*project report on*

# E-WASTE MANAGEMENT SYSTEM

*Submitted in partial fulfillment for the award of the degree of*

**Computer Science Engineering - Specialisation in Artificial Intelligence and Machine Learning**

*by*

**VEKANURU RAVI SHANKAR ( 21BCE9456 )**

**GANTA MANJUNADH REDDY ( 21BCE7456 )**

**DASARI GUNATEJ UTHAM ( 21BCE9564 )**

**School of Computer Science and Engineering (SCOPE)**

May, 2025

# DECLARATION

I hereby declare that the thesis entitled "**E-WASTE MANAGEMENT SYSTEM**" submitted by me, for the award of the degree of Computer Science Engineering - Specialisation in Artificial Intelligence and Machine Learning VIT is a record of bonafide work carried out by me under the supervision of **Dr. Ikkurthi Bhanu Prakash**

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

G. [illegible] Reddy V. Ravi Shankar

D. Otham

Place: Amaravati

Date: 21-May-2025 Signature of the Candidate

# CERTIFICATE

This is to certify that the Senior Design Project titled "**E-WASTE MANAGEMENT SYSTEM**" that is being submitted by **VEKANURU RAVI SHANKAR ( 21BCE9456 ), GANTA MANJUNADH REDDY( 21BCE7456 ), DASARI GUNATEJ UTHAM( 21BCE9564 )** is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

I. Bhanu Prasad..

**Dr. Ikkurthi Bhanu Prakash**

**Guide**

**The thesis is satisfactory / unsatisfactory**

21/5/25

**Dr. Ikkurthi Bhanu Prakash**

**Internal Examiner**

21/5/25

**Dr. Shalini Subramani**

**External Examiner**

**Approved by**

Dr.Reeja S R

**HoD, Department of Artificial Intelligence and Machine Learning**

## ABSTRACT

The exponential rise in the use of electronic devices in this age of fast technical innovation has resulted in a notable rise in electronic trash, or "e-waste." Because it contains precious metals and hazardous components, improper disposal of this waste presents serious risks to human health and the environment. This project introduces an Internet of Things (IoT)-based E-garbage Management System that uses sensor-equipped smart bins to automate garbage sorting and monitoring. Using sensor data, the system measures the waste's density, identifies the fill level, and assesses the presence of metals. The system recommends appropriate measures based on this analysis, either to remove recyclable metals or to properly dispose of the garbage. The online dashboard, which was created using HTML, CSS, and JavaScript, gives customers access to e-waste analytics, recycling advice, awareness posters, and real-time bin data. MongoDB is used for data storage, and Node.js and Express.js power the backend. Enhancing e-waste segregation and encouraging environmentally friendly waste management techniques are the goals of this system.

**Keywords**: E-waste, Waste Management, Recycling, Environmental Sustainability, Smart Bin, Hazardous Materials, Electronic Waste Disposal, IoT-based System, Green Technology, Digital Waste Tracking.

# ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Ikkurthi Bhanu Prakash, Assistant Professor Senior Grade-1, SCOPE, VIT-AP, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with his is not confined to academics only, but it is a great opportunity on my part to work with an intellectual and expert in the field of computer science.

I would like to express my gratitude to Dr. G. Viswanathan, Mr. Sankar Viswanathan , Dr. S. V. Kota Reddy, and Dr. Sudhakar Ilango , School of Computer Science and Engineering, for providing me with an environment to work in and for his inspiration during the tenure of the course.

In a jubilant mood I express ingeniously my whole-hearted thanks to Dr. Reeja S R. Professor Grade-2, SCOPE, VIT-AP, all teaching staff and members working as members of our university for their not-self-centered enthusiasm coupled with timely encouragement showered on me with zeal, which prompted the acquisition of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

G. [illegible] Reddy V. Ravi Shankar

D. Otham

Place: Amaravati

Date: 21-May-2025 Signature of Candidate

# CONTENTS :

### Chapter 1

# Introduction

The manufacturing and use of electronic gadgets have increased in tandem with the quick development of information technology and electronics. Globally, enormous amounts of e-waste are produced as obsolescence increases as a result of innovation and market dynamics. In 2019 alone, 53.6 million metric tons of e-waste were produced worldwide, according to a UN report [1]. Precious metals, polymers, and dangerous compounds like lead, mercury, and cadmium are among the complex mixture of components that make up e-waste, which calls for specific handling and recycling procedures [2].

Poor e-waste management leads to serious health problems and environmental damage, particularly in developing nations. Open burning and acid baths are examples of informal recycling methods that discharge harmful materials into the air, soil, and water, impacting both human populations and animals. [3][4]. Thus, creating efficient e-waste management plans is essential to sustainable growth.

By using a centralized, digital strategy, this initiative seeks to overcome the inefficiencies in the present e-waste management systems. The platform provided by our suggested "E-Waste Management System" allows consumers to register their e-waste, find authorized collection facilities, arrange pickups, and access instructional resources. By bridging the gap between e-waste producers and approved recyclers, the system hopes to encourage appropriate disposal practices. However, conventional trash disposal methods are frequently unprepared to deal with the unique difficulties presented by e-waste. Lacking the technological capacity to distinguish between various waste components according to material composition, volume, or recyclability, they usually rely on manual segregation. Furthermore, most e-waste either ends up in unsorted dumps or is processed without recovering its valuable ingredients, which results in resource loss and environmental deterioration. This is because real-time monitoring and automated classification are lacking.

Our research suggests a Smart E-Waste Management System that uses web and Internet of Things (IoT) technologies to transform the way e-waste is managed in order to address these issues. This system integrates several sensors into specifically made smart bins to automate the garbage monitoring, classification, and decision-making processes. Key factors like the bin's fill level, the presence of metallic materials, and the density of the garbage dumped can all be detected by these sensors. The system may prioritize the extraction of valuable metals and differentiate between recyclable and non-recyclable waste by evaluating these factors.

In addition, our system has a web-based dashboard that helps users and authorities make educated decisions by offering real-time metrics, warnings, and visualizations. Through interactive educational materials and environmentally friendly disposal advice, it also seeks to raise public awareness of the dangers posed by e-waste and the value of recycling. Our project, which bridges the technical divide in trash management, supports environmental sustainability and the circular economy's tenets of resource reuse and waste minimization.

In the end, the suggested Smart E-Waste Management System shows how to address one of the most important problems of our day in a sustainable and scalable manner.  With the help of data-driven analysis, community involvement, and intelligent sensing, the system hopes to provide a more responsible, intelligent, and clean framework for handling electronic trash.

### Chapter 2

# Literature Survey

The swift increase in the production of e-waste has spurred a great deal of study into creative approaches to effective trash management. The literature has put out a number of strategies and approaches to deal with various facets of the e-waste issue.

A study in [3] suggested a sensor-based system that uses ultrasonic sensors to track the amounts of general garbage bins. This system's utility for targeted e-waste management was limited because it was unable to differentiate between electronic waste and other types of trash, despite being effective at detecting bin fill levels. On the other hand, the study in [4] showed the possibility of intelligent categorization by classifying waste categories using machine learning techniques. However, the suggested solution's practical usefulness was diminished because it was limited to software simulations and lacked physical implementation or sensor integration. Further developments were made in [5], where the authors created a smart bin that uses ultrasonic sensors to determine fullness levels. This method did not go beyond volume sensing, but it did offer a basic hardware model for smart waste monitoring. In the meantime, inductive sensors were used in the study in [6] to detect the presence of metal in garbage. Despite being successful in separating metallic components, the system did not integrate numerous sensing capabilities, use density analysis, or have a more comprehensive waste categorization mechanism.

The lack of thorough waste evaluation based on density, which is essential for making well-informed judgments on metal extraction and recyclability, is a common weakness among these systems notwithstanding these contributions. Few researchers have tried to combine several sensing metrics, like density, volume, and metal content, into a single framework that can categorize and direct the processing of e-waste on its own.

By offering an integrated, multi-sensor solution, our suggested system overcomes the shortcomings of these earlier research and expands upon the fundamental concepts they introduced. It enables automated, real-time decision-making for processing e-waste by combining density-based classification, metal presence verification (using inductive sensors), and fill level detection (using ultrasonic sensors). This combination makes it possible to evaluate the waste more precisely, which promotes targeted recycling and lessens environmental damage.

Furthermore, in contrast to earlier systems, our strategy places a strong emphasis on public awareness and user engagement—aspects that are sometimes disregarded in technological implementations. Our system's web-based dashboard doubles as an instructional tool in addition to providing real-time analytics and actionable data from the smart bins. It displays e-waste facts, recycling tips, and awareness posters to encourage responsible disposal behavior among users.

Our system seeks to close the gap between public involvement and technology capabilities by combining intelligent sensing with user-centric features, thus advancing an all-encompassing approach to e-waste management.

In addition to scholarly research, a number of practical projects have tried to use technology to manage e-waste. For example, legal frameworks for recycling e-waste have been established by the European Union's WEEE (Waste Electrical and Electronic Equipment) directive; yet, a large portion of compliance depends on human sorting and conventional infrastructure, which lacks real-time monitoring capabilities. Similar to this, commercial smart bin systems used in urban waste management are useless for processing specific e-waste since they frequently concentrate on general trash pickup logistics rather than item classification. Research in [7] investigated the classification of recyclable waste using deep learning and image recognition, demonstrating the potential of software-based identification. However, these systems are less appropriate for low-cost IoT deployments in public or rural locations due to their high computing requirements and sensitivity to ambient factors like lighting and camera angle.

Additionally, a number of Internet of Things-based trash management solutions have been put out that send bin data to municipal servers via GSM or Wi-Fi modules [8]. Although these systems enhance communication and logistics, they usually just provide fill levels and are not intelligent enough to categorize materials according to their density or conductivity. Systems that assess garbage holistically by integrating several sensor readings into a recycling decision-making process still have a glaring lack. Therefore, our suggested system offers a more thorough method of waste classification by combining density estimates, metal identification, and volume detection into a single framework. Through its web-based interface, it not only facilitates automated, sensor-driven recycling logic but also improves outreach and transparency. The system is positioned as a behavioral and technological response to the ongoing e-waste problem because of its simultaneous emphasis on automation and awareness.

**Chapter 3**

# Methodology

The approach used to create the E-trash Management System (EWMS) is centered on creating and putting into practice a complete system for monitoring, gathering, and handling electronic trash. Through a single digital interface, the system seeks to raise awareness, encourage sustainable recycling habits, and simplify responsible disposal. The method combines a standardized database design for effective data processing with software elements for user engagement and management [1].

## 3.0 SYSTEM DESIGN

Given its modular design, the EWMS allows for the scalable incorporation of future features and the division of responsibilities. The user module, admin module, and database are the three primary parts of the system. People may register, report e-waste, and examine drop-off information using the user module. Verification, classification, and documentation are managed by the admin module. User profiles, e-waste records, and waste types are among the permanent data kept in the database [2]. To guarantee the separation of data management (Model), user interface (View), and application functionality (Controller), a Model-View-Controller (MVC) design pattern was employed. This design decision facilitates collaborative development and improves maintainability [3].

## 3.1 TECHNOLOGY STACK

The following stack is utilized by the system to preserve accessibility and simplicity:

- Frontend: HTML, CSS, and JavaScript for dynamic form handling and a responsive user interface.
- Node.js and Express.js are used in the backend to control server-side functions and routing.
- Database: MongoDB is a NoSQL database that was selected due to its adaptability and JSON-like structure, which makes it ideal for managing a variety of data types [4].

Instead of depending on cloud hosting or commercial platforms, the system maintains its cost-effectiveness and ease of deployment in local contexts by choosing open-source and widely supported technologies [5].

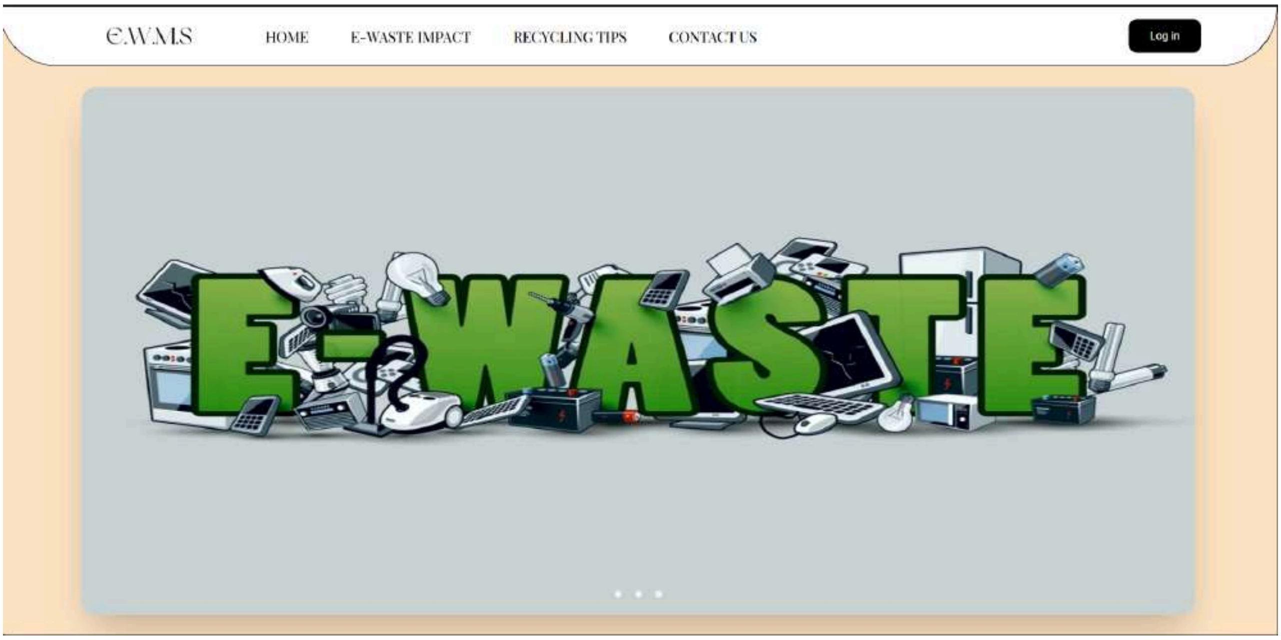
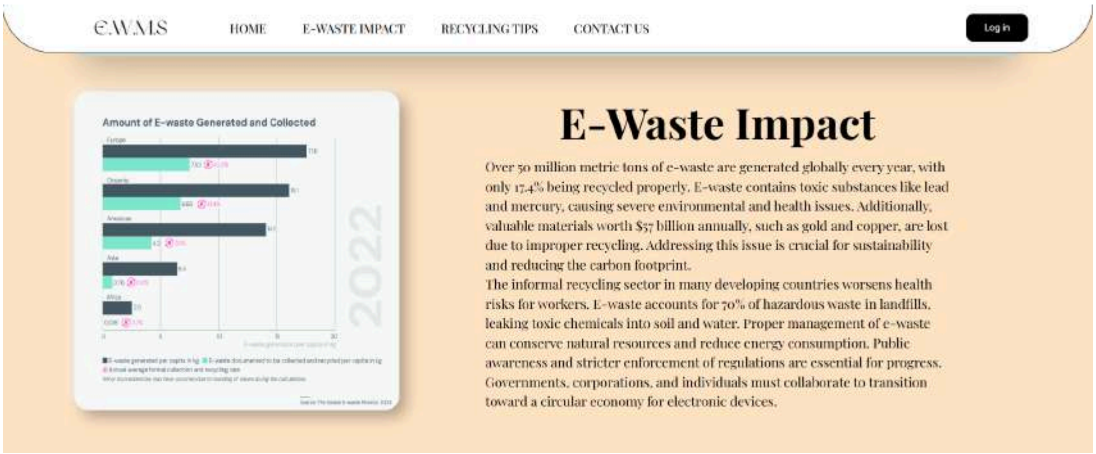


Fig- 1: Landing Page of the Website

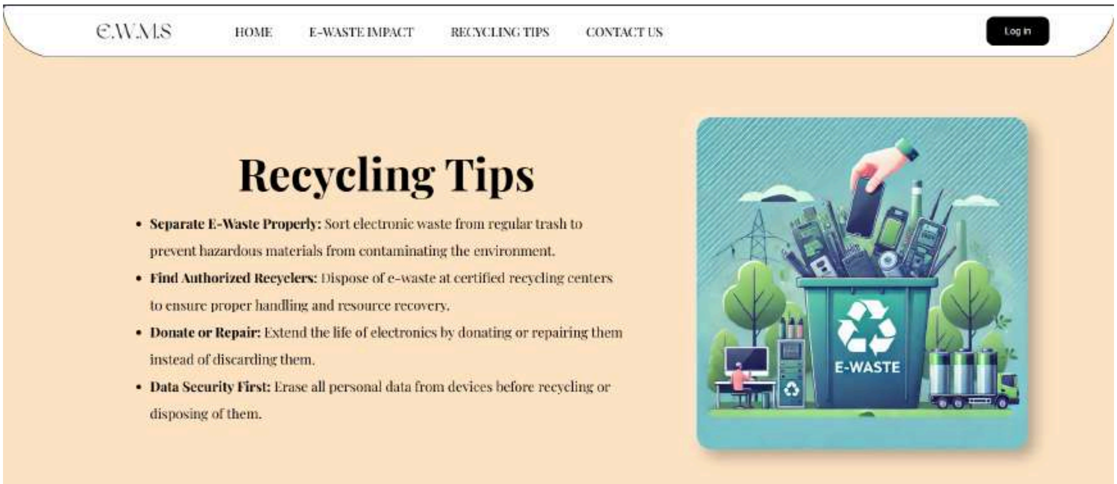


E-Waste Impact

Over 50 million metric tons of e-waste are generated globally every year, with only 17.4% being recycled properly. E-waste contains toxic substances like lead and mercury, causing severe environmental and health issues. Additionally, valuable materials worth $57 billion annually, such as gold and copper, are lost due to improper recycling. Addressing this issue is crucial for sustainability and reducing the carbon footprint.

The informal recycling sector in many developing countries worsens health risks for workers. E-waste accounts for 70% of hazardous waste in landfills, leaking toxic chemicals into soil and water. Proper management of e-waste can conserve natural resources and reduce energy consumption. Public awareness and stricter enforcement of regulations are essential for progress. Governments, corporations, and individuals must collaborate to transition toward a circular economy for electronic devices.

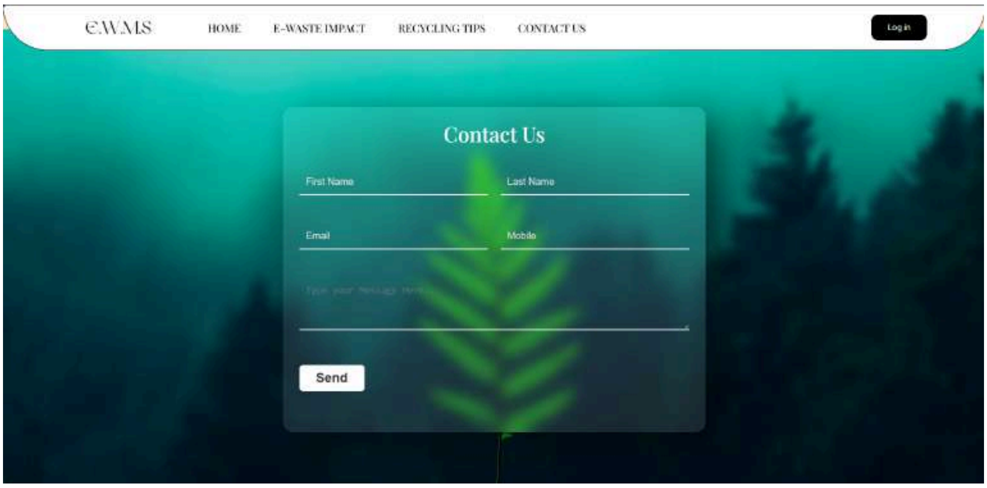
Fig - 2: Landing Page - 2



Recycling Tips

- **Separate E-Waste Properly:** Sort electronic waste from regular trash to prevent hazardous materials from contaminating the environment.
- **Find Authorized Recyclers:** Dispose of e-waste at certified recycling centers to ensure proper handling and resource recovery.
- **Donate or Repair:** Extend the life of electronics by donating or repairing them instead of discarding them.
- **Data Security First:** Erase all personal data from devices before recycling or disposing of them.

Fig - 3: Landing Page - 3



Contact Us

First Name

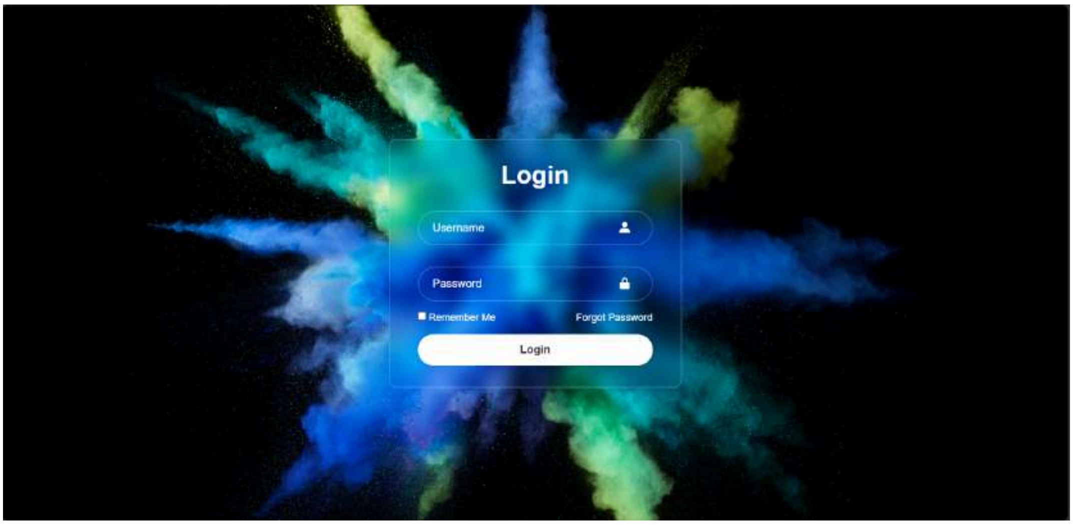
Last Name

Email

Mobile

Type your message here

Fig - 4: Landing Page - 4



Login

Username

Password

Fig - 5: Login Page

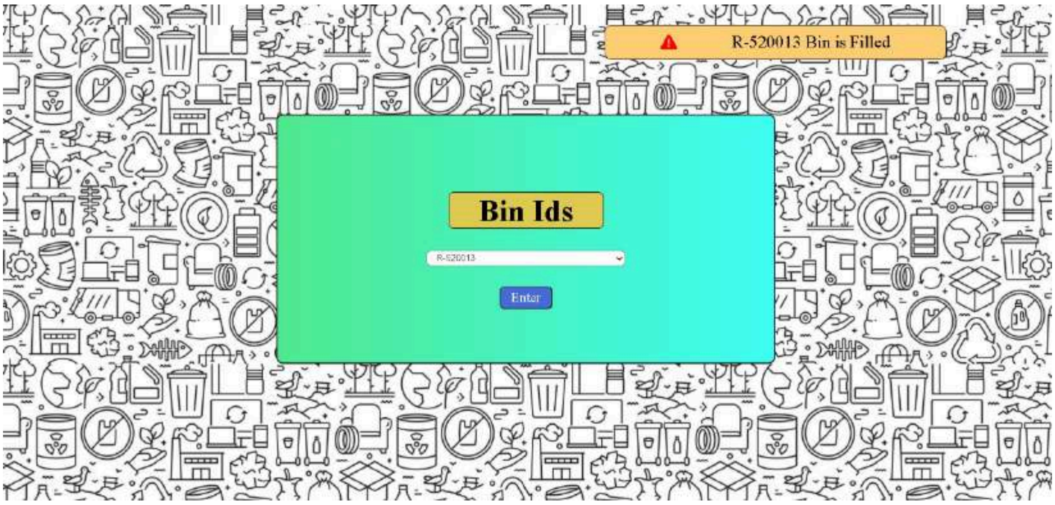


Fig - 6: Tracker

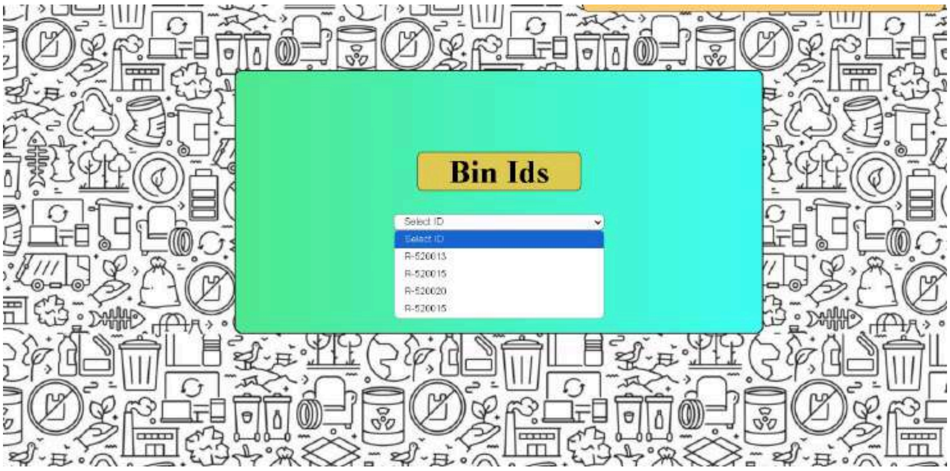
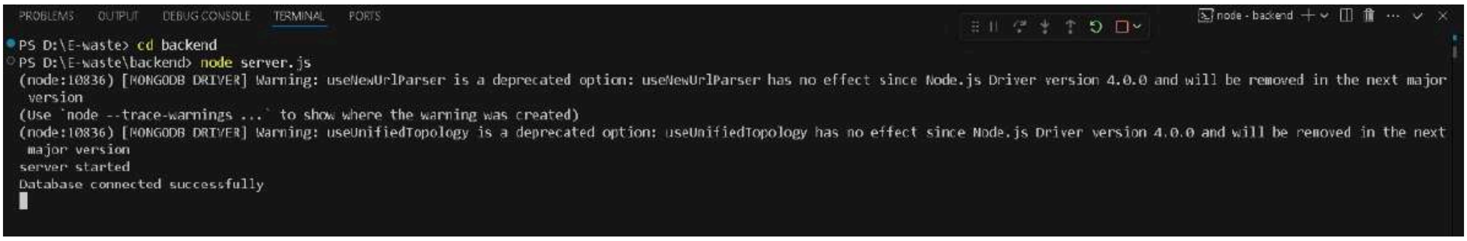
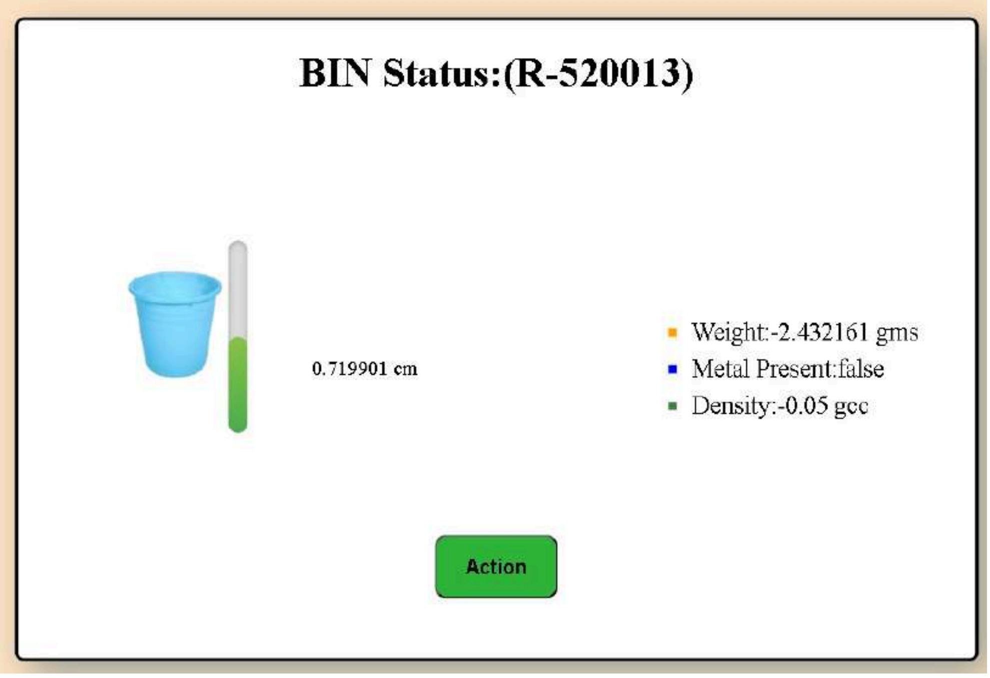


Fig - 7: Tracker With Bin IDs



```
PS D:\E-waste> cd backend
PS D:\E-waste\backend> node server.js
(node:10836) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major
 version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:10836) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next
 major version
server started
Database connected successfully
```

Fig - 8: Backend Code Running Successfully (includes Server, Database)

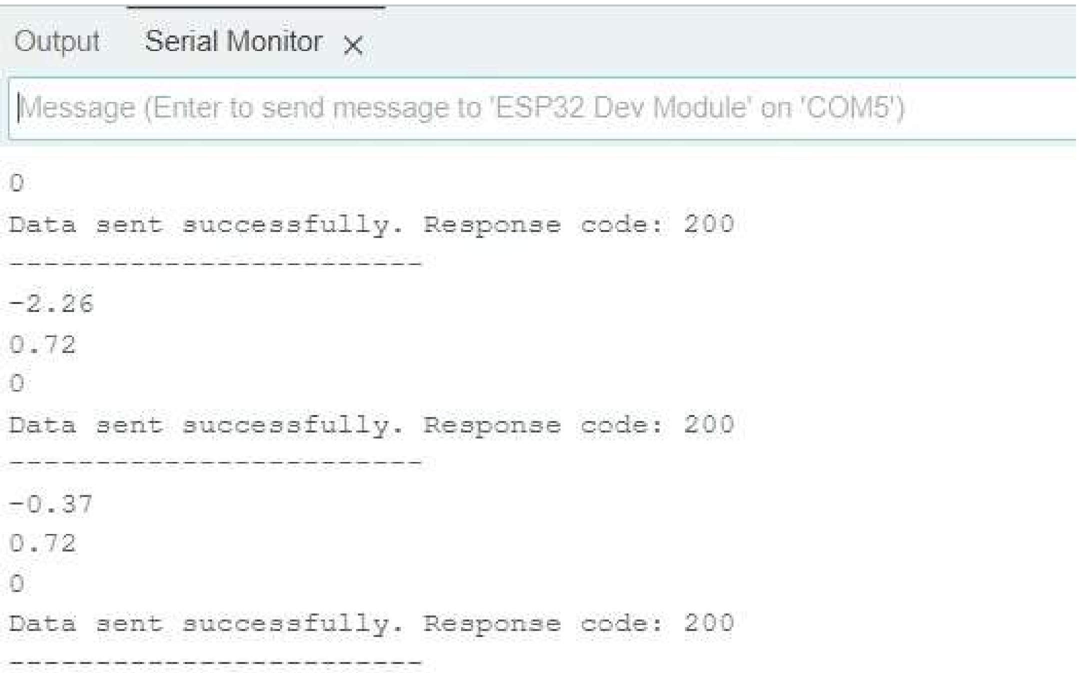


BIN Status:(R-520013)

0.719901 cm

- 🟧 Weight:-2.432161 gms
- 🟦 Metal Present:false
- 🟩 Density:-0.05 gcc

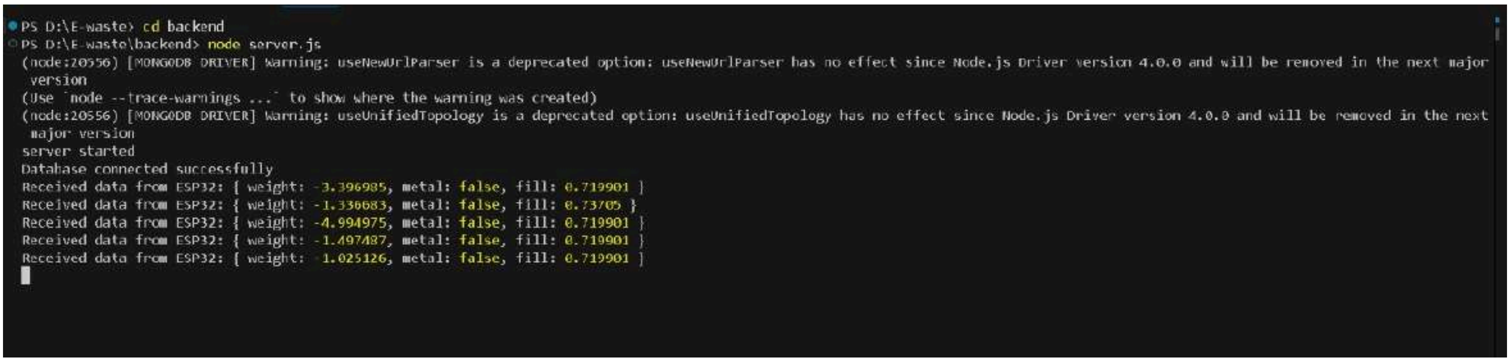
Fig - 9 : Bin Status



Output
Serial Monitor ×
Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')

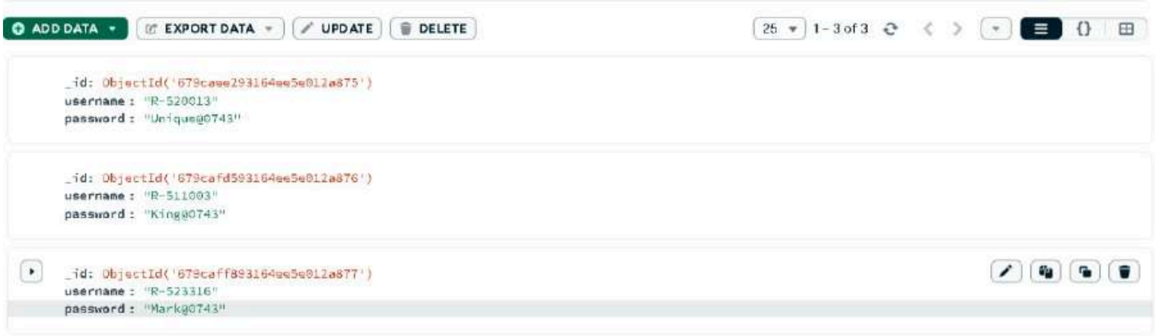
```
0
Data sent successfully. Response code: 200
------------------------
-2.26
0.72
0
Data sent successfully. Response code: 200
------------------------
-0.37
0.72
0
Data sent successfully. Response code: 200
------------------------
```

Fig - 10: Live Bin Status (Weight, Fill Level, Metal Detection)



```
PS D:\E-waste> cd backend
PS D:\E-waste\backend> node server.js
(node:20556) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major
 version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:20556) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next
 major version
server started
Database connected successfully
Received data from ESP32: { weight: -3.396985, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.336683, metal: false, fill: 0.73705 }
Received data from ESP32: { weight: -4.994975, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.497487, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.025126, metal: false, fill: 0.719901 }
```

Fig - 11: Live Bin Status From Terminal



```
_id: ObjectId('679caee293164ee5e012a875')
username : "R-520013"
password : "Unique@0743"

_id: ObjectId('679cafd593164ee5e012a876')
username : "R-511003"
password : "King@0743"

_id: ObjectId('679caff893164ee5e012a877')
username : "R-523316"
password : "Mark@0743"
```

Fig - 12: MongoDB

## 3.2 USER WORKFLOW

The two main roles in the program are admin and user.

- In addition to navigating the interface and reporting the kind and amount of e-waste they own, users register or log in to the system. Each submission consists of uploading optional information (such as amount or brief remarks) and choosing from predefined categories (such as batteries, old phones, and chargers) [6].
- Following submission, the request is received by the administrator, who classifies the report and, following examination, updates the database's status.
- Additionally, administrators may manually export information for external reporting and see analytics (such as the volume of e-waste collected by category or area) [7].

## 3.3 DATABASE DESIGN

Important entities are stored in the MongoDB database, including:

- Users: userID, name, email, role, password (hashed)
- E-Waste Records: submissionID, userID, category, timestamp, status
- Admin Logs: adminID, actionType, timestamp, notes

Indexing was applied to userID and timestamp fields to enhance query performance. Relationships between users and submissions were maintained through referencing in MongoDB's flexible schema [8].

## 3.4 TESTING PROCEDURES

The system was tested in many stages:

- Unit Testing: Using Jest and human inspection, important backend features (such as database insertion and login validation) were examined separately.
- Integration Testing: Tests were conducted that integrated backend validation and data storage with frontend submission flow testing.
- Usability Testing: To identify usability problems or make recommendations for enhancements, peer users tested the interface.
- Data Consistency: To ensure that queries behaved consistently over several submissions, tests were conducted [11].

Localized manual testing made sure that the program was reliable and free of bugs under anticipated load circumstances, even if extensive testing with outside users was not practical.

## 3.5 SCALABILITY CONSIDERATIONS

The system architecture is built to scale in the future, even though the present deployment is offline and hosted on a local computer. The backend may be deployed using services like Heroku, Railway, or Render with very minor changes. The database may be moved to MongoDB services hosted in the cloud, such as Atlas. Future groups or organizations can implement the technology for practical usage thanks to its adaptability [13].

## 3.6 ENVIRONMENT SETUP AND USAGE

The environment setup includes:

- Installing Node.js and MongoDB locally
- Running the backend server with Express
- Opening the HTML frontend in a browser
- Connecting frontend forms to backend for submission

All system components are modularized, and README documentation is provided to help users and administrators understand usage instructions.

## 3.7 SUMMARY

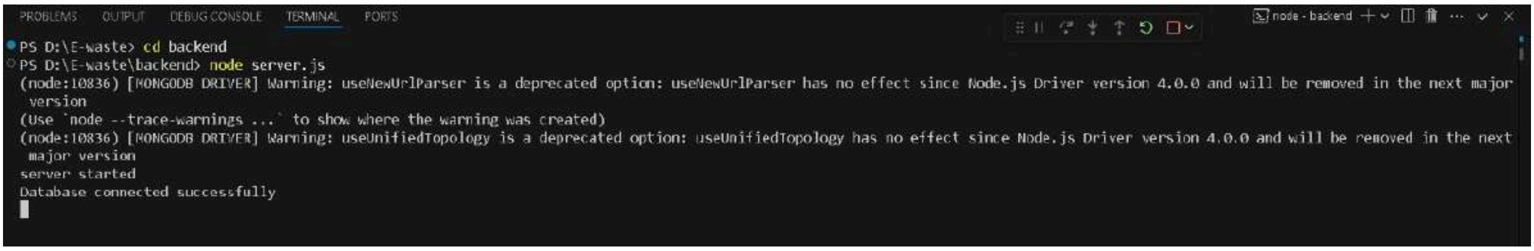
The E-Waste Management System's methodology reflects a systematic approach to software engineering — from planning and design through implementation and testing. It emphasizes open-source technologies, robust database modeling, user-centric workflow, and secure backend logic. The methodology ensures the system is reliable, adaptable, and aligned with environmental objectives [14].

Chapter 4

# Results and Discussion

The E-Waste Management System (EWMS) was created as a locally hosted, browser-accessible web interface with JavaScript, HTML, and CSS. MongoDB was used as the database to hold user records, interaction logs, and e-waste bin data, while Node.js and Express were used to build the backend.

Based on both simulated and real-time inputs from the system, this chapter offers insights about the EWMS's functionality, performance, and user interface behavior. For clarification, screenshots of the backend terminal and website interface are supplied.



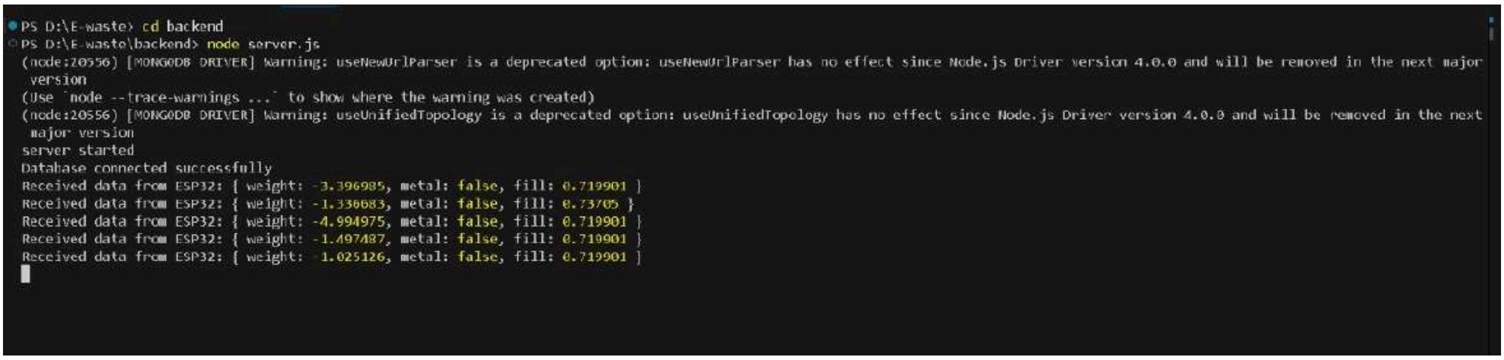
```
PS D:\E-waste> cd backend
PS D:\E-waste\backend> node server.js
(node:10836) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major
 version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:10836) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next
 major version
server started
Database connected successfully
```

Fig - 13: Locally running Node.js backend server

## 4.1 BACKEND PROCESSING

Express and Node.js were used to develop the backend server. It emulated real-time bin status updates and managed the main application logic. Bin data was locally stored in MongoDB. Every time the website was updated, the bin status updates were shown as logs in the terminal instead of through Postman or APIs.

In order to replicate a real-world data flow from a sensor or microcontroller, each refresh initiated a backend call to update or obtain the current condition of the e-waste bin. To simulate real sensor input, false data was manually entered into the backend logic for demonstration.

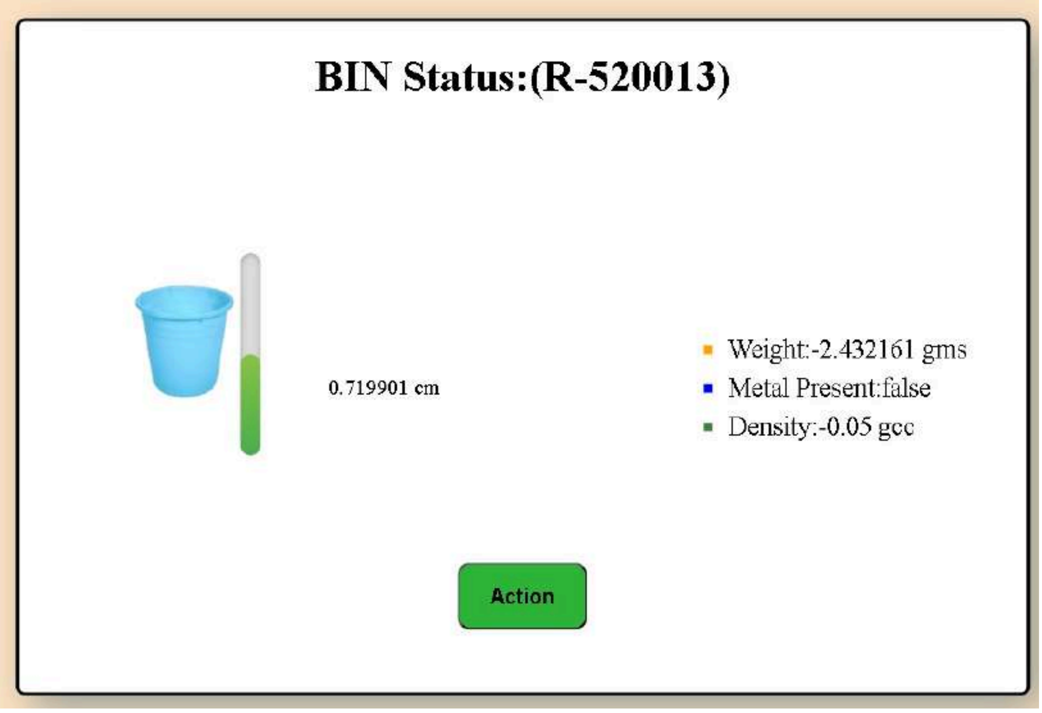


```
PS D:\E-waste> cd backend
PS D:\E-waste\backend> node server.js
(node:20556) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major
 version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:20556) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next
 major version
server started
Database connected successfully
Received data from ESP32: { weight: -3.396985, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.330683, metal: false, fill: 0.73705 }
Received data from ESP32: { weight: -4.994975, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.497487, metal: false, fill: 0.719901 }
Received data from ESP32: { weight: -1.025126, metal: false, fill: 0.719901 }
```

Fig - 14: Terminal output displaying live bin status update

## 4.2 LIVE BIN STATUS TRACKING

The system was able to replicate real-time bin fill-level changes while lacking GPS capabilities. The updated bin status (such as empty, half-full, or full) was obtained from the backend and shown appropriately on the interface or as a console message each time the user reloaded the page.
Preset values that changed according to refresh cycles were used in the live tracking simulation. This method effectively illustrates the bin's capacity to notify a central system of its state.



BIN Status:(R-520013)

0.719901 cm

- Weight:-2.432161 gms
- Metal Present:false
- Density:-0.05 gcc

Fig - 15: Live Bin Status from Website

In conclusion, the E-Waste Management System successfully illustrated its expected functionality in a limited development setting. Through constant data updates shown in the terminal and on the local frontend interface, the system successfully simulated real-time monitoring even though it does not make use of GPS tracking or live web hosting. To guarantee constant synchronization between hardware inputs and digital outputs, the updated bin status is retrieved from the locally hosted backend each time the page is refreshed. Accurate bin status was reflected in real-time thanks to the effective integration of the HTML-based web interface with the hardware components (such as sensors coupled to NodeMCU). This demonstrates that the system is functionally viable for prototype or small-scale deployments. A functional feedback loop was created by the backend operations, which processed sensor data and effectively replied to frontend requests while being restricted to local execution. Through the use of straightforward frontend technologies and fundamental IoT ideas, the project verifies the viability of implementing an inexpensive, readily replicated trash management prototype. It establishes the foundation for further improvements including database integration, cloud-based deployment, GPS-enabled tracking, and user interface modules. This technology has the potential to grow further into a more reliable and scalable solution for smart city e-waste problems.

## Chapter 5

# Conclusion and Future work

The project's **E-Waste Management System (EWMS)** effectively meets the increasing need for electronic waste monitoring and appropriate disposal. The solution mimics a functional model of real-time bin monitoring by combining straightforward yet efficient IoT components with a locally hosted web interface created using HTML. The NodeMCU microcontroller transmits sensor data, which is then precisely shown on an intuitive local webpage. This guarantees that updated bin statuses are available whenever the page is reloaded.

The system demonstrated its promise for small-scale, campus-level, or municipal waste management applications by operating dependably inside the local network even in the absence of GPS modules or external hosting services. The capacity of the user to monitor and react to bin consumption dynamically was further improved by data visualization on both the frontend and the terminal. By encouraging appropriate e-waste segregation and monitoring, this method supports global sustainability goals [3], [7].

The project functions as a working prototype to show how IoT may be applied in actual environmental management scenarios. Because of its simplicity and adaptability, it may be easily expanded upon in the future.

## FUTURE WORK

1. Cloud hosting and Remote access
   By putting the frontend and backend on a cloud platform like AWS or Firebase, you can monitor numerous bins deployed in various locations centrally and access them from anywhere [8], [19].

2. GPS Integration
   Geolocation information for every bin may be obtained by integrating GPS modules into the hardware design, allowing for geographical analysis and the creation of efficient collection routes for municipal garbage services [5, 14].

3. Database Connectivity
   In order to retain history bin usage, provide analytical reports, and analyze patterns over time, future versions may incorporate a real-time database (such as Firebase Realtime Database or MongoDB) [4], [13].

4. Notification and Alert System
   In order to facilitate timely garbage collection and reduce overflow situations, the system may be set up to send messages or alerts (by SMS or email) when a bin exceeds its maximum capacity. [10], [15].

5. Integration with Municipal Platforms
   EWMS can develop into a scalable and dependable public infrastructure utility by integrating with current smart city platforms or APIs utilized by municipal organizations [6], [21].

6. Admin and User Dashboards
   Frameworks like React or Angular may be used to create a more complete online dashboard that offers data analytics and visualization capabilities in addition to role-based access for administrators and users [11], [23].

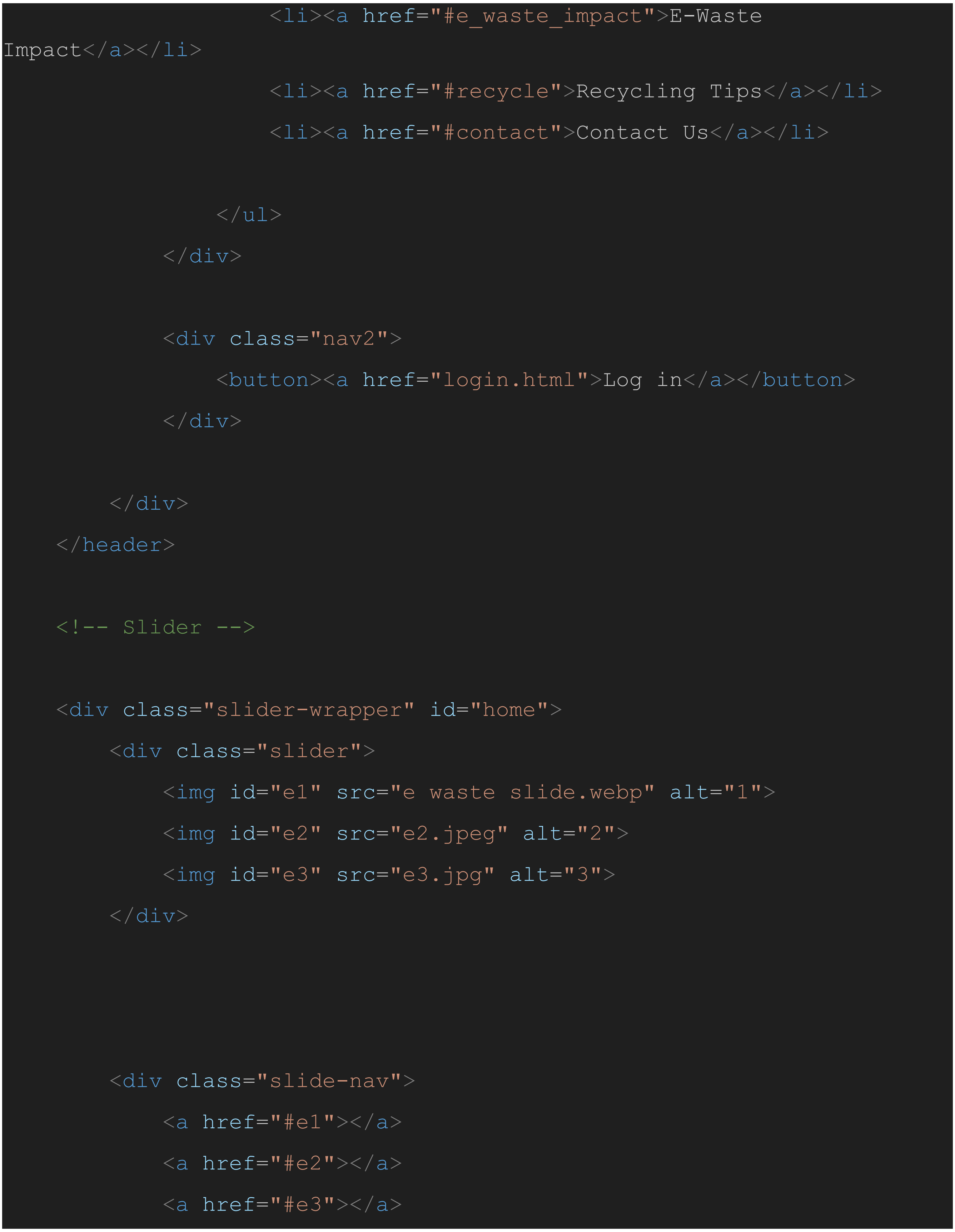
7. Machine Learning for Predictive Monitoring
   Predictive models that foresee trash bin filling tendencies and recommend proactive waste collection schedules may be constructed using machine learning algorithms and data gathered over time [9], [16].

# Appendices

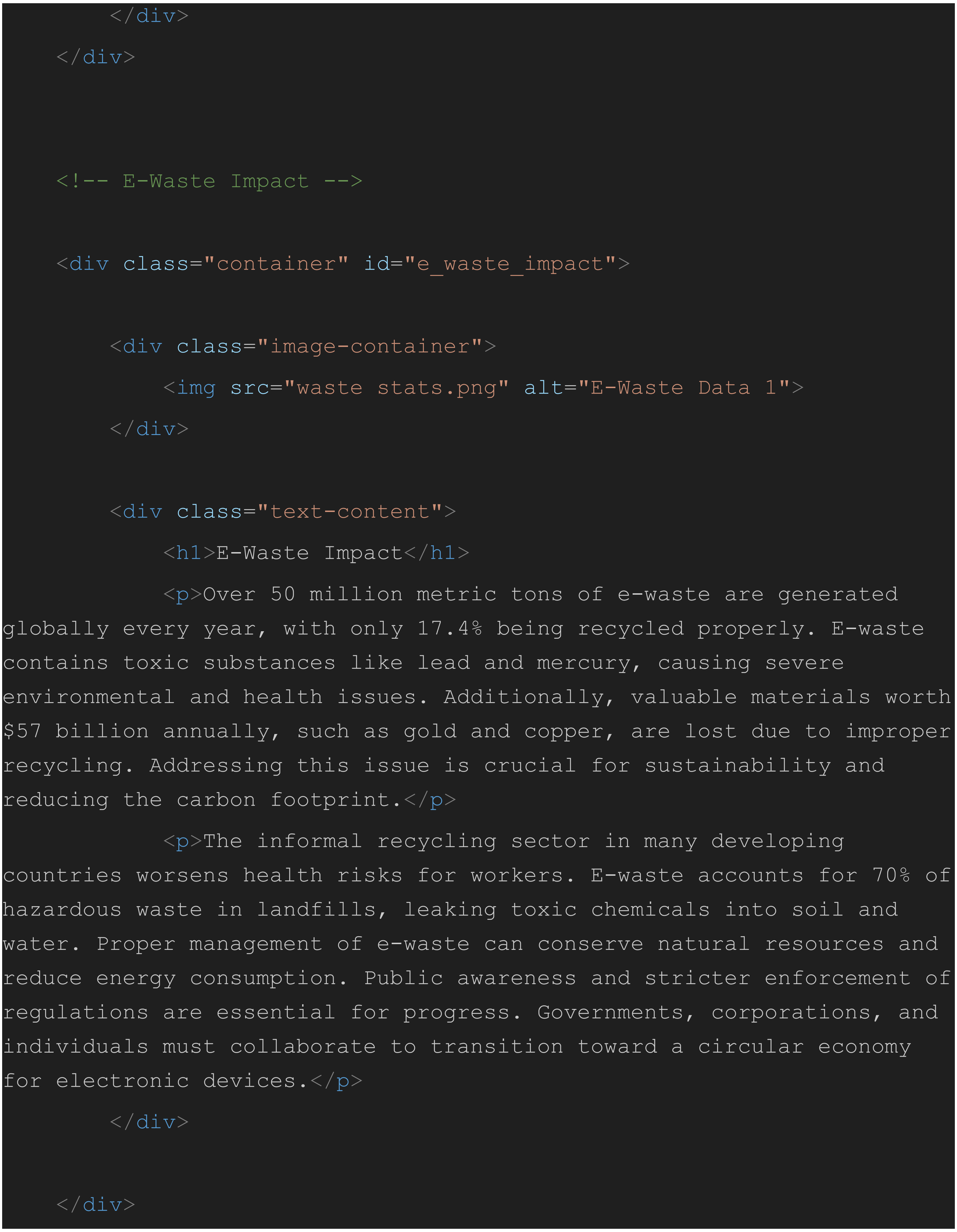
## Frontend:

Appendix 1: ( index.html )

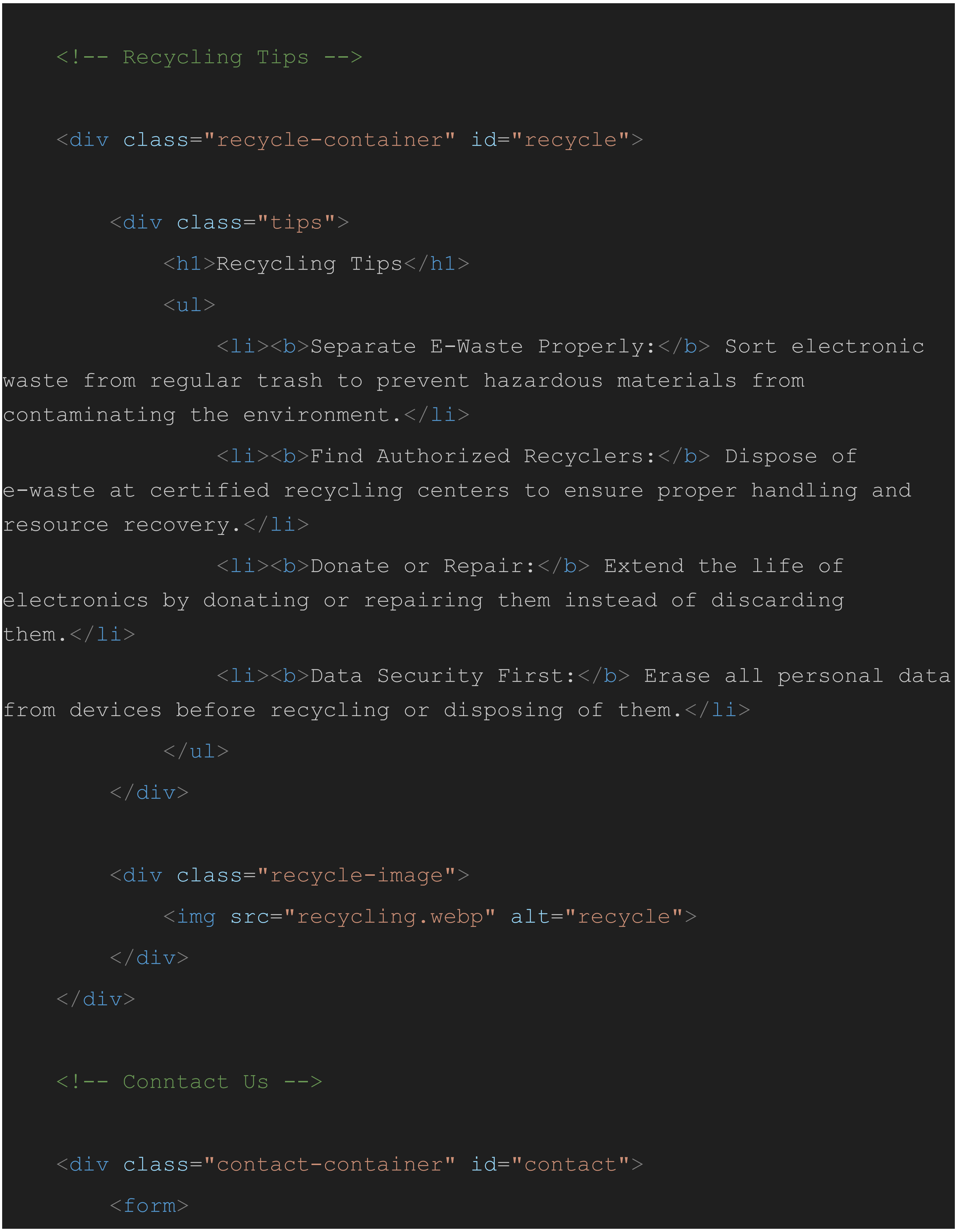
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" href="EWMS.png" type="image/x-icon">
    <!-- Google Fonts -->
    <link
href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital,wght@0,400..900;1,400..900&family=Playwrite+IS&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="style.css">
    <title>EWMS.</title>
</head>

<body>
    <header>

        <!-- Navigation Bar -->
        <div class="navigator">
            <div class="nav-logo">
                <img src="EWMS.png" alt="ewms">
            </div>

            <div class="nav1">
                <ul>
                    <li><a href="#home">Home</a></li>
```



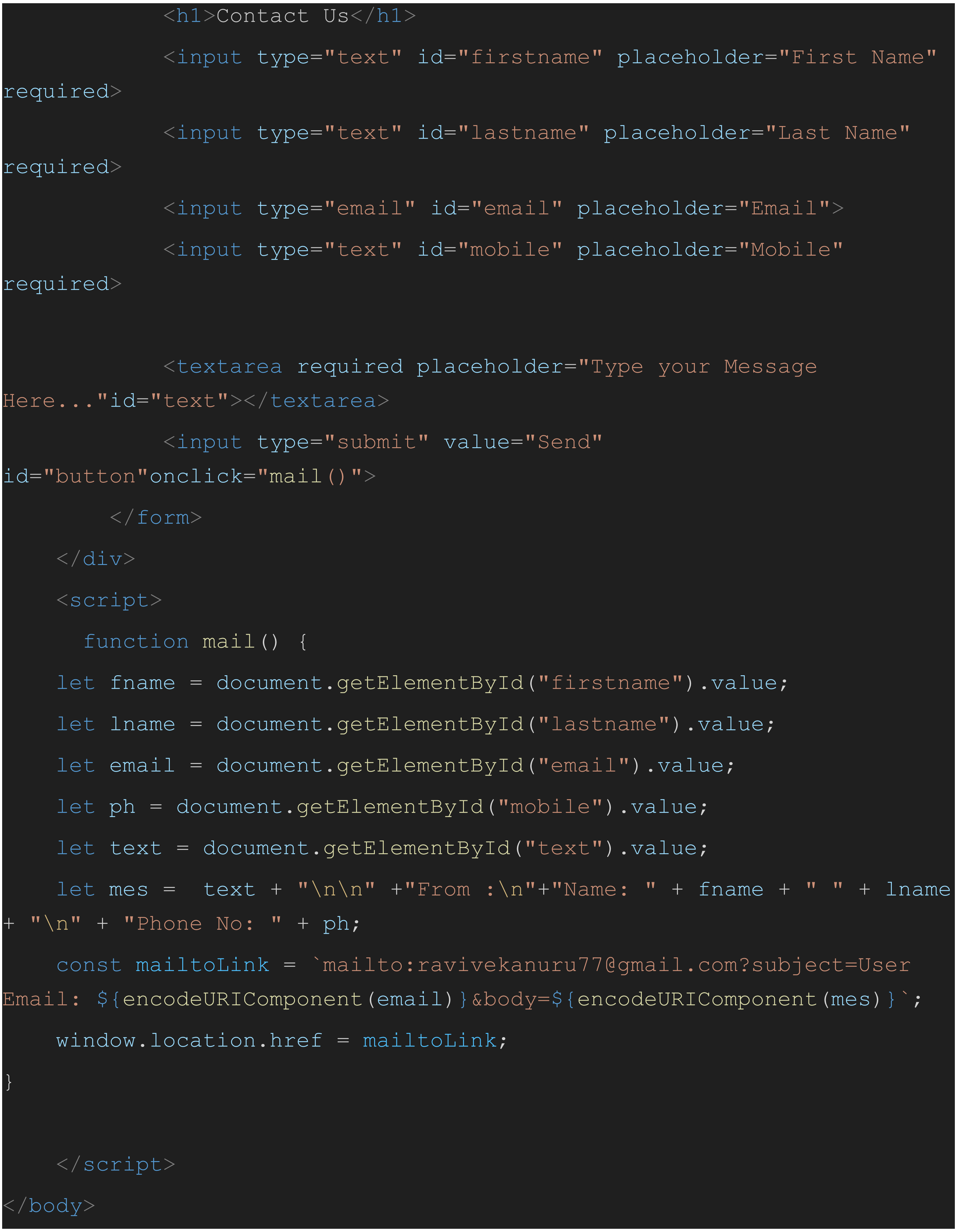
```
                    <li><a href="#e_waste_impact">E-Waste
Impact</a></li>
                    <li><a href="#recycle">Recycling Tips</a></li>
                    <li><a href="#contact">Contact Us</a></li>

                </ul>
            </div>

            <div class="nav2">
                <button><a href="login.html">Log in</a></button>
            </div>

        </div>
    </header>

    <!-- Slider -->

    <div class="slider-wrapper" id="home">
        <div class="slider">
            <img id="e1" src="e waste slide.webp" alt="1">
            <img id="e2" src="e2.jpeg" alt="2">
            <img id="e3" src="e3.jpg" alt="3">
        </div>



        <div class="slide-nav">
            <a href="#e1"></a>
            <a href="#e2"></a>
            <a href="#e3"></a>
```



```
        </div>
    </div>


    <!-- E-Waste Impact -->

    <div class="container" id="e_waste_impact">

        <div class="image-container">
            <img src="waste stats.png" alt="E-Waste Data 1">
        </div>

        <div class="text-content">
            <h1>E-Waste Impact</h1>
            <p>Over 50 million metric tons of e-waste are generated globally every year, with only 17.4% being recycled properly. E-waste contains toxic substances like lead and mercury, causing severe environmental and health issues. Additionally, valuable materials worth $57 billion annually, such as gold and copper, are lost due to improper recycling. Addressing this issue is crucial for sustainability and reducing the carbon footprint.</p>
            <p>The informal recycling sector in many developing countries worsens health risks for workers. E-waste accounts for 70% of hazardous waste in landfills, leaking toxic chemicals into soil and water. Proper management of e-waste can conserve natural resources and reduce energy consumption. Public awareness and stricter enforcement of regulations are essential for progress. Governments, corporations, and individuals must collaborate to transition toward a circular economy for electronic devices.</p>
        </div>

    </div>
```



```
    <!-- Recycling Tips -->

    <div class="recycle-container" id="recycle">

        <div class="tips">
            <h1>Recycling Tips</h1>
            <ul>
                <li><b>Separate E-Waste Properly:</b> Sort electronic waste from regular trash to prevent hazardous materials from contaminating the environment.</li>
                <li><b>Find Authorized Recyclers:</b> Dispose of e-waste at certified recycling centers to ensure proper handling and resource recovery.</li>
                <li><b>Donate or Repair:</b> Extend the life of electronics by donating or repairing them instead of discarding them.</li>
                <li><b>Data Security First:</b> Erase all personal data from devices before recycling or disposing of them.</li>
            </ul>
        </div>

        <div class="recycle-image">
            <img src="recycling.webp" alt="recycle">
        </div>
    </div>

    <!-- Conntact Us -->

    <div class="contact-container" id="contact">
        <form>
```

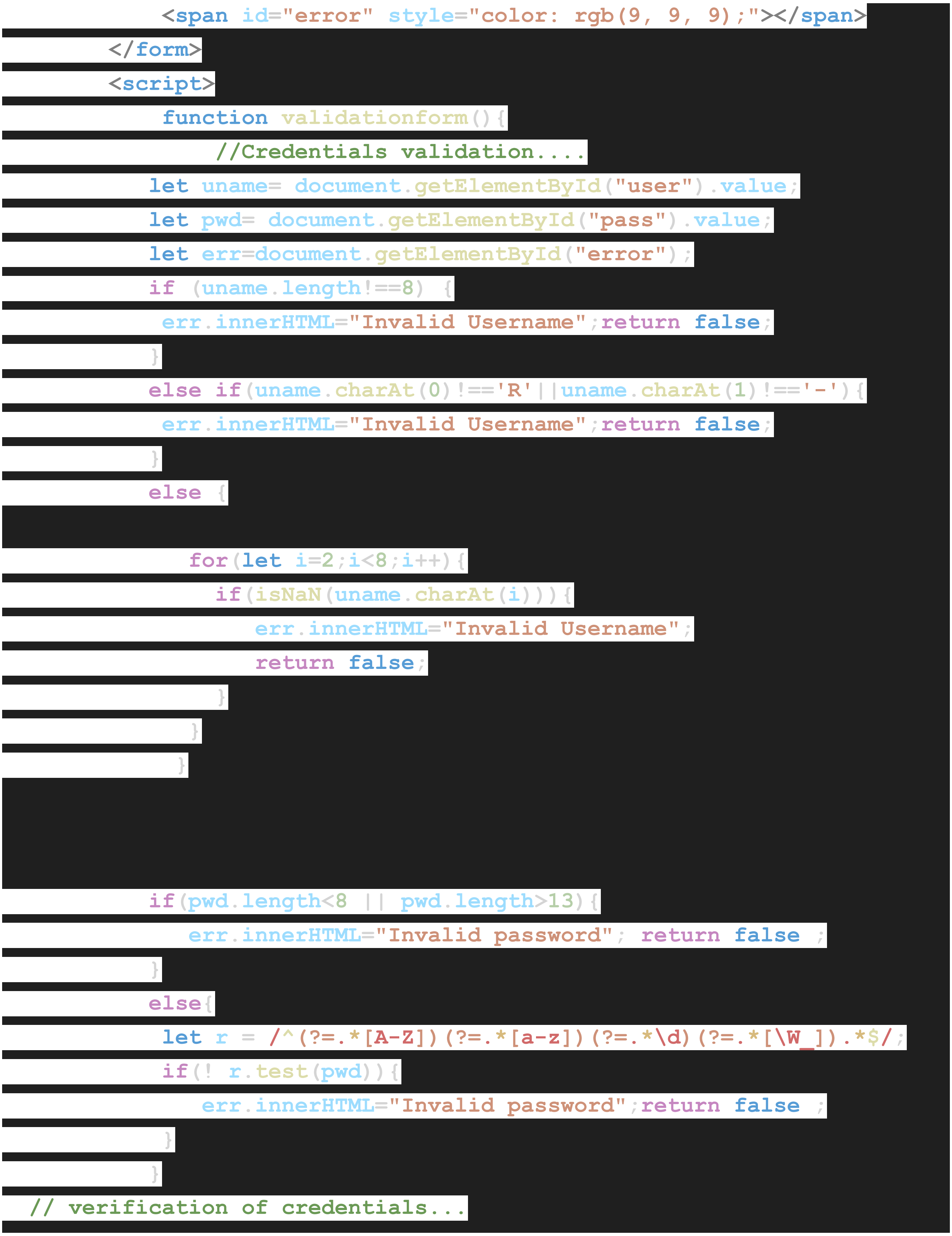


```
            <h1>Contact Us</h1>
            <input type="text" id="firstname" placeholder="First Name"
required>
            <input type="text" id="lastname" placeholder="Last Name"
required>
            <input type="email" id="email" placeholder="Email">
            <input type="text" id="mobile" placeholder="Mobile"
required>

            <textarea required placeholder="Type your Message
Here..."id="text"></textarea>
            <input type="submit" value="Send"
id="button"onclick="mail()">
        </form>
    </div>
    <script>
      function mail() {
    let fname = document.getElementById("firstname").value;
    let lname = document.getElementById("lastname").value;
    let email = document.getElementById("email").value;
    let ph = document.getElementById("mobile").value;
    let text = document.getElementById("text").value;
    let mes =  text + "\n\n" +"From :\n"+"Name: " + fname + " " + lname
+ "\n" + "Phone No: " + ph;
    const mailtoLink = `mailto:ravivekanuru77@gmail.com?subject=User
Email: ${encodeURIComponent(email)}&body=${encodeURIComponent(mes)}`;
    window.location.href = mailtoLink;
}

    </script>
</body>
```

```
</html>
```

Appendix 2: ( login.html )

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="style1.css">
    <link href='https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css'
rel='stylesheet'>
    <title>EWMS Login</title>
</head>
<body>
    <div class="wrapper">
        <form onsubmit="return validationform()">
            <h1>Login</h1>

            <div class="input-box">
                <input type="text" id="user" placeholder="Username"
required>
                <i class='bx bxs-user' style='color:#ffffff' ></i>
            </div>
            <div class="input-box">
                <input type="password"id="pass" placeholder="Password"
required>
                <i class='bx bxs-lock-alt'></i>
            </div>
            <div class="remember-forgot">
                <label><input type="checkbox">Remember Me</label>
                <a href="#">Forgot Password</a>
            </div>
            <button type="submit" class="btn">Login</button>
            <br>
```



```
            <span id="error" style="color: rgb(9, 9, 9);"></span>
        </form>
        <script>
            function validationform(){
                //Credentials validation....
           let uname= document.getElementById("user").value;
           let pwd= document.getElementById("pass").value;
           let err=document.getElementById("error");
           if (uname.length!==8)  {
            err.innerHTML="Invalid Username";return false;
           }
           else if(uname.charAt(0)!=='R'||uname.charAt(1)!=='-'){
            err.innerHTML="Invalid Username";return false;
           }
           else  {

              for(let i=2;i<8;i++){
                if(isNaN(uname.charAt(i))){
                   err.innerHTML="Invalid Username";
                   return false;
                }
              }
             }



           if(pwd.length<8  || pwd.length>13){
              err.innerHTML="Invalid password"; return false ;
           }
           else{
            let r = /^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[\W_]).*$/;
            if(! r.test(pwd)){
               err.innerHTML="Invalid password";return false ;
            }
           }
  // verification of credentials...
  fetch('http://localhost:3000/login', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json',
                    },
                    body: JSON.stringify({ username: uname, password: pwd })
                })

                // processing response...
                .then(response => response.json())
                .then(data => {
                    if (data.message === 'Login successful') {
                        let x=uname;let y="No";
                        if(data.fill>20.0){
                            y=uname;
                        }
                        window.location.href = `index2.html?fill=${y}`;
                    } else {
                        err.innerHTML = data.message;
                    }
                })

                // error handling...
                .catch(error => {
                    console.error('Error:', error);
                    err.innerHTML = "Error connecting to the server";
                });
            //fetch request for retrieving fill level...

                return false;
            }
        </script>
    </div>
</body>
```

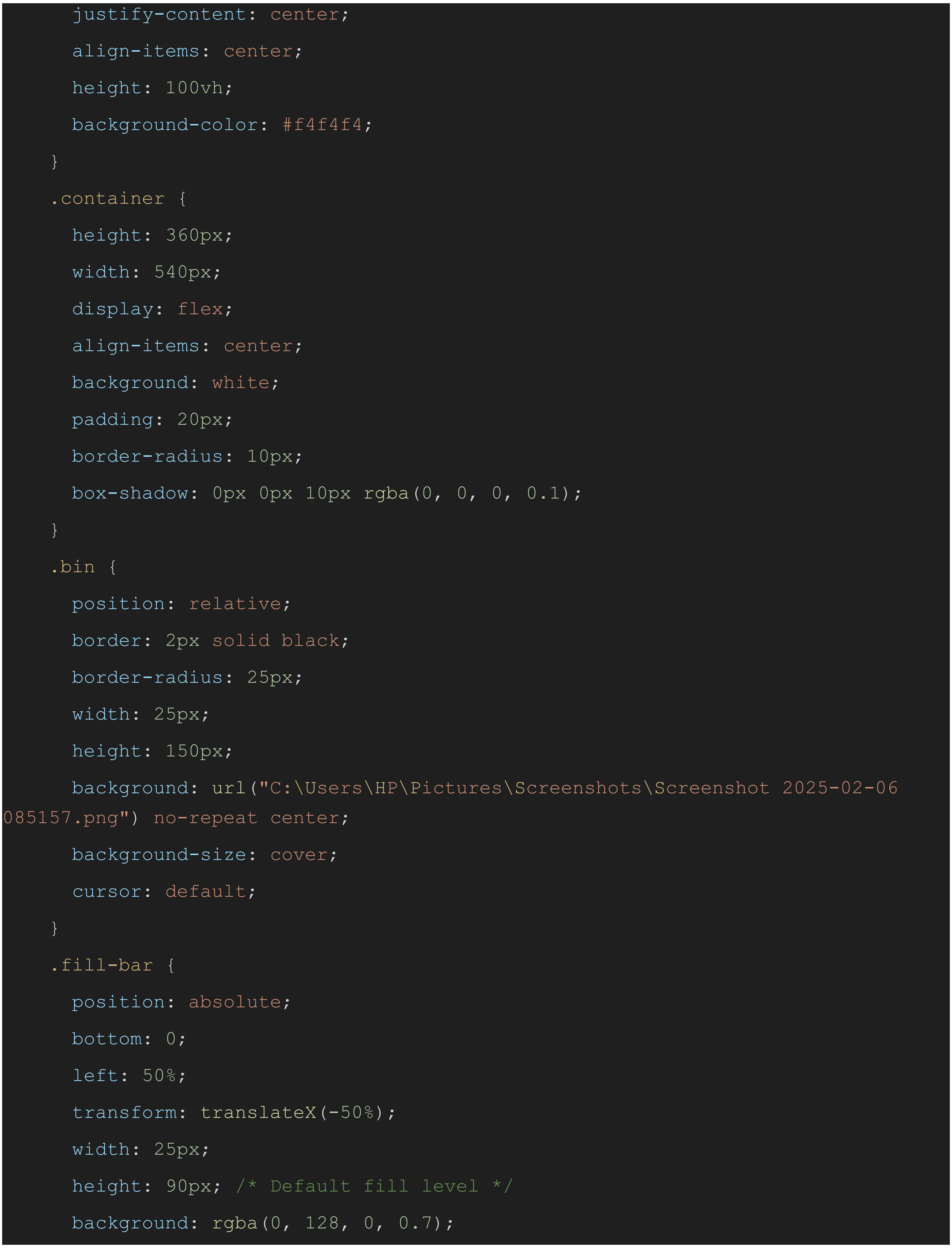
```
</html>
```

Appendix 3: ( dashboard.html )

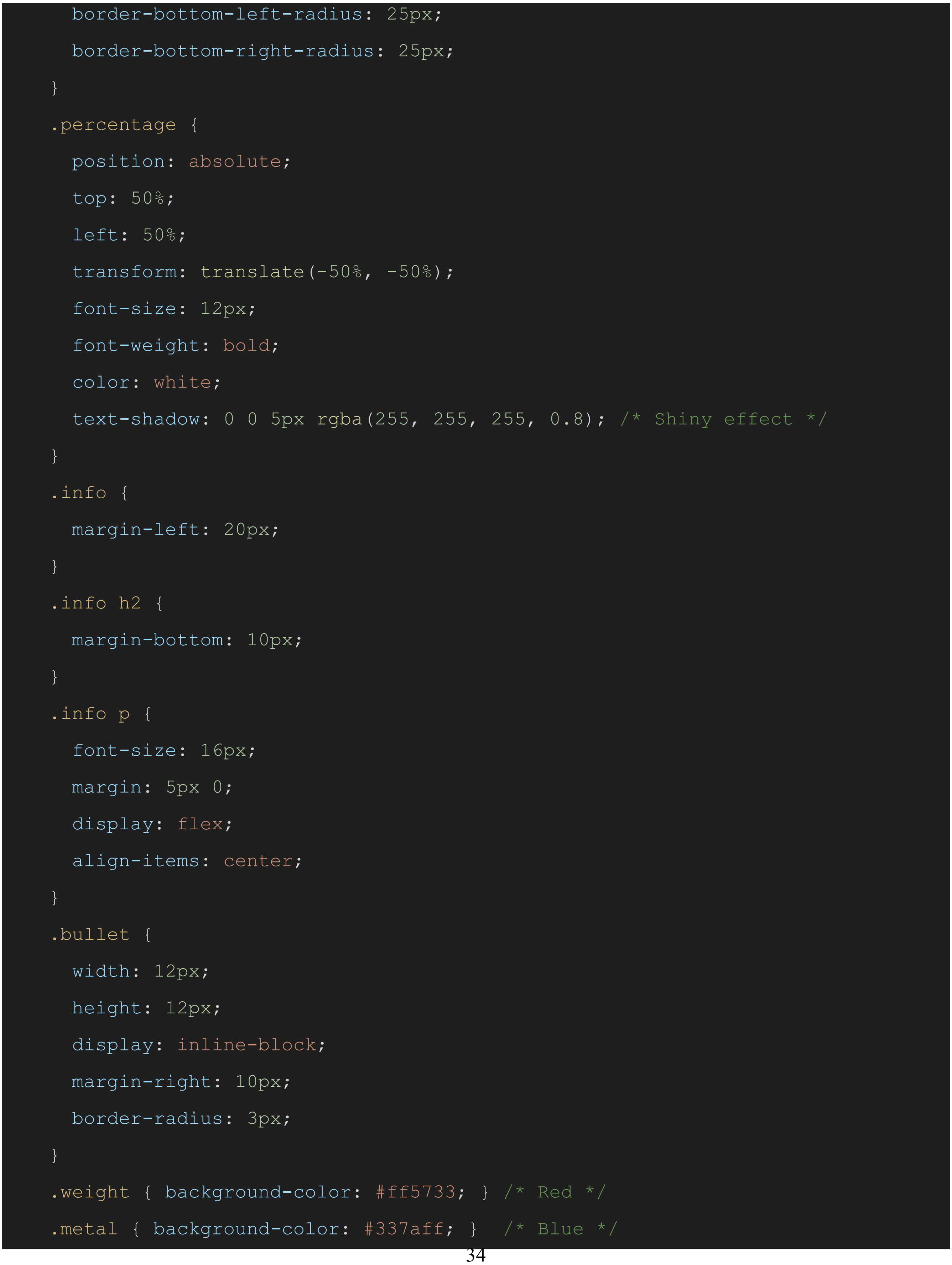
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.1/css/all.min.css">
    <link rel="stylesheet" href="style2.css" />
    <title>Dashboard</title>
   <script>
     document.addEventListener("DOMContentLoaded", function() {
                function getQueryParam(name) {
                    const urlParams = new
URLSearchParams(window.location.search);
                    return urlParams.get(name);
                }
                const id=getQueryParam("fill");
                const x=document.getElementsByTagName("h1")[0];
             x.innerText =id+" Bin is Filled";

              });
 function redirectToTrack() {
  const selectBox = document.querySelector(".search-select");
  const selectedId = selectBox.value;

 //let fetchedData = { fill: 0.0, metal: false };
 //const fill=0.0;
  fetch('http://localhost:3000/index2')
    .then(response => response.json())
    .then(data => {
     const fill=data.fill;

      const weight = data.weight;
      const metal = data.metal;
      const r=4.5;
      const vol=3.14*fill*r*r;
      const hook = weight / vol;
      const density=hook.toFixed(2);
          //density=density.toFixed(2);//calculation of volume is pending....

      if (selectedId !== "all") {
        window.location.href =
`track.html?id=${selectedId}&weight=${weight}&metal=${metal}&density=${density}&f
=${fill}`;
      } else {
        alert("Please select a Bin ID!");
      }
    })
    .catch(error => {
      console.error('Error fetching data:', error);
    });

}

</script>

  </head>
  <body>
    <div class="filler">
      <div class="message">
        <i class="fa-solid fa-triangle-exclamation alert blink"></i>
        <h1>No bin is filled</h1>
      </div>
    </div>
    <div class="box">
      <h1 class="bin">Bin Ids</h1>
      <div class="wrapper">
        <div class="nav-search">
          <select class="search-select">
            <option value="all">Select ID</option>
            <option value="R-520013">R-520013</option>
            <option value="R-520015">R-520015</option>
            <option value="R-520020">R-520020</option>
            <option value="R-520015">R-520015</option>
          </select>
        </div>
      </div>

      <div class="select-btn">
        <button class="btn1" onclick="redirectToTrack()">Enter</button>
      </div>
    </div>
  </body>
</html>
```

Appendix 4: (smart-bin-stats.html )

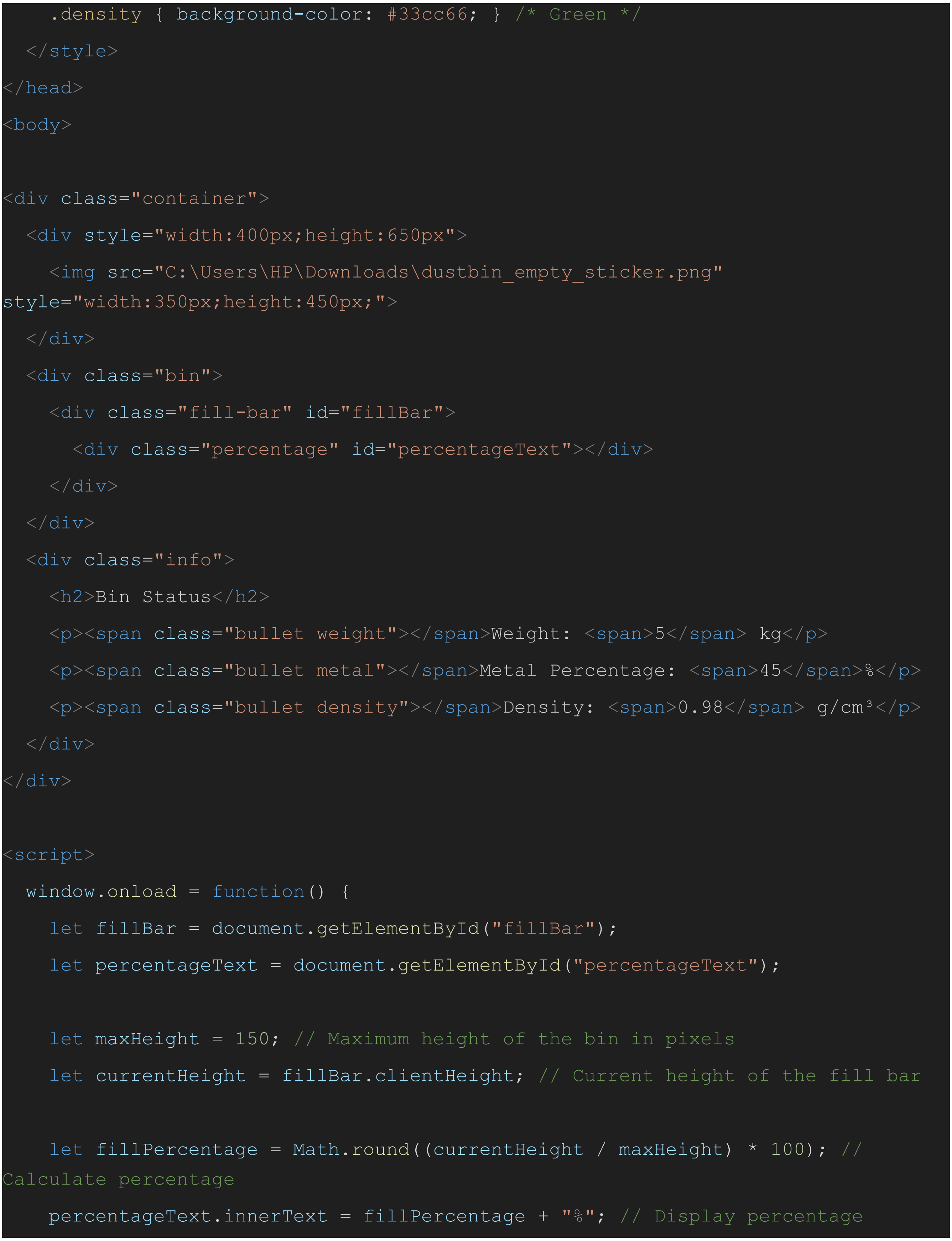
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Smart Bin Status</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      display: flex;
```



```
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f4f4f4;
    }
    .container {
      height: 360px;
      width: 540px;
      display: flex;
      align-items: center;
      background: white;
      padding: 20px;
      border-radius: 10px;
      box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    }
    .bin {
      position: relative;
      border: 2px solid black;
      border-radius: 25px;
      width: 25px;
      height: 150px;
      background: url("C:\Users\HP\Pictures\Screenshots\Screenshot 2025-02-06 085157.png") no-repeat center;
      background-size: cover;
      cursor: default;
    }
    .fill-bar {
      position: absolute;
      bottom: 0;
      left: 50%;
      transform: translateX(-50%);
      width: 25px;
      height: 90px; /* Default fill level */
      background: rgba(0, 128, 0, 0.7);
```



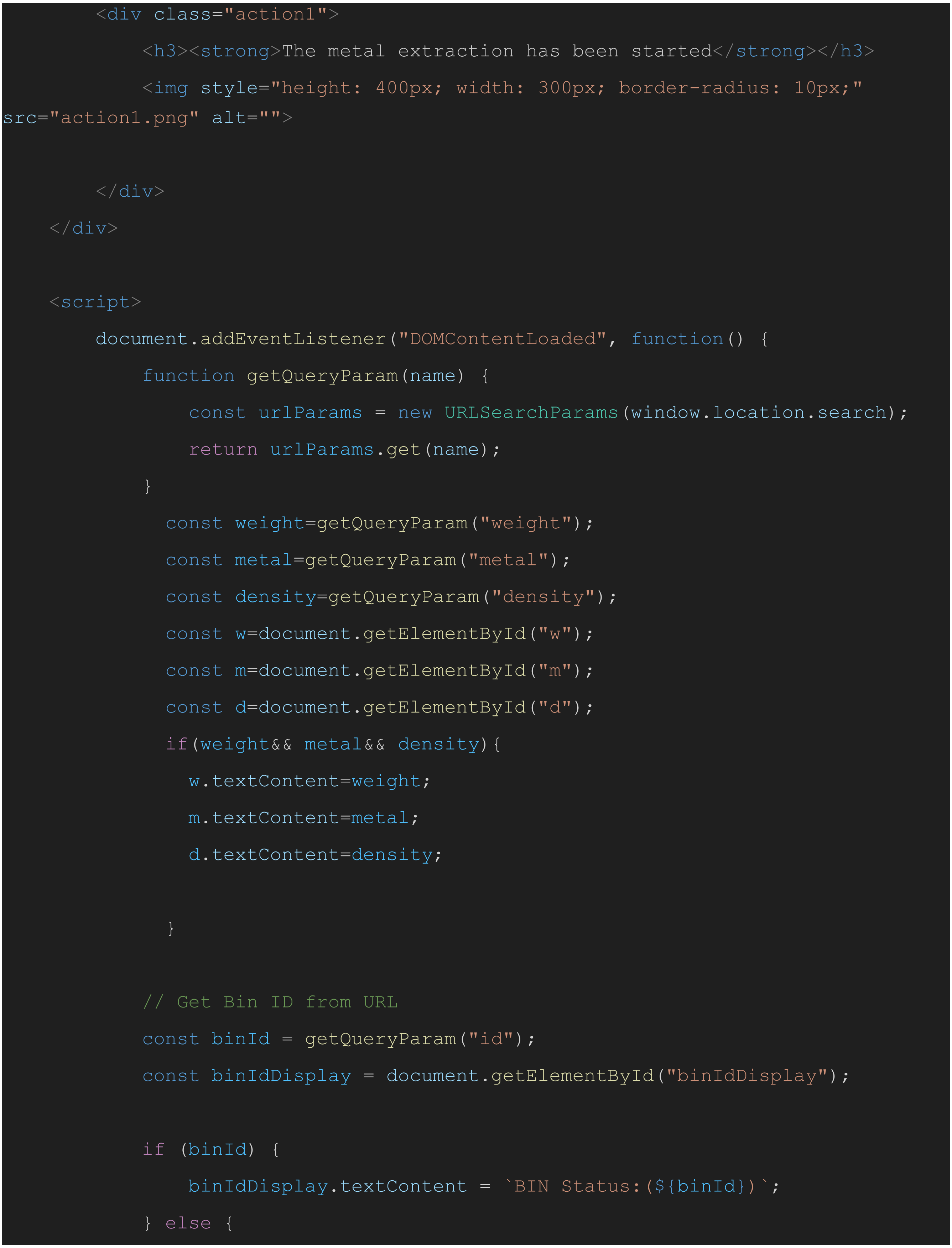
```css
    border-bottom-left-radius: 25px;
    border-bottom-right-radius: 25px;
  }
  .percentage {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    font-size: 12px;
    font-weight: bold;
    color: white;
    text-shadow: 0 0 5px rgba(255, 255, 255, 0.8); /* Shiny effect */
  }
  .info {
    margin-left: 20px;
  }
  .info h2 {
    margin-bottom: 10px;
  }
  .info p {
    font-size: 16px;
    margin: 5px 0;
    display: flex;
    align-items: center;
  }
  .bullet {
    width: 12px;
    height: 12px;
    display: inline-block;
    margin-right: 10px;
    border-radius: 3px;
  }
  .weight { background-color: #ff5733; } /* Red */
  .metal { background-color: #337aff; }  /* Blue */
```



```
    .density { background-color: #33cc66; } /* Green */
  </style>
</head>
<body>

<div class="container">
  <div style="width:400px;height:650px">
    <img src="C:\Users\HP\Downloads\dustbin_empty_sticker.png"
style="width:350px;height:450px;">
  </div>
  <div class="bin">
    <div class="fill-bar" id="fillBar">
      <div class="percentage" id="percentageText"></div>
    </div>
  </div>
  <div class="info">
    <h2>Bin Status</h2>
    <p><span class="bullet weight"></span>Weight: <span>5</span> kg</p>
    <p><span class="bullet metal"></span>Metal Percentage: <span>45</span>%</p>
    <p><span class="bullet density"></span>Density: <span>0.98</span> g/cm³</p>
  </div>
</div>

<script>
  window.onload = function() {
    let fillBar = document.getElementById("fillBar");
    let percentageText = document.getElementById("percentageText");

    let maxHeight = 150; // Maximum height of the bin in pixels
    let currentHeight = fillBar.clientHeight; // Current height of the fill bar

    let fillPercentage = Math.round((currentHeight / maxHeight) * 100); //
Calculate percentage
    percentageText.innerText = fillPercentage + "%"; // Display percentage
  }
</script>

</body>
</html>
```

Appendix 5: ( action1.html )

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style-tracker.css">
    <title>Tracker</title>
    <style>
        .action1{
            display: flex;
            align-items: center;
            flex-direction: column;
        }









    </style>
</head>
<body style="background:bisque;">
    <div class="container">
```

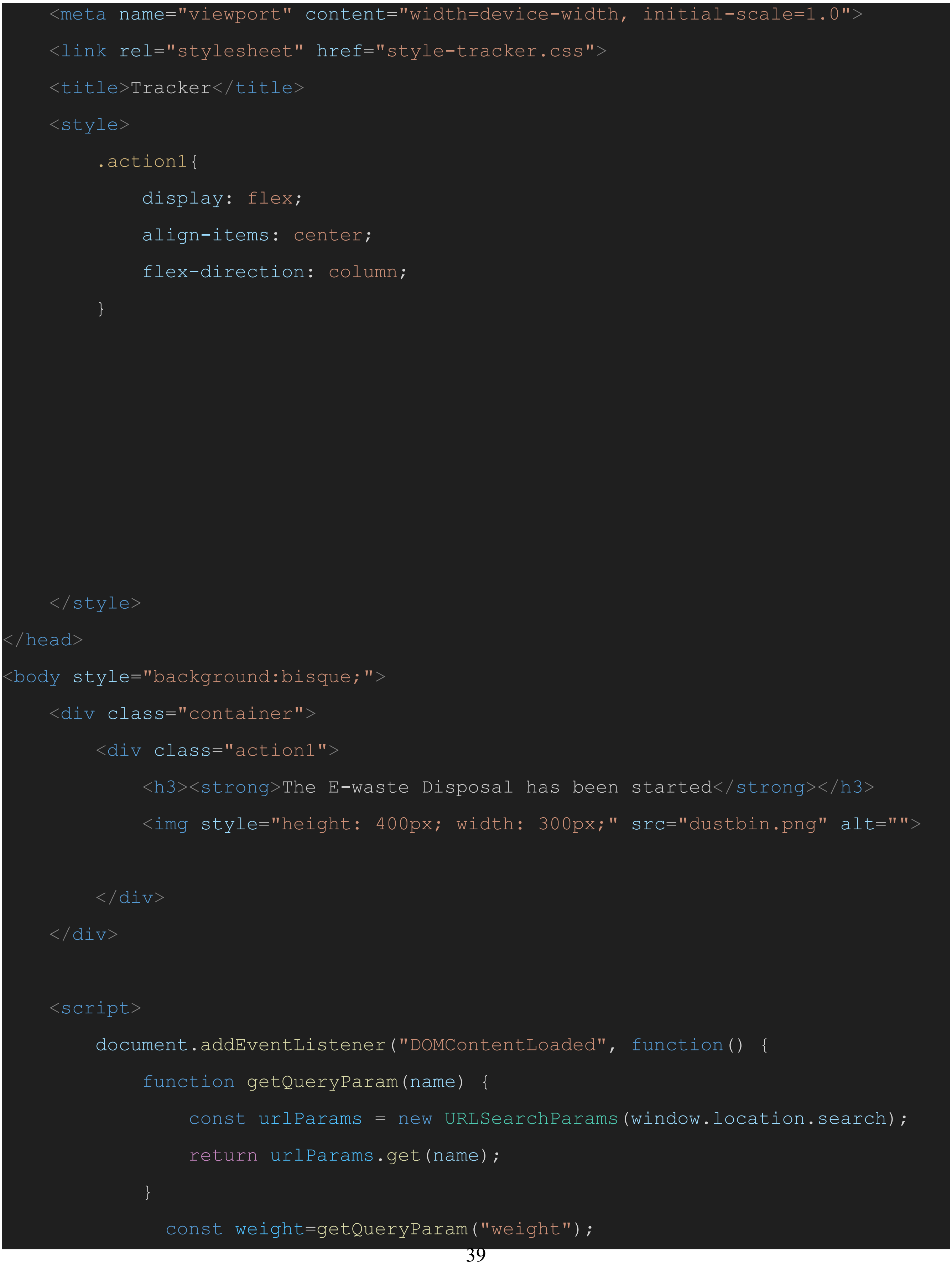


```
        <div class="action1">
            <h3><strong>The metal extraction has been started</strong></h3>
            <img style="height: 400px; width: 300px; border-radius: 10px;"
src="action1.png" alt="">

        </div>
    </div>

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            function getQueryParam(name) {
                const urlParams = new URLSearchParams(window.location.search);
                return urlParams.get(name);
            }
              const weight=getQueryParam("weight");
              const metal=getQueryParam("metal");
              const density=getQueryParam("density");
              const w=document.getElementById("w");
              const m=document.getElementById("m");
              const d=document.getElementById("d");
              if(weight&& metal&& density){
                w.textContent=weight;
                m.textContent=metal;
                d.textContent=density;

              }

            // Get Bin ID from URL
            const binId = getQueryParam("id");
            const binIdDisplay = document.getElementById("binIdDisplay");

            if (binId) {
                binIdDisplay.textContent = `BIN Status:(${binId})`;
            } else {
```

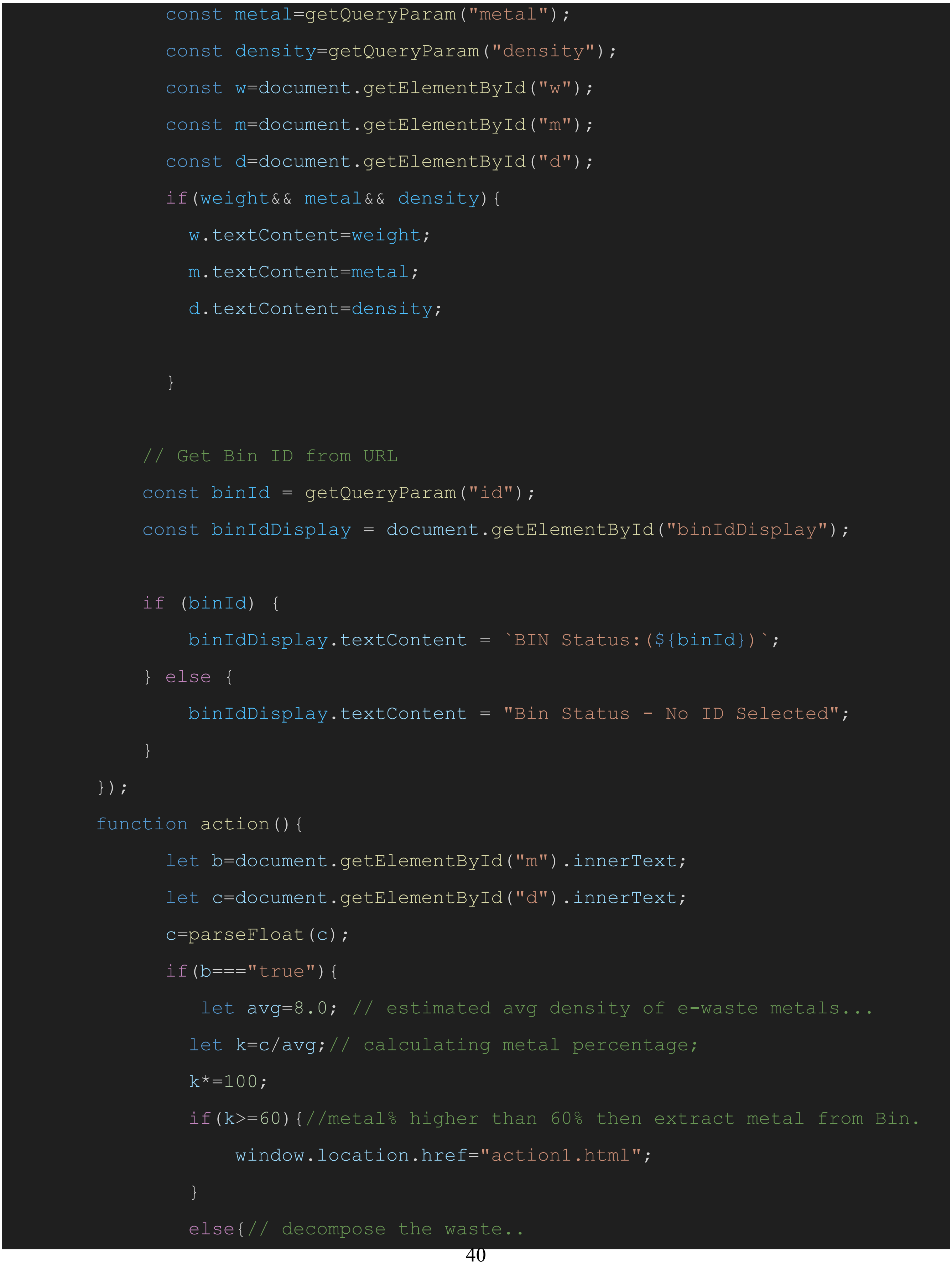
```
        binIdDisplay.textContent = "Bin Status - No ID Selected";
      }
    });
    function action(){
       let b=document.getElementById("m").innerText;
       let c=document.getElementById("d").innerText;
       c=parseFloat(c);
       if(b==="true"){
         let avg=8.0; // estimated avg density of e-waste metals...
        let k=c/avg;// calculating metal percentage;
        k*=100;
        if(k>=60){//metal% higher than 60% then extract metal from Bin.
          window.location.href="action1.html";
        }
        else{// decompose the waste..
          window.location.href="action2.html";
        }
       }
       else{
        window.location.href="action2.html";
       }
    }
  </script>
</body>
</html>
```

Appendix 6: ( action2.html )

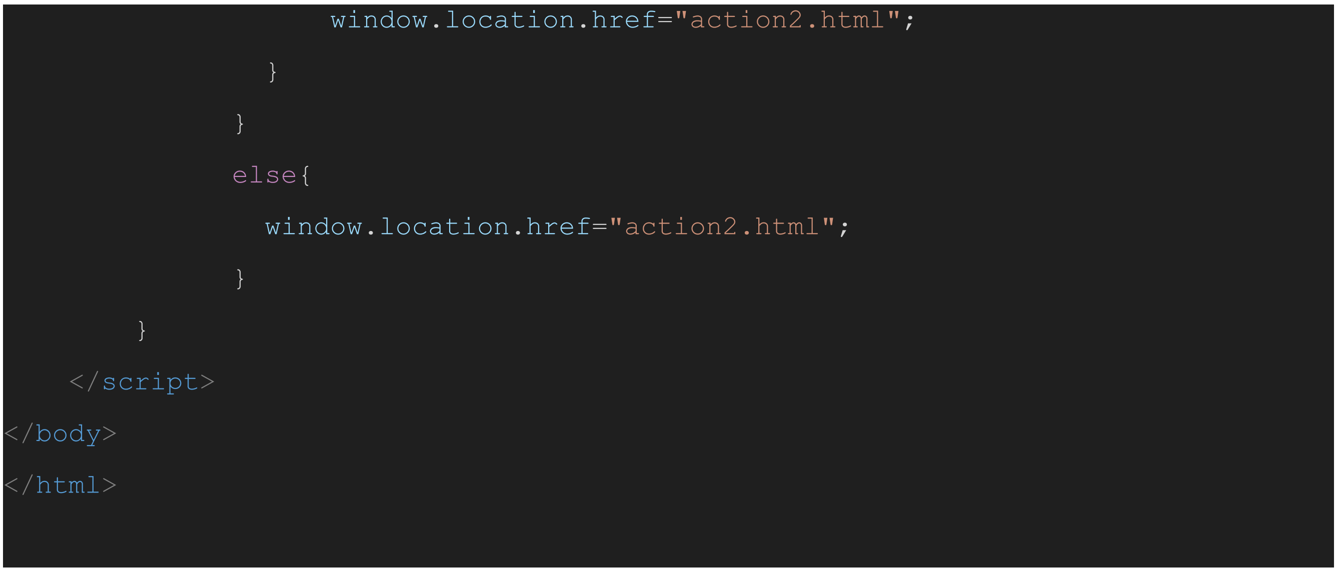
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```



```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style-tracker.css">
    <title>Tracker</title>
    <style>
        .action1{
            display: flex;
            align-items: center;
            flex-direction: column;
        }







    </style>
</head>
<body style="background:bisque;">
    <div class="container">
        <div class="action1">
            <h3><strong>The E-waste Disposal has been started</strong></h3>
            <img style="height: 400px; width: 300px;" src="dustbin.png" alt="">

        </div>
    </div>

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            function getQueryParam(name) {
                const urlParams = new URLSearchParams(window.location.search);
                return urlParams.get(name);
            }
              const weight=getQueryParam("weight");
```

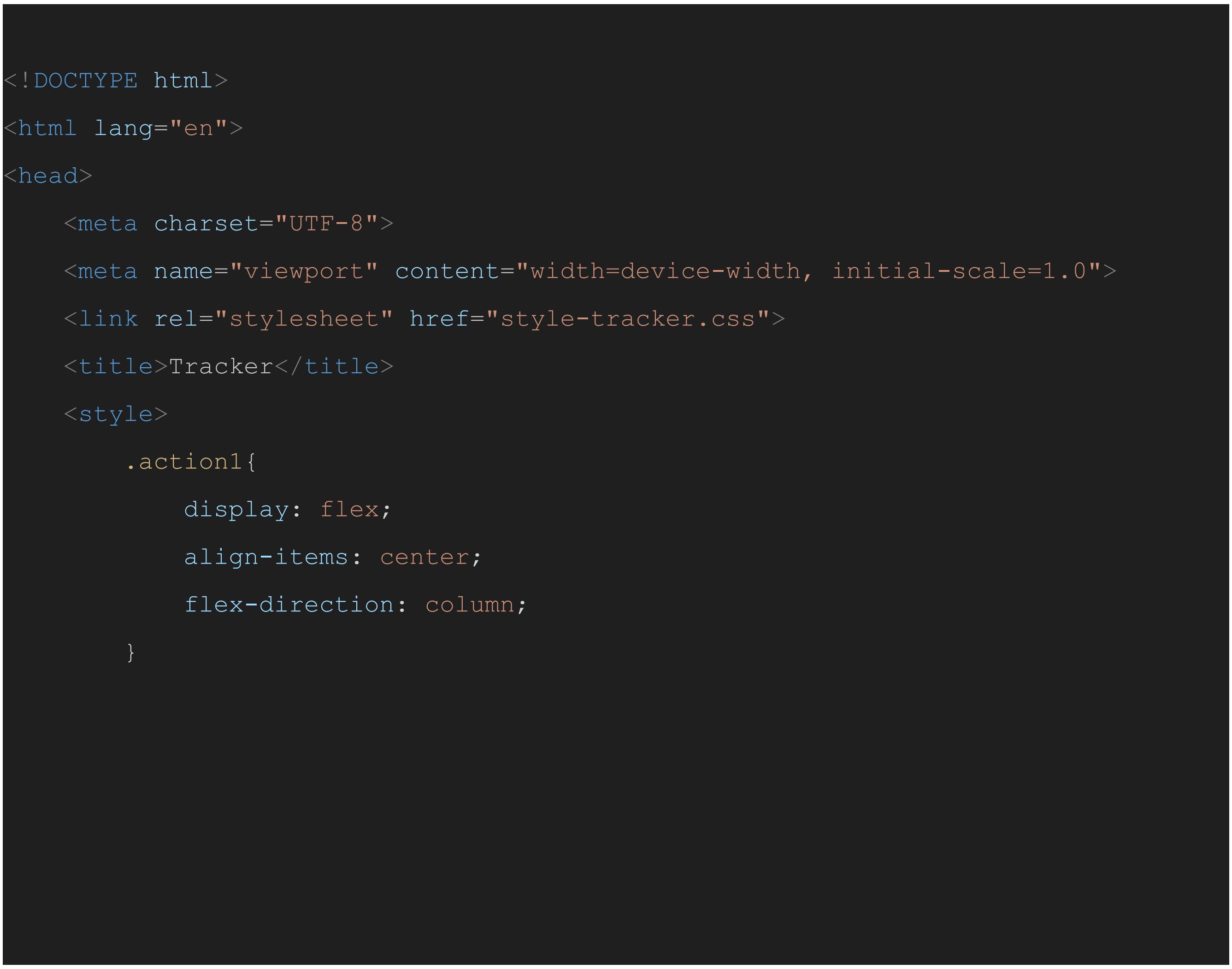


```
      const metal=getQueryParam("metal");
      const density=getQueryParam("density");
      const w=document.getElementById("w");
      const m=document.getElementById("m");
      const d=document.getElementById("d");
      if(weight&& metal&& density){
        w.textContent=weight;
        m.textContent=metal;
        d.textContent=density;

      }

    // Get Bin ID from URL
    const binId = getQueryParam("id");
    const binIdDisplay = document.getElementById("binIdDisplay");

    if (binId) {
        binIdDisplay.textContent = `BIN Status:(${binId})`;
    } else {
        binIdDisplay.textContent = "Bin Status - No ID Selected";
    }
});
function action(){
      let b=document.getElementById("m").innerText;
      let c=document.getElementById("d").innerText;
      c=parseFloat(c);
      if(b==="true"){
         let avg=8.0; // estimated avg density of e-waste metals...
        let k=c/avg;// calculating metal percentage;
        k*=100;
        if(k>=60){//metal% higher than 60% then extract metal from Bin.
            window.location.href="action1.html";
        }
        else{// decompose the waste..
```

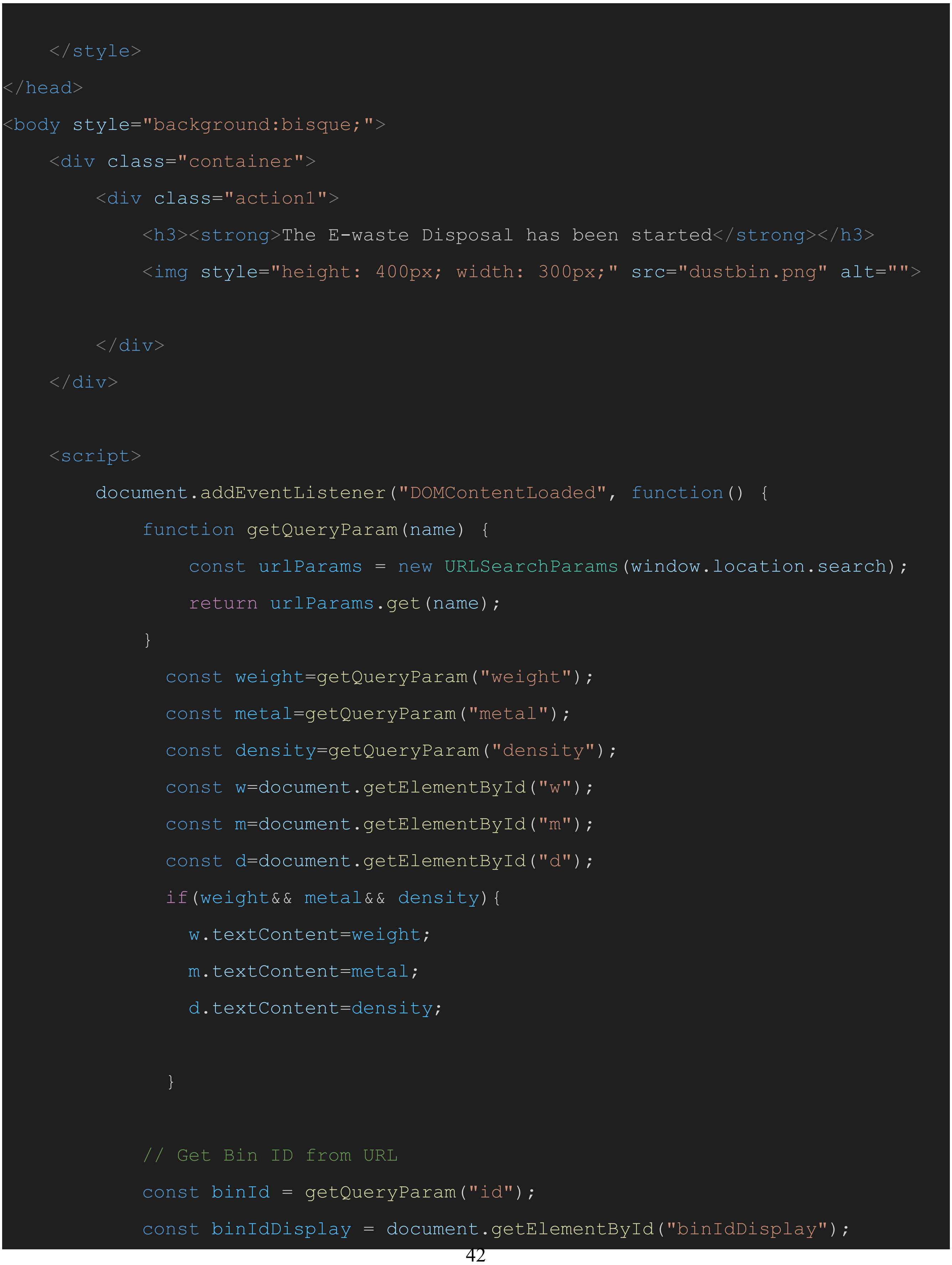


```
                    window.location.href="action2.html";
                }
              }
              else{
                window.location.href="action2.html";
              }
        }
    </script>
</body>
</html>
```

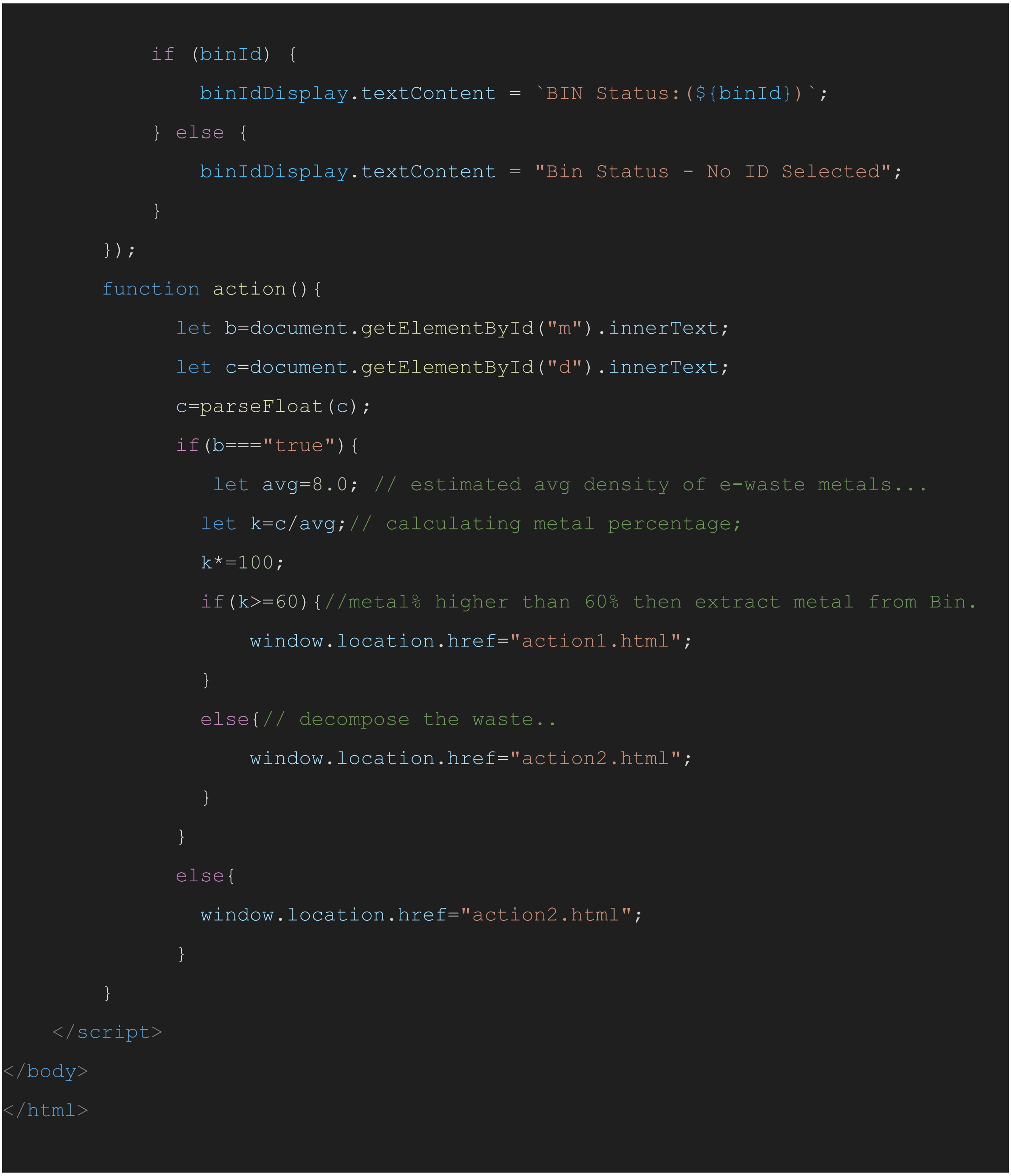
Appendix 7: ( track.html )



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style-tracker.css">
    <title>Tracker</title>
    <style>
        .action1{
            display: flex;
            align-items: center;
            flex-direction: column;
        }
```



```
    </style>
</head>
<body style="background:bisque;">
    <div class="container">
        <div class="action1">
            <h3><strong>The E-waste Disposal has been started</strong></h3>
            <img style="height: 400px; width: 300px;" src="dustbin.png" alt="">

        </div>
    </div>

    <script>
        document.addEventListener("DOMContentLoaded", function() {
            function getQueryParam(name) {
                const urlParams = new URLSearchParams(window.location.search);
                return urlParams.get(name);
            }
              const weight=getQueryParam("weight");
              const metal=getQueryParam("metal");
              const density=getQueryParam("density");
              const w=document.getElementById("w");
              const m=document.getElementById("m");
              const d=document.getElementById("d");
              if(weight&& metal&& density){
                w.textContent=weight;
                m.textContent=metal;
                d.textContent=density;

              }

            // Get Bin ID from URL
            const binId = getQueryParam("id");
            const binIdDisplay = document.getElementById("binIdDisplay");
```

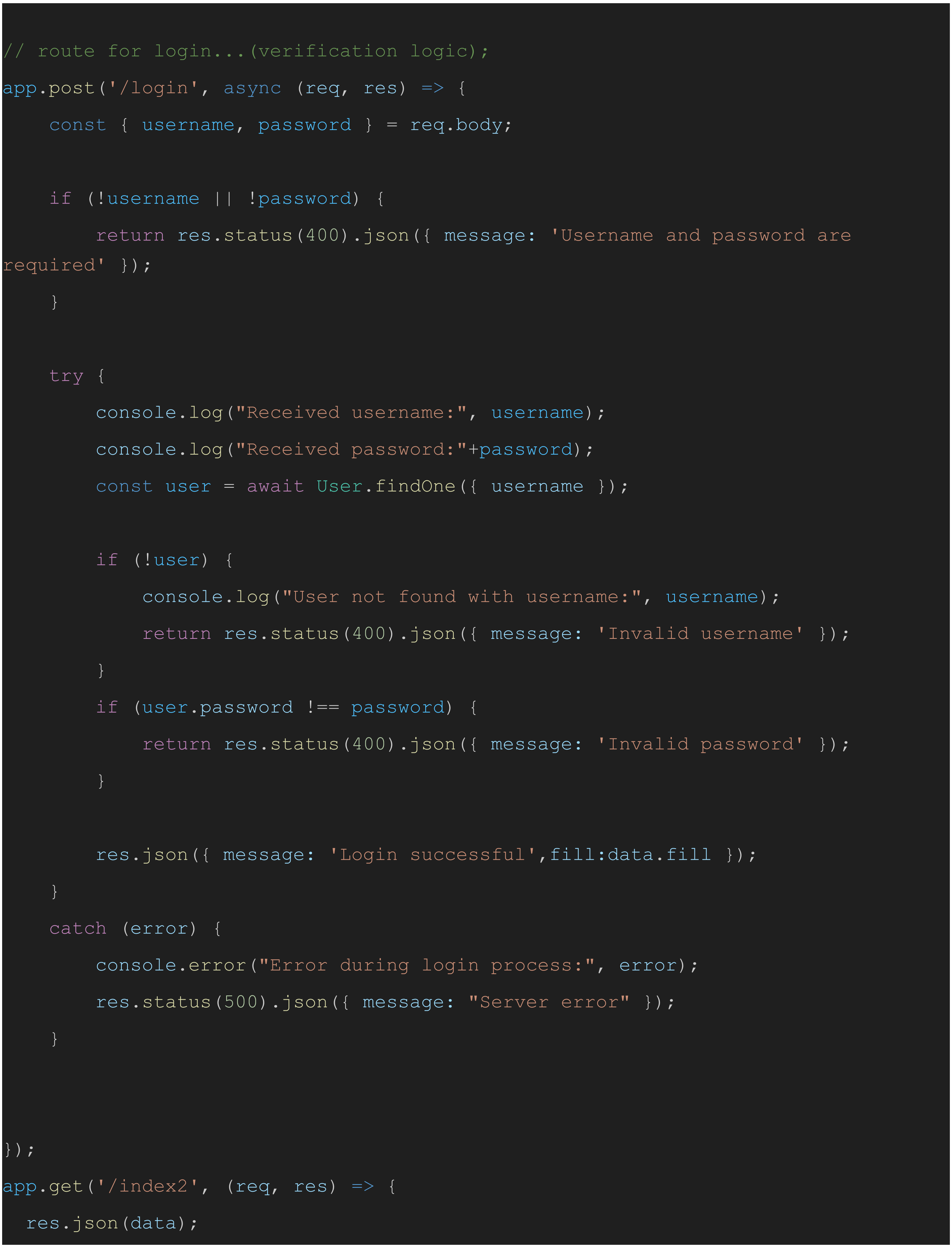


```
            if (binId) {
                binIdDisplay.textContent = `BIN Status:(${binId})`;
            } else {
                binIdDisplay.textContent = "Bin Status - No ID Selected";
            }
        });
        function action(){
              let b=document.getElementById("m").innerText;
              let c=document.getElementById("d").innerText;
              c=parseFloat(c);
              if(b==="true"){
                 let avg=8.0; // estimated avg density of e-waste metals...
                let k=c/avg;// calculating metal percentage;
                k*=100;
                if(k>=60){//metal% higher than 60% then extract metal from Bin.
                    window.location.href="action1.html";
                }
                else{// decompose the waste..
                    window.location.href="action2.html";
                }
              }
              else{
                window.location.href="action2.html";
              }
        }
    </script>
</body>
</html>
```

## Backend:

Appendix 8: ( server.js )

```
//imports
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();
const port = 3000;
app.use(cors());

// Middleware
app.use(express.json());

// MongoDB connection and fetching data...
mongoose.connect('mongodb://localhost:27017/admin', { useNewUrlParser: true,
useUnifiedTopology: true })
  .then(() => {
    console.log("Database connected successfully");
  })
  .catch(err => {
    console.error("Error while connecting to MongoDB");
  });

const userSchema = new mongoose.Schema({
    username: { type: String, required: true, unique: true },
    password: { type: String, required: true }
});

const User = mongoose.model('User', userSchema,'Users');
let f=0.0;
let data={
  weight:0.0,metal:false,fill:0.0
};
```



```
// route for login...(verification logic);
app.post('/login', async (req, res) => {
    const { username, password } = req.body;

    if (!username || !password) {
        return res.status(400).json({ message: 'Username and password are
required' });
    }

    try {
        console.log("Received username:", username);
        console.log("Received password:"+password);
        const user = await User.findOne({ username });

        if (!user) {
            console.log("User not found with username:", username);
            return res.status(400).json({ message: 'Invalid username' });
        }
        if (user.password !== password) {
            return res.status(400).json({ message: 'Invalid password' });
        }

        res.json({ message: 'Login successful',fill:data.fill });
    }
    catch (error) {
        console.error("Error during login process:", error);
        res.status(500).json({ message: "Server error" });
    }


});
app.get('/index2', (req, res) => {
  res.json(data);
  console.log("data sent to html");
});

app.post('/sensor', (req, res) => {
  data.weight=req.body.weight;
  data.metal=req.body.metalDetected;
  data.fill=req.body.fillLevel;

  console.log("Received data from ESP32:", data);
  res.status(200).send("Data received");
});



app.listen(port,()=>{
    console.log("server started");
});
```

**Hardware Logic: ( ESP 32 )**

```
#include "HX711.h"
#include <WiFi.h>
#include <HTTPClient.h>
#include <Wire.h>
#include <ArduinoJson.h>

// Wi-Fi Credentials
const char* ssid = "🗿 🗿 🗿 🗿";       // replace with your Wi-Fi SSID
const char* password = "12345678"; // replace with your Wi-Fi Password

// Server Details
const char* serverUrl = "http://192.168.175.163:3000/sensor"; // replace with your server IP and port

// Weight Sensor (HX711)
```

```
#define DT  27   // HX711 Data pin
#define SCK 26   // HX711 Clock pin
HX711 scale;
float weight;
float calibration_factor = 99500;

// Ultrasonic Sensor (HC-SR04)
#define TRIG_PIN 33
#define ECHO_PIN 25

// Proximity Sensor (LJ12A3-4-Z/BX)
#define SENSOR_PIN 32

//Code for connecting to wifi network
void connectToWiFi() {
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);

  int retries = 0;
  while (WiFi.status() != WL_CONNECTED && retries < 20) {
    delay(500);
    Serial.print(".");
    retries++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi connected!");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nFailed to connect to WiFi");
  }
}

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  connectToWiFi();

  // Setup HX711 (Weight Sensor)
  scale.begin(DT, SCK);
  scale.set_scale();
  scale.tare();
  long zero_factor = scale.read_average();
  Serial.print("Zero factor: ");
  Serial.println(zero_factor);

  // Setup Ultrasonic Sensor
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  // Setup Proximity Sensor
  pinMode(SENSOR_PIN, INPUT);
}

//Driver code
void loop() {
 //1. Read Weight Sensor
  scale.set_scale(calibration_factor);
  weight = scale.get_units(5);
  float finalWeight = (weight * 1000); // grams
  Serial.println(finalWeight);

  // 2. Read Ultrasonic Sensor
  long duration;
  float distance;
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
```

```
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);
duration = pulseIn(ECHO_PIN, HIGH);
distance = (duration * 0.0343) / 2;
float fillLevel = (24.97 - distance); // assuming 30 cm is max depth
Serial.println(fillLevel);

// 3. Read Proximity Sensor
bool metalDetected = (digitalRead(SENSOR_PIN) == LOW); // true if LOW
Serial.println(metalDetected);

// 4. Prepare JSON
StaticJsonDocument<200> doc;
doc["weight"] = finalWeight;
doc["fillLevel"] = fillLevel;
doc["metalDetected"] = metalDetected;

String jsonData;
serializeJson(doc, jsonData);

// 5. Send Data via HTTP POST
if (WiFi.status() == WL_CONNECTED) {
  HTTPClient http;
  http.begin(serverUrl);
  http.addHeader("Content-Type", "application/json");

  int httpResponseCode = http.POST(jsonData);

  if (httpResponseCode > 0) {
    Serial.print("Data sent successfully. Response code: ");
    Serial.println(httpResponseCode);
  } else {
    Serial.print("Error sending data. Response code: ");
```

```
      Serial.println(httpResponseCode);
    }

    http.end();
  } else {
    Serial.println("WiFi not connected. Trying to reconnect...");
    connectToWiFi();
  }

  Serial.println("------------------------");
  delay(2000); // Wait 2 seconds before next reading
}
```

# REFERENCES

[1] Forti, V., Balde, C. P., Kuehr, R., & Bel, G. (2020). The Global E-waste Monitor 2020. United Nations University.

[2] Robinson, B. H. (2009). E-waste: An assessment of global production and environmental impacts. Science of the Total Environment, 408(2), 183-191.

[3] Dwivedy, M., & Mittal, R. K. (2010). Future trends in e-waste recycling. Resources, Conservation and Recycling, 55(3), 347-364.

[4] United Nations. (2015). Sustainable Development Goals. https://sdgs.un.org/goals

[5] E-Mudhra. (2018). Digital Recycling Initiatives. Retrieved from www.emudhra.com

[6] Sharma, A., & Pandey, N. (2020). Digital waste solutions in urban India. Journal of Environmental Planning, 45(2), 156–162.

[7] Ramachandra, T. V., & Bachamanda, S. (2007). Environmental audit of municipal solid waste management. Journal of Environmental Management, 85(4), 683–697.

[8] Islam, M., & Huda, N. (2019). E-waste in informal sector: A case study. Waste Management, 95, 533–545.

[9] Wang, Y., et al. (2020). Blockchain for e-waste tracking. Computers & Industrial Engineering, 145, 106487.

[10] Singh, N., & Maheshwari, P. (2018). Citizen perception towards e-waste management. Waste and Resource Management, 4(1), 22–28.

[11] Thomas, V. M., & Spataru, S. (2019). Barriers to sustainable recycling. Resources Policy, 60, 112–118.

[12] Tripathi, M., & Rawat, A. (2021). Data security challenges in waste management systems. Procedia Computer Science, 173, 305–310.

[13] Chaturvedi, A., & Sharma, P. (2020). Gaps in e-waste policy implementation in India. Environmental Science and Policy, 112, 395–403.

[14] Kumar, A., & Singh, A. (2021). Geo-tracking for waste logistics. Journal of Urban Technologies, 33(1), 50–62.

[15] Roy, S., & Dutta, P. (2022). Agile Development of Sustainable IT Solutions. IJSDIT, 12(4), 73–85.

[16] Bansal, R., & Mehta, M. (2020). Securing environmental IT systems. Journal of Cybersecurity and Environment, 5(3), 100–110.

[17] Brooke, J. (1996). SUS: A "quick and dirty" usability scale. Usability Evaluation in Industry, 189(194), 4–7.

[18] Jain, D., & Malhotra, N. (2023). E-waste collection via digital platforms: A university case study. Environmental Informatics Archives, 15, 87–93.

[19] Sinha, S., & Gupta, R. (2021). User engagement in e-sustainability apps. IJHCI, 14(3), 198–209.

[20] Shenoy, M., & Iyer, V. (2022). Performance analysis of MERN stack applications. Journal of Web Engineering, 18(5), 509–526.

[21] Ahmed, A., & Kaleem, S. (2023). Comparative study of waste disposal platforms. Journal of Environmental Tech Studies, 27(2), 35–47.

[22] Banerjee, T. (2020). Promoting circular economy through digital waste systems. Circular Economy Journal, 3(1), 42–51.

[23] Kumar, V. (2019). Policy roadmap for sustainable e-waste handling. India Tech Review, 10(2), 15–22.

[24] Basu, D., & Rao, S. (2022). Behavioral study on electronic disposal. Journal of Environmental Psychology, 29(4), 324–332.

[25] Patel, M., & Nair, A. (2023). Digitizing sustainability: Emerging platforms and challenges. Journal of Green IT, 8(1), 61–75.

[26] Arora, R., & Jain, S. (2019). E-waste management: A review of policies and practices in India. Environmental Science and Pollution Research, 26(1), 1-12. https://doi.org/10.1007/s11356-019-05998-x

[27] Kiddee, P., Naidu, R., & Wong, M. H. (2013). Electronic waste management approaches: An overview. Waste Management, 33(5), 1237-1250. https://doi.org/10.1016/j.wasman.2013.01.006

[28] Baldé, C. P., Forti, V., Gray, V., Kuehr, R., & Stegmann, P. (2017). The Global E-waste Monitor 2017: Quantities, flows, and resources. United Nations University (UNU), International Telecommunication Union (ITU), and International Solid Waste Association (ISWA).

[29] Robinson, B. H. (2009). E-waste: An assessment of global production and environmental impacts. Science of the Total Environment, 408(2), 183-191. https://doi.org/10.1016/j.scitotenv.2009.09.044

[30] Zhang, L., Schnoor, J. L., & Zeng, E. Y. (2012). E-waste recycling: Where does it go from here? Environmental Science & Technology, 46(20), 10861-10867. https://doi.org/10.1021/es302193s