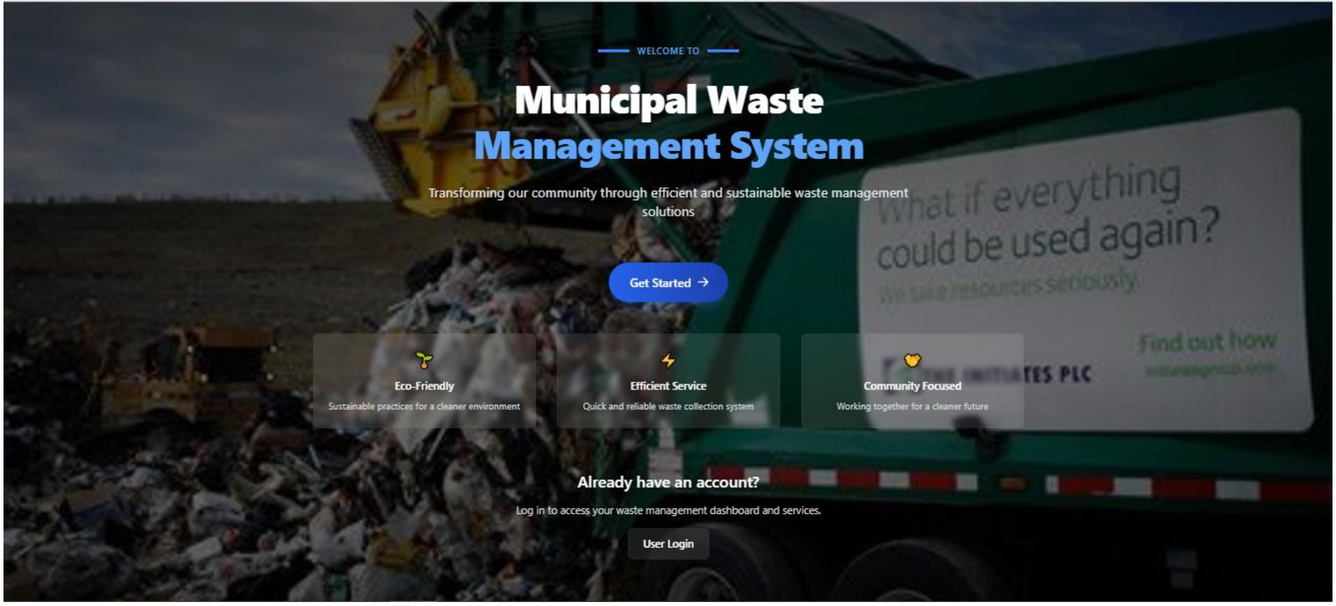
```sql
CREATE PROCEDURE GenerateAnalysisReport(
    IN p_admin_id INT,
    OUT p_total_household DECIMAL(10,2),
    OUT p_total_organic DECIMAL(10,2),
    OUT p_total_recyclable DECIMAL(10,2)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        RESIGNAL;
    END;

    START TRANSACTION;

    INSERT INTO analysis_report (
        report_id,
        Household,
        Organic,
        Recyclable,
        date,
        admin_id,
        collection_id
    )
    SELECT
        CONCAT('AR-', wc.collection_id),
        CASE WHEN LOWER(wr.waste_type) = 'household' THEN wc.collection_quantity ELSE 0 END,
        CASE WHEN LOWER(wr.waste_type) = 'organic' THEN wc.collection_quantity ELSE 0 END,
        CASE WHEN LOWER(wr.waste_type) = 'recyclable' THEN wc.collection_quantity ELSE 0 END,
        CURRENT_TIMESTAMP,
        wr.admin_id,
        wc.collection_id
    FROM waste_collection wc
    JOIN waste_request wr ON wc.req_id = wr.req_id
    WHERE wr.admin_id = p_admin_id
    AND wc.collection_id IN (
        SELECT collection_id
        FROM (
            SELECT wc2.collection_id
            FROM waste_collection wc2
            JOIN waste_request wr2 ON wc2.req_id = wr2.req_id
            WHERE wr2.admin_id = p_admin_id
            AND wc2.collection_id NOT IN (
                SELECT ar.collection_id
                FROM analysis_report ar
```



```sql
            FROM waste_collection wc2
            JOIN waste_request wr2 ON wc2.req_id = wr2.req_id
            WHERE wr2.admin_id = p_admin_id
            AND wc2.collection_id NOT IN (
                SELECT ar.collection_id
                FROM analysis_report ar
                WHERE ar.admin_id = p_admin_id
            )
        ) AS pending_collections
    );

    SELECT
        household_total,
        organic_total,
        recyclable_total
    INTO
        p_total_household,
        p_total_organic,
        p_total_recyclable
    FROM (
        SELECT
            COALESCE(SUM(Household), 0) as household_total,
            COALESCE(SUM(Organic), 0) as organic_total,
            COALESCE(SUM(Recyclable), 0) as recyclable_total
        FROM (
            SELECT
                ar.Household,
                ar.Organic,
                ar.Recyclable
            FROM analysis_report ar
            WHERE ar.admin_id = p_admin_id
            AND EXISTS (
                SELECT 1
                FROM waste_collection wc3
                JOIN waste_request wr3 ON wc3.req_id = wr3.req_id
                WHERE wc3.collection_id = ar.collection_id
                AND wr3.admin_id = p_admin_id
            )
        ) AS detailed_reports
    ) AS total_summary;

    COMMIT;
END //

DELIMITER ;
```

FRONT END DEVELOPMENT
(FUNCTIONALITIES/FEATURES OF THE APPLICATION)

Home Page:



WELCOME TO

Municipal Waste Management System

Transforming our community through efficient and sustainable waste management solutions

Eco-Friendly
Sustainable practices for a cleaner environment

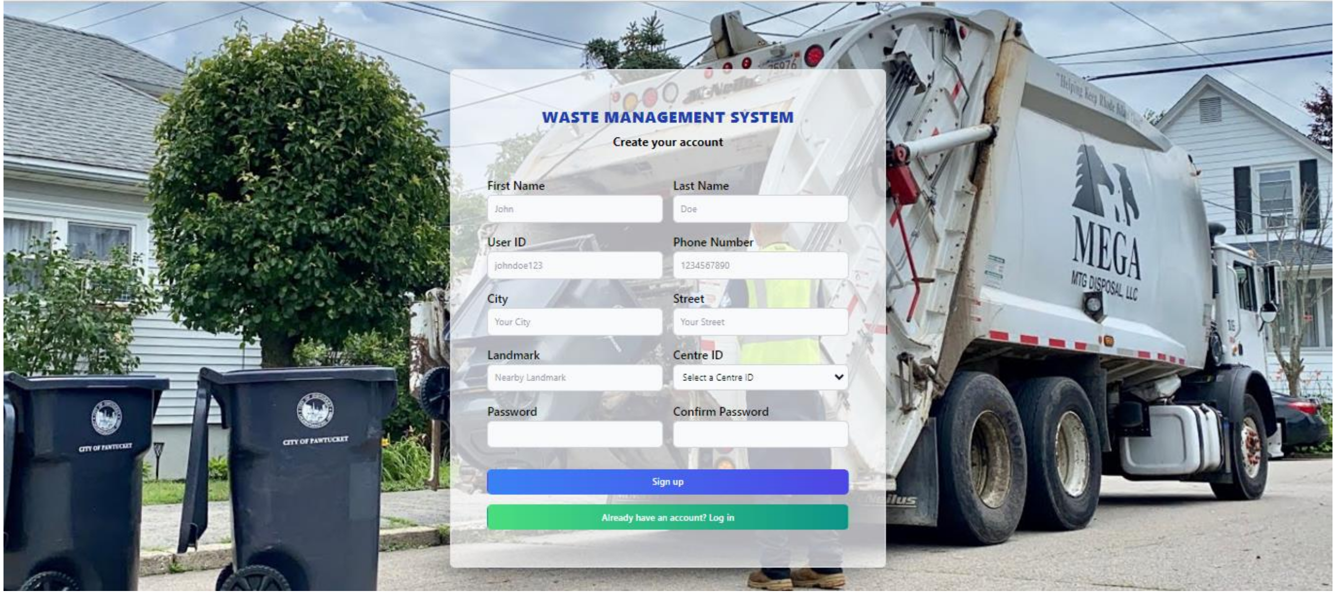
Efficient Service
Quick and reliable waste collection system

Community Focused
Working together for a cleaner future

Already have an account?

Log in to access your waste management dashboard and services.

User Signup:

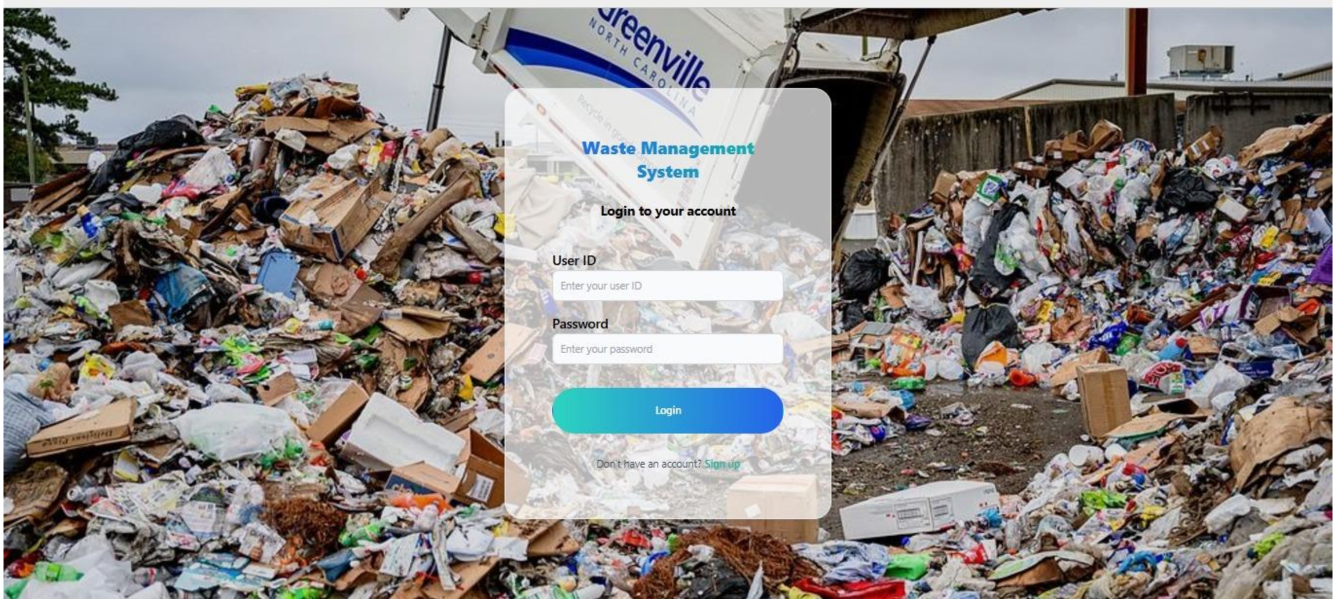


WASTE MANAGEMENT SYSTEM

Create your account

First Name: John
Last Name: Doe
User ID: johndoe123
Phone Number: 1234567890
City: Your City
Street: Your Street
Landmark: Nearby Landmark
Centre ID: Select a Centre ID
Password:
Confirm Password:

User login:



Waste Management System

Login to your account

User ID

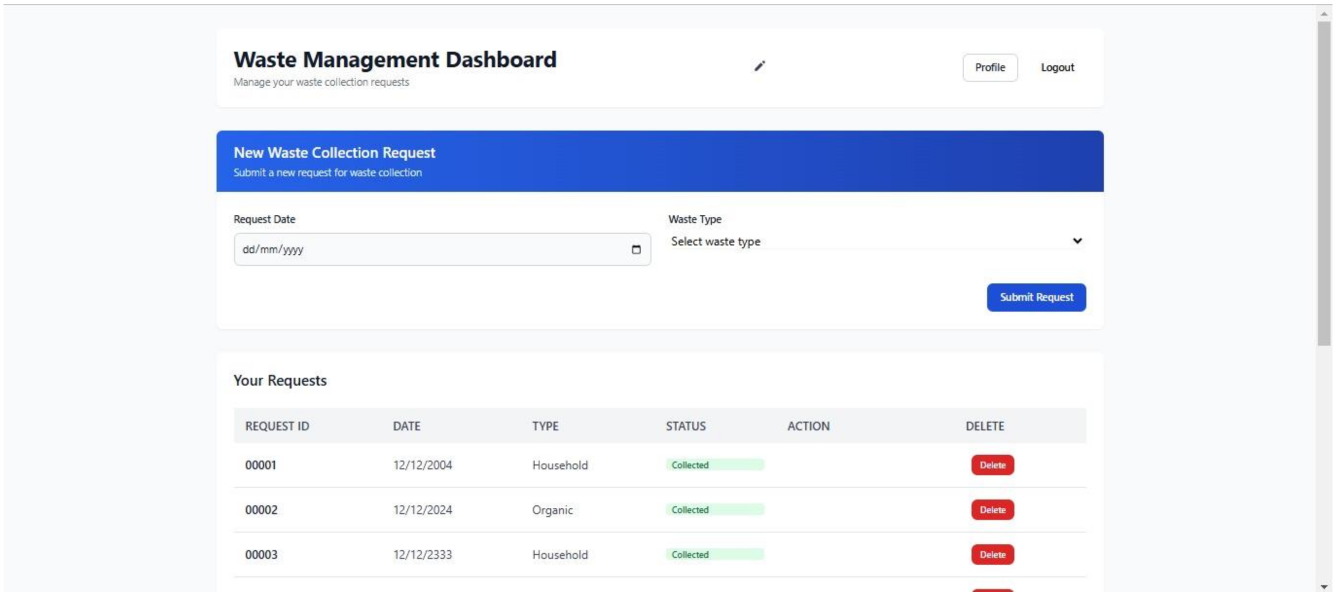
Enter your user ID

Password

Enter your password

Don't have an account? Sign up

User Dashboard:



Waste Management Dashboard

Manage your waste collection requests

New Waste Collection Request

Submit a new request for waste collection

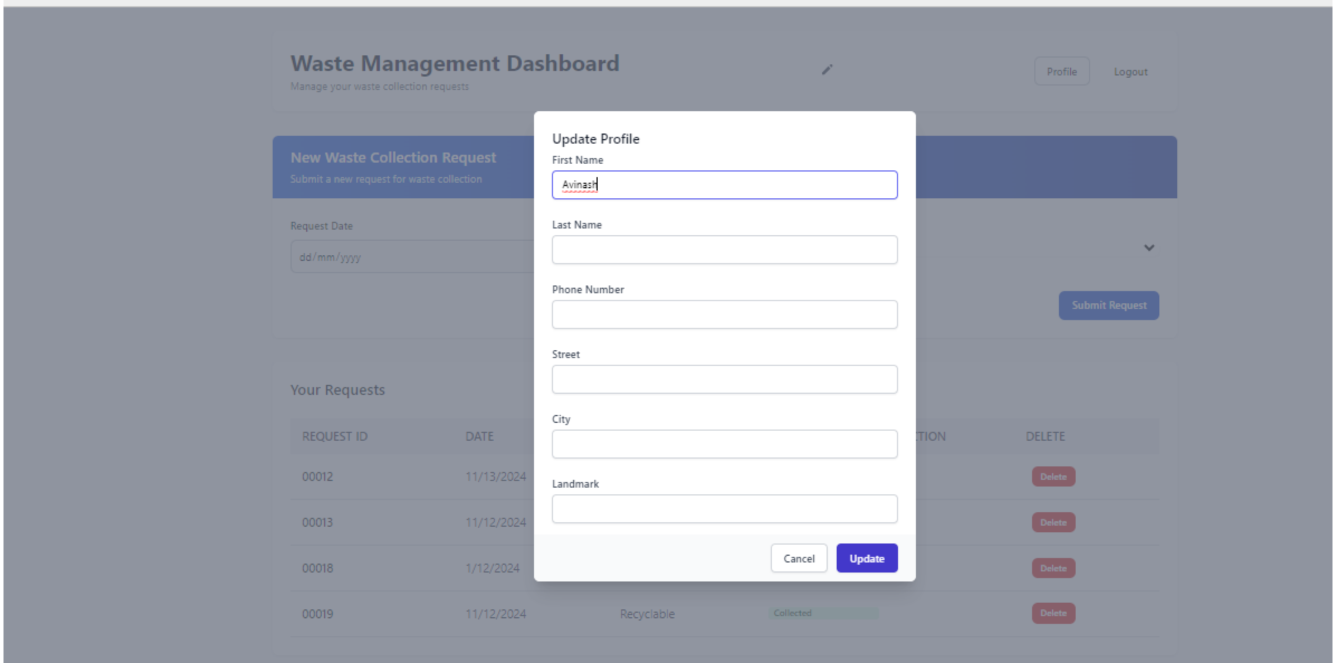
Request Date: dd/mm/yyyy

Waste Type: Select waste type

Your Requests

REQUEST ID	DATE	TYPE	STATUS	ACTION	DELETE
00001	12/12/2004	Household	Collected		Delete
00002	12/12/2024	Organic	Collected		Delete
00003	12/12/2333	Household	Collected		Delete

Updating User Profile:



Waste Management Dashboard

Manage your waste collection requests

New Waste Collection Request

Submit a new request for waste collection

Request Date

dd/mm/yyyy

Your Requests

REQUEST ID	DATE			TION	DELETE
00012	11/13/2024				Delete
00013	11/12/2024				Delete
00018	1/12/2024				Delete
00019	11/12/2024	Recyclable	Collected		Delete

Update Profile

First Name: Avinash

Last Name

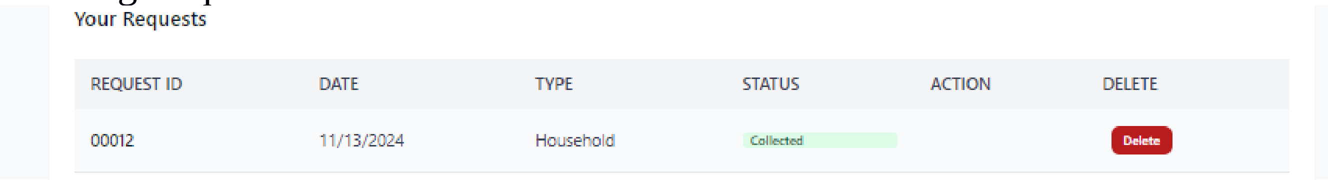
Phone Number

Street

City

Landmark

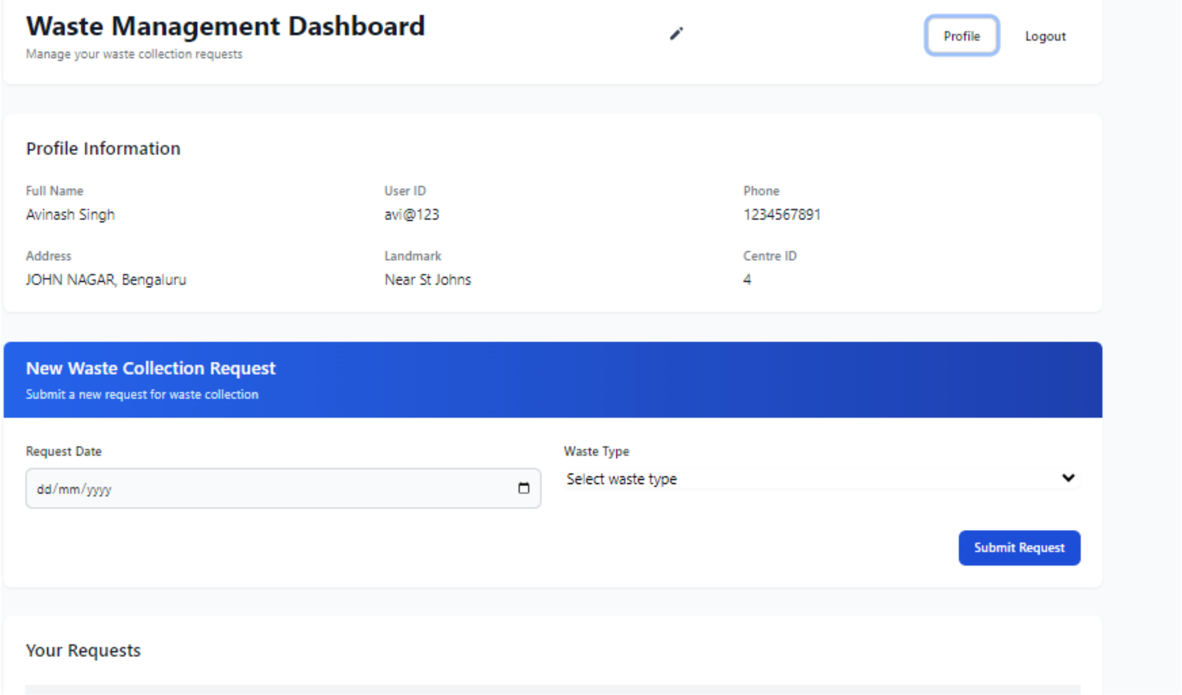
Deleting Request:



Your Requests

REQUEST ID	DATE	TYPE	STATUS	ACTION	DELETE
00012	11/13/2024	Household	Collected		Delete

Viewing Profile:



Waste Management Dashboard

Manage your waste collection requests

Profile Information

Full Name Avinash Singh	User ID avi@123	Phone 1234567891
Address JOHN NAGAR, Bengaluru	Landmark Near St Johns	Centre ID 4

New Waste Collection Request

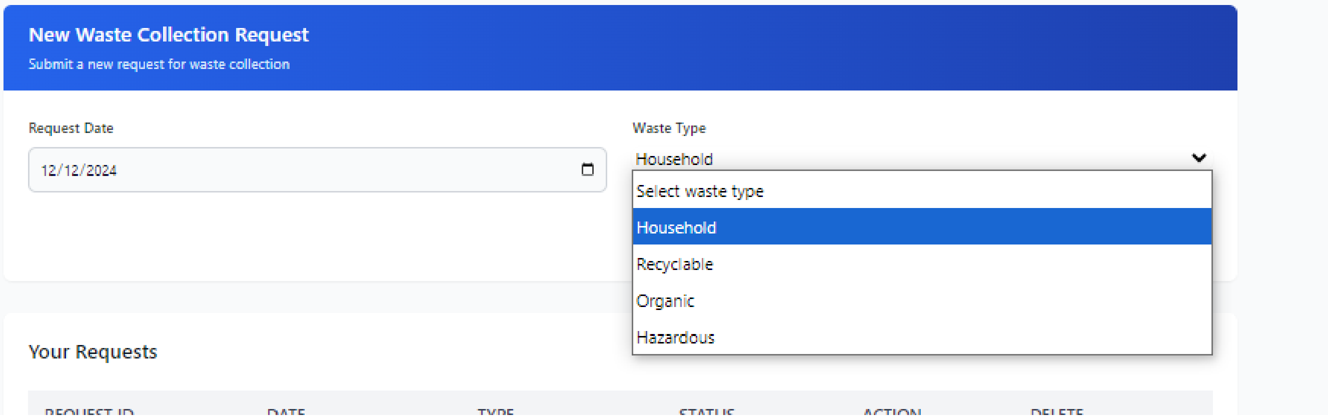
Submit a new request for waste collection

Request Date: dd/mm/yyyy

Waste Type: Select waste type

Your Requests

Making Request:



New Waste Collection Request

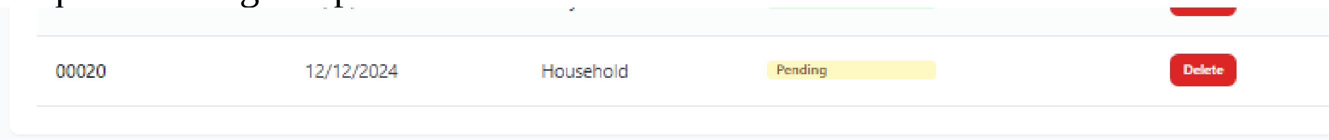
Submit a new request for waste collection

Request Date: 12/12/2024

Waste Type: Household

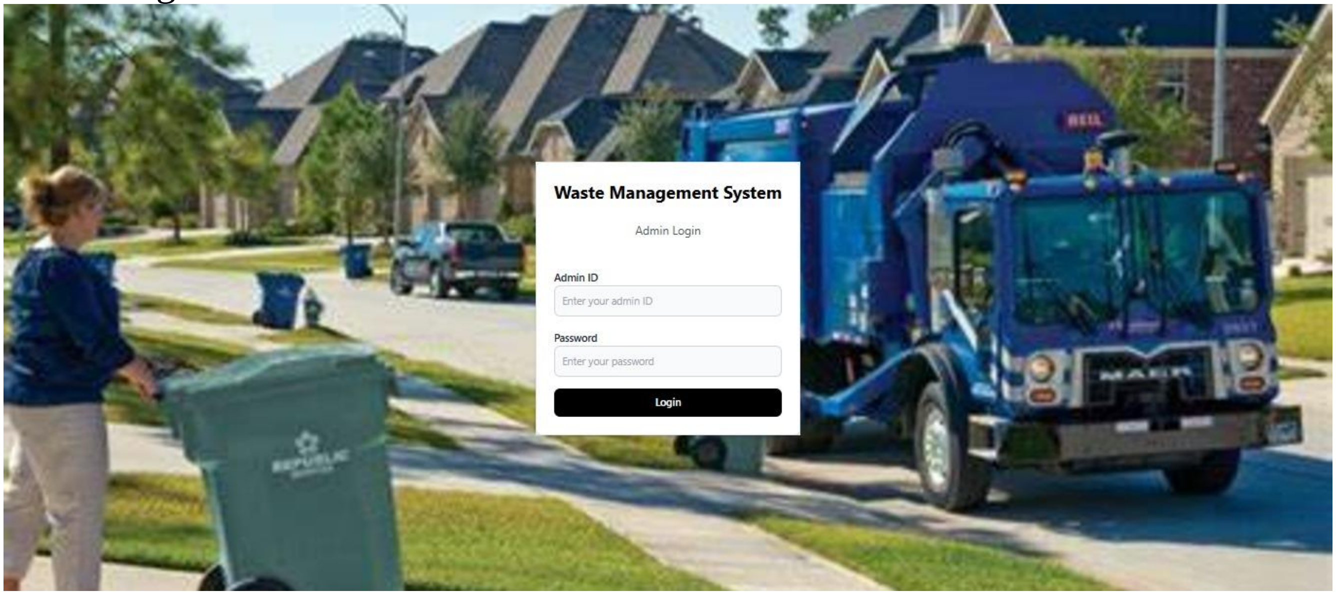
Your Requests

Request Table gets updated:



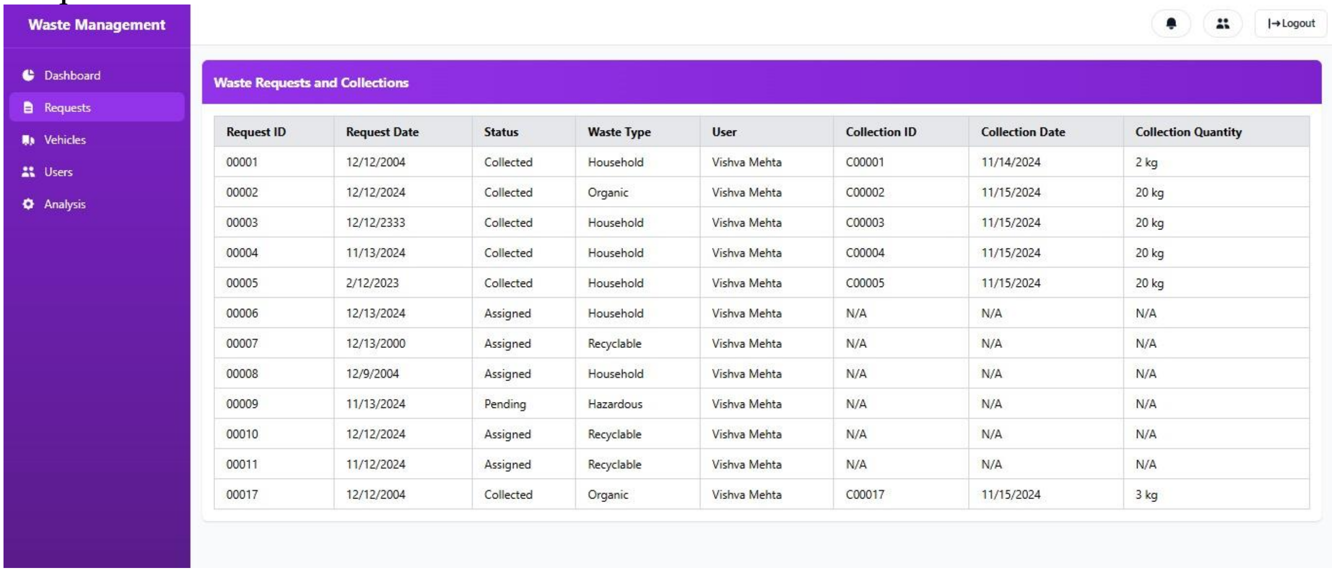
00020	12/12/2024	Household	Pending	Delete

Admin login:



Admin Dashboard:

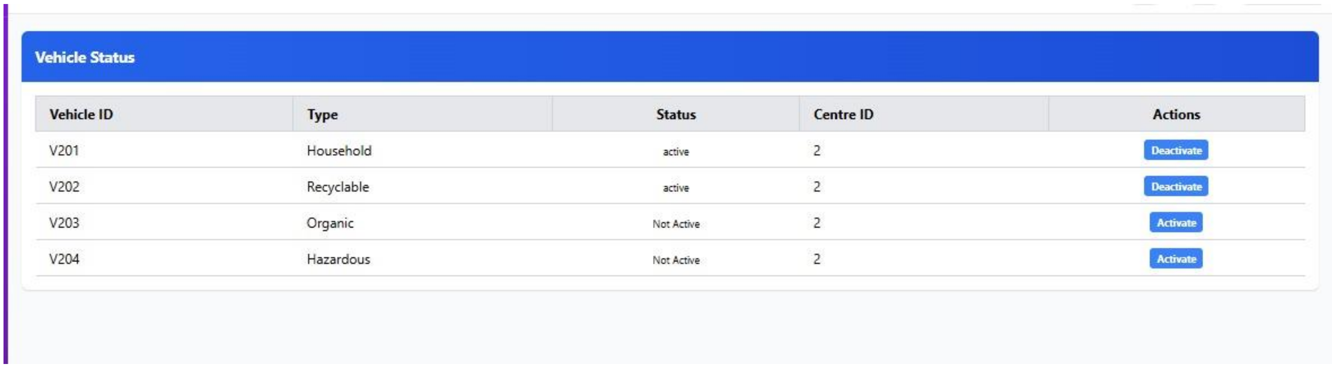
Requests and Collections Table Joined:



Waste Requests and Collections

Request ID	Request Date	Status	Waste Type	User	Collection ID	Collection Date	Collection Quantity
00001	12/12/2004	Collected	Household	Vishva Mehta	C00001	11/14/2024	2 kg
00002	12/12/2024	Collected	Organic	Vishva Mehta	C00002	11/15/2024	20 kg
00003	12/12/2333	Collected	Household	Vishva Mehta	C00003	11/15/2024	20 kg
00004	11/13/2024	Collected	Household	Vishva Mehta	C00004	11/15/2024	20 kg
00005	2/12/2023	Collected	Household	Vishva Mehta	C00005	11/15/2024	20 kg
00006	12/13/2024	Assigned	Household	Vishva Mehta	N/A	N/A	N/A
00007	12/13/2000	Assigned	Recyclable	Vishva Mehta	N/A	N/A	N/A
00008	12/9/2004	Assigned	Household	Vishva Mehta	N/A	N/A	N/A
00009	11/13/2024	Pending	Hazardous	Vishva Mehta	N/A	N/A	N/A
00010	12/12/2024	Assigned	Recyclable	Vishva Mehta	N/A	N/A	N/A
00011	11/12/2024	Assigned	Recyclable	Vishva Mehta	N/A	N/A	N/A
00017	12/12/2004	Collected	Organic	Vishva Mehta	C00017	11/15/2024	3 kg

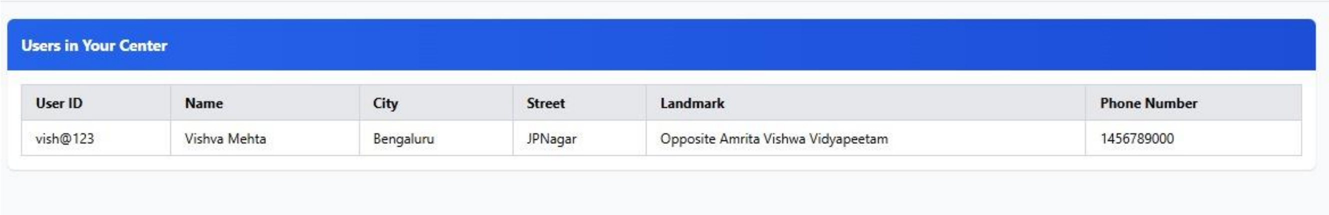
Vehicles Details:



Vehicle Status

Vehicle ID	Type	Status	Centre ID	Actions
V201	Household	active	2	Deactivate
V202	Recyclable	active	2	Deactivate
V203	Organic	Not Active	2	Activate
V204	Hazardous	Not Active	2	Activate

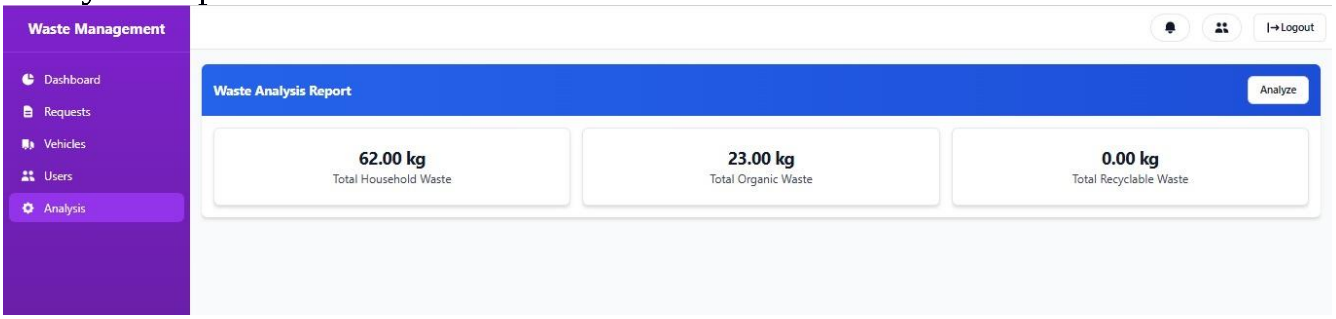
Users Details:



Users in Your Center

User ID	Name	City	Street	Landmark	Phone Number
vish@123	Vishva Mehta	Bengaluru	JPNagar	Opposite Amrita Vishwa Vidyapeetam	1456789000

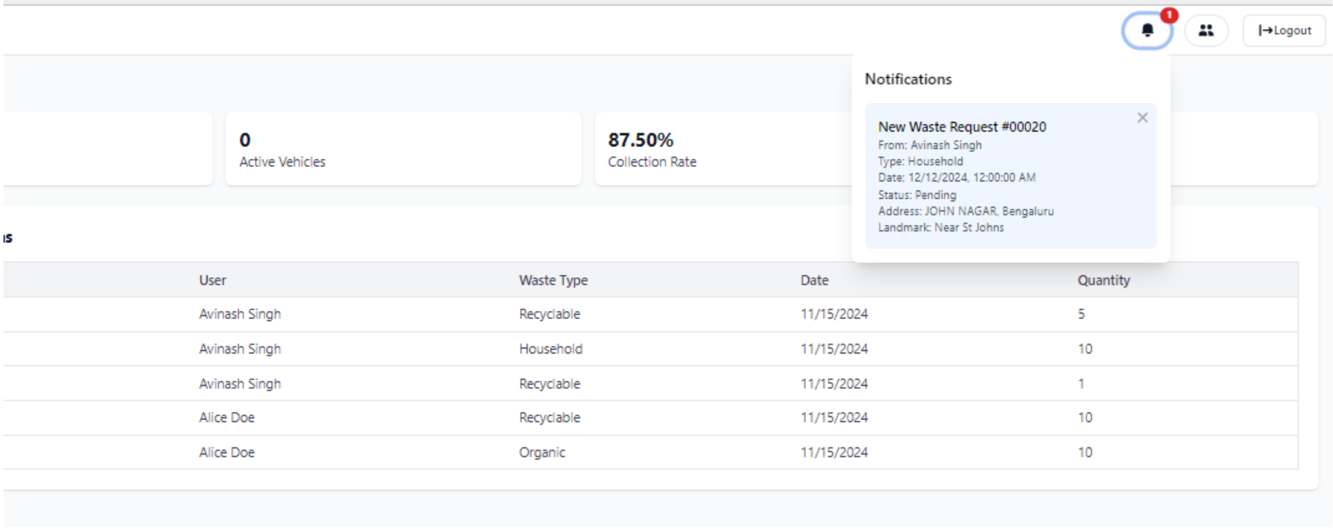
Analysis Report Details:



Waste Analysis Report

62.00 kg	23.00 kg	0.00 kg
Total Household Waste	Total Organic Waste	Total Recyclable Waste

Notification to the admin when new request made:



Notifications

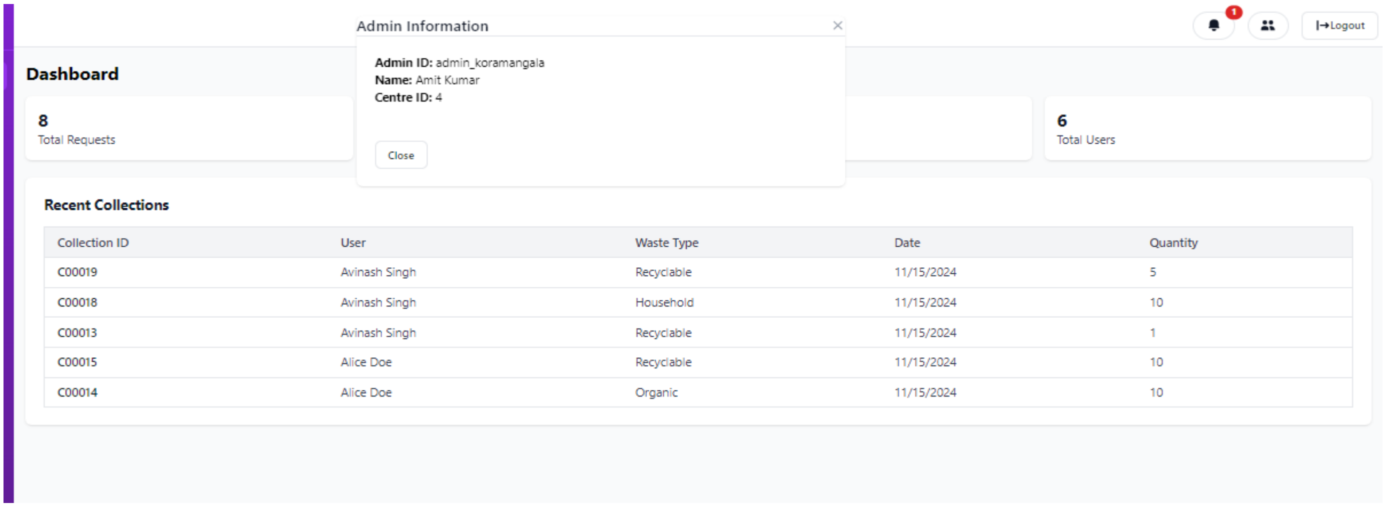
New Waste Request #00020
From: Avinash Singh
Type: Household
Date: 12/12/2024, 12:00:00 AM
Status: Pending
Address: JOHN NAGAR, Bengaluru
Landmark: Near St Johns

0
Active Vehicles

87.50%
Collection Rate

User	Waste Type	Date	Quantity
Avinash Singh	Recyclable	11/15/2024	5
Avinash Singh	Household	11/15/2024	10
Avinash Singh	Recyclable	11/15/2024	1
Alice Doe	Recyclable	11/15/2024	10
Alice Doe	Organic	11/15/2024	10

View Admin Profile:



Admin Information

Admin ID: admin_koramangala
Name: Amit Kumar
Centre ID: 4

Close

Dashboard

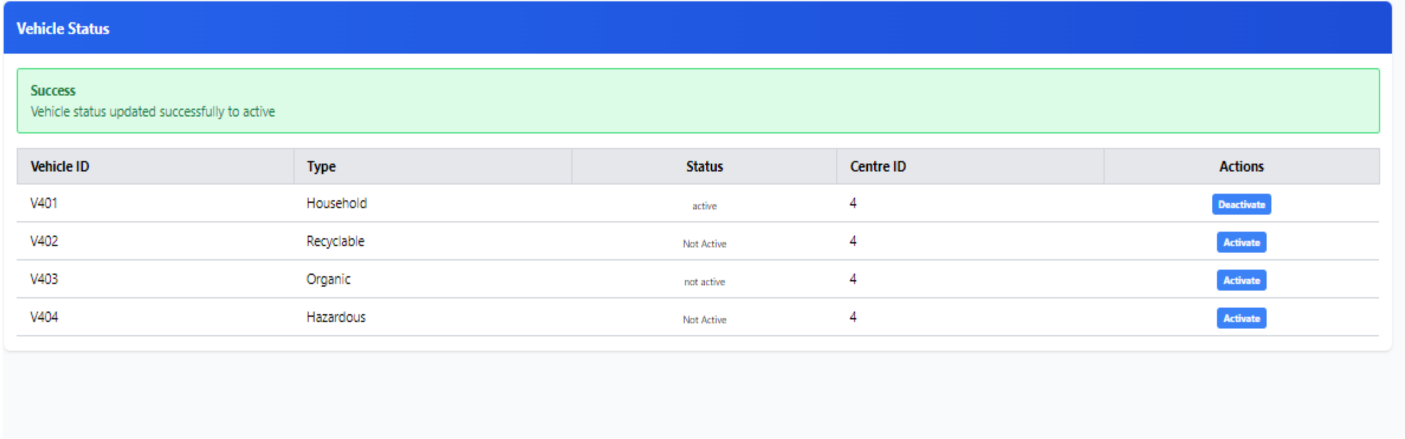
8
Total Requests

6
Total Users

Recent Collections

Collection ID	User	Waste Type	Date	Quantity
C00019	Avinash Singh	Recyclable	11/15/2024	5
C00018	Avinash Singh	Household	11/15/2024	10
C00013	Avinash Singh	Recyclable	11/15/2024	1
C00015	Alice Doe	Recyclable	11/15/2024	10
C00014	Alice Doe	Organic	11/15/2024	10

Assigning Vehicles:

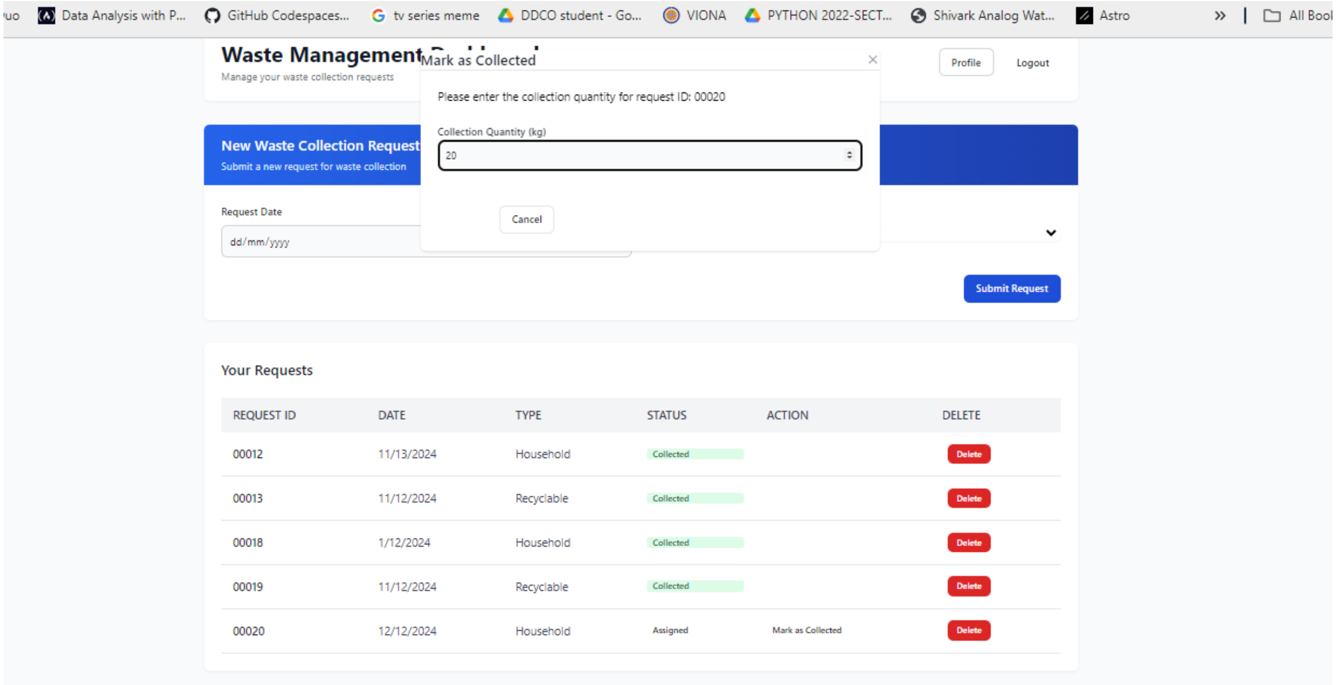


Vehicle Status

Success
Vehicle status updated successfully to active

Vehicle ID	Type	Status	Centre ID	Actions
V401	Household	active	4	Deactivate
V402	Recyclable	Not Active	4	Activate
V403	Organic	not active	4	Activate
V404	Hazardous	Not Active	4	Activate

Once Vehicle gets assigned waste request status updated to assigned (User Dashboard):



Waste Management Dashboard
Manage your waste collection requests

Mark as Collected

Please enter the collection quantity for request ID: 00020

Collection Quantity (kg): 20

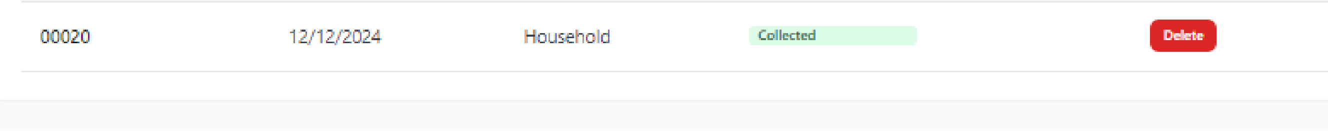
New Waste Collection Request
Submit a new request for waste collection

Request Date: dd/mm/yyyy

Your Requests

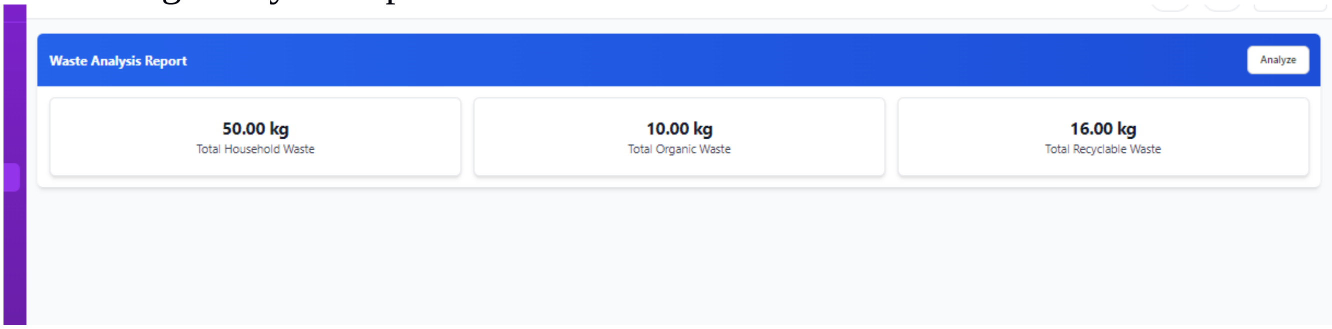
REQUEST ID	DATE	TYPE	STATUS	ACTION	DELETE
00012	11/13/2024	Household	Collected		Delete
00013	11/12/2024	Recyclable	Collected		Delete
00018	1/12/2024	Household	Collected		Delete
00019	11/12/2024	Recyclable	Collected		Delete
00020	12/12/2024	Household	Assigned	Mark as Collected	Delete

On clicking marked as collected and entering collection quantity (User Dashboard):



00020	12/12/2024	Household	Collected	Delete

Generating Analysis Report:



Waste Analysis Report

50.00 kg	10.00 kg	16.00 kg
Total Household Waste	Total Organic Waste	Total Recyclable Waste

Github Repository Link

https://github.com/Vionarose9/Waste-Management-System

References/Bibliography

1. Flask Documentation: https://flask.palletsprojects.com/

2. Flask with React: https://www.geeksforgeeks.org/how-to-connect-reactjs-with-flask-api/

3. Flask, react and Mysql: https://medium.com/@sandyjtech/building-a-full-stack-app-with-flask-react-mysql-a78fcc235ff0

4. MySQL Docs: https://dev.mysql.com/doc/

5. Flow-Bite Docs: https://flowbite.com/docs/getting-started/introduction/

APPENDIX A DEFINITIONS, ACRONYMS AND ABBREVIATIONS

a. ORM: Object Relational Mapping, a way of bridging classes in OOPS to schemas in DBMS