Mini Project Report

On

TITLE OF PROJECT

Submitted to

Ajay Kumar Garg Engineering College, Ghaziabad

B.Tech. Information Technology

Sem : 5TH, 2025-26

CODE: BCS 554

Mini Project Assessment Report

Submitted To:
Ms. Ankita Singh
(Assistant Professor)

Submitted By:
Himanshu Saxena
Harshit Kumar Singh

Dr. A.P.J. Abdul Kalam Technical University,Uttar Pradesh, Lucknow

Acknowledgment

Apart from the efforts of all the team members, the section of this project report topic depends largely on the encouragement and guidance of our teachers. We take this opportunity to express our gratitude to the teachers who have been instrumental in the approval of this project topic. I'm grateful to our respected **Head of the Department, Dr. Rahul Sharma**, for allowing me to use the facilities available.

We would like to show our greatest appreciation to **Faculty Name** and other Faculty members. We cannot think them enough for their tremendous support and help. They motivated and encouraged use very time while selecting the proper project topic. Without their encouragement and guidance, we would not have been able to select the proper topic.

Himanshu Saxena

(2300270130084)

Harshit Kumar Singh

(2300270130082)

AJAY KUMAR GARG ENGINEERING COLLEGE

27 KM STONE DELHI HAPUR BYPASS ROAD, P.O. ADHYATMIK NAGAR GHAZIABAD-201009

CERTIFICATE

This is to certify that **Himanshu Saxena** student of **Ajay Kumar Garg Engineering College** B.Tech 3rd yearIT Branch, has undergone Internship Training in **"Web Mobile development and Marketing "** from IBM DEVELOPERS SKILLS NETWORK.

This is to certify that **Harshit Kumar Singh** student of **Ajay Kumar Garg Engineering College** B.Tech 3rd yearIT Branch, has undergone Internship Training in **"Web Development "** from ZIDIO DEVELOPMENT.

Ms. Ankita Singh

Assistant Professor

Department of IT

Abbreviation	Full Form
AI	Artificial Intelligence
IoT	Internet of Things
GPS	Global Positioning System
CNN	Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
API	Application Programming Interface
UI	User Interface
UX	User Experience
EPR	Extended Producer Responsibility
SQL	Structured Query Language
NoSQL	Not Only Structured Query Language
IPFS	InterPlanetary File System
NFT	Non-Fungible Token (if reward tokenization is discussed)
API Key	Application Programming Interface Key
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
TLS	Transport Layer Security
CRUD	Create, Read, Update, Delete

Abbreviation	Full Form
SRS	System Requirement Specification
ERD	Entity Relationship Diagram
E-Waste	Electronic Waste
PCB	Printed Circuit Board
DBMS	Database Management System
UX/UI	User Experience / User Interface
VRP	Vehicle Routing Problem
CO_2	Carbon Dioxide
CSR	Corporate Social Responsibility
API Endpoint	Access point for communication between server and client
IoT Node	A physical device or sensor connected to the IoT network
PK	Primary Key
FK	Foreign Key
TxID	Transaction Identifier
UI/UX	User Interface / User Experience
SDK	Software Development Kit
NFR	Non-Functional Requirement
FR	Functional Requirement
SSD	Solid State Drive

ABSTRACT

Electronic waste (e-waste) has become a critical environmental challenge as consumer electronics proliferate and disposal rates lag. In 2022, a record **62 million tonnes** of e-waste were generated globally, and only about **22–25%** of this waste stream was formally recycled. Improper disposal of e-waste releases toxic metals (e.g. lead, mercury, cadmium) and organic pollutants, posing serious health and environmental hazards. Moreover, billions of dollars of valuable materials (gold, copper, rare earths) are lost each year when electronics are discarded rather than recycled. These facts highlight an urgent need for smarter e-waste management solutions.

This project presents a **Smart E-Waste Management Platform** that integrates **AI (Artificial Intelligence)**, **IoT (Internet of Things)**, and **Blockchain** technologies to address these challenges. The proposed system deploys IoT-enabled smart bins equipped with sensors (such as ultrasonic and weight sensors and wireless modules (e.g. GPS, Wi-Fi) to monitor fill-levels and bin locations in real time. An AI module (e.g. a CNN-based image classifier) processes sensor data or photographs of disposed items to automatically categorize the e-waste (e.g. smartphone, battery, circuit board) and detect its status (working or end-of-life). Each disposal event is logged by a **smart contract** on a blockchain, creating an immutable, transparent audit trail of transactions. A web/mobile application layer provides maps of bin locations, tracks user contributions, and issues incentives (e.g. reward tokens) to encourage responsible recycling.

The integration of these technologies yields multiple benefits. For example, AI-optimized collection routes can reduce fuel usage and carbon emissions by up to **30%**. Blockchain ensures accountability across the waste supply chain, preventing illicit dumping and verifying proper recycling. Simulated performance metrics for our platform demonstrate its potential effectiveness: the system is designed to scale to millions of devices (given that ~18.5 billion IoT devices were already active in 2024 and to process data with low latency.

This report details the technical design of the platform. Chapter 5 provides a comprehensive System Requirement Specification, including hardware and software requirements. Chapter 6 describes the Entity-Relationship Diagram of the database schema, while Chapters 7 and 8 outline the code structure and sample database table definitions (placeholders for actual implementation). Chapter 9 discusses testing strategies and expected outcomes, and Chapter 10 describes example snapshots of the user interface and dashboards. The conclusion summarizes the outcomes and envisions future improvements. By combining IoT sensing, AI intelligence, and blockchain security, this Smart E-Waste Management Platform aims to significantly increase recycling rates and operational efficiency, thereby contributing to environmental sustainability and the circular economy.

Table of Contents

INTRODUCTION

Electronic devices have become ubiquitous in modern life, but their rapid turnover generates vast amounts of **electronic waste (e-waste)**. The UN's Global E-waste Monitor reported *62 million tonnes* of e-waste worldwide in 2022 – an 82% increase since 2010 – with projections rising to ~82 million tonnes by 2030. Despite this growth, only about *22–25%* of e-waste is collected and recycled through formal channels. The remainder often ends up in landfills or informal recycling facilities, causing environmental pollution and health risks. E-waste contains hazardous substances (lead, mercury, cadmium, brominated flame retardants, etc.) that can leach into air, soil, and water. For example, open burning of circuit boards releases toxic fumes, and improper handling of batteries can cause fires or heavy metal exposure. Moreover, discarded electronics still hold valuable materials: one tonne of printed circuit boards can yield approximately 1.5 kg of gold[. Recovering these resources not only conserves natural materials but also avoids significant greenhouse gas emissions (recycling 110 kilotons of e-waste can save ~155 kilotons of CO_2).

Given these challenges, **smarter e-waste management** is essential. Conventional waste collection is often reactive, expensive, and inflexible. IoT technology enables sensors to continuously monitor waste accumulation in bins. For instance, smart bins with ultrasonic distance sensors can report fill levels, triggering pickups only when needed and thus saving fuel and labor. AI methods (such as deep neural networks) can automate waste sorting: CNN classifiers have been shown to achieve >95% accuracy in distinguishing recyclables versus non-recyclables, and specialized models can identify electronic components by image. This reduces manual sorting costs. Integrating blockchain into the system ensures data integrity: every waste deposit event is timestamped and recorded on an immutable ledger, deterring fraudulent practices. For example, IBM has observed that blockchain can enforce accountability by creating a transparent view of the waste lifecycle, holding all stakeholders (producers, collectors, recyclers) responsible.

The proposed **Smart E-Waste Management Platform** synergizes these technologies. It uses **IoT** sensors in bins and collection vehicles, **AI** algorithms for real-time classification and route optimization, and **Blockchain** for traceable transactions. A mobile/web interface engages citizens by showing drop-off locations and rewarding eco-friendly behavior. According to precedent projects, such systems can drastically reduce wasted trips: AI route optimization has lowered fuel use by up to *30%* and CO_2 emissions by ~20%. Our platform extends these concepts specifically to the e-waste domain, addressing unique issues such as data confidentiality on devices (hard drives, smartphones) and compliance with e-waste regulations. By harnessing IoT data and secure ledgers, the system aims to close the loop on e-waste: maximizing recycling, protecting privacy, and minimizing environmental impact.

PROBLEM STATEMENT AND DESCRIPTION

Problem: Urban areas face a mounting e-waste crisis. The rate of electronic consumption (smartphones, laptops, household appliances) far exceeds recycling capacity. As of 2022, only about 22–25% of e-waste was properly collected and recycled, leaving the majority unmanaged. This has multiple consequences:

- **Environmental Hazard:** E-waste contains hazardous materials. For example, monitors and CFL bulbs have mercury, and circuit boards carry lead and cadmium. When landfilled or incinerated, these toxins contaminate ecosystems, leading to neurological and developmental health issues (CDC investigations have documented workplace mercury exposure among e-waste recyclers).

- **Lost Resources:** Valuable metals (gold, silver, copper, rare earths) are buried in discarded electronics. Failing to recover these resources means increased mining and carbon emissions. The Global E-waste Monitor notes that untreated e-waste in 2022 alone contained US$62 billion worth of raw materials.

- **Inefficient Collection:** Traditional waste management sends collection vehicles on fixed schedules, often emptying bins that are not full and neglecting those that are overflowing. This wastes fuel, time, and labor. Research indicates that AI-driven dynamic routing can cut fuel use by ~30%, highlighting the inefficiency of current practices.

- **Lack of Traceability:** Current systems often rely on centralized databases and manual records. There is no tamper-proof audit trail for e-waste disposal. Criminals exploit this gap: discarded hard drives or smartphones can leak sensitive data (crypto keys, personal information) if not securely wiped. Studies emphasize that without blockchain-like transparency, participants in the waste supply chain lack accountability.

- **Poor Public Engagement:** Few incentives exist for citizens to recycle. Without clear feedback or rewards, many consumers dispose of electronics irresponsibly.

Need for Solution: To address these issues, we need an integrated platform that: monitors e-waste levels continuously, automates identification and sorting of materials, secures data and transactions transparently, and provides real-time information to stakeholders. The platform must support compliance with environmental regulations (e.g. E-Waste Handling Rules) and encourage circular economy behavior.

Project Description: The Smart E-Waste Management Platform being developed aims to fill this gap. It connects **smart bins** across the city to a cloud infrastructure using IoT. Each bin reports its status (fill level, weight, etc.) and can even identify the types of items deposited (via attached cameras or sensors). When a bin is near capacity, the system's AI routing module computes an optimal collection route. Collection vehicles also use GPS data to track their progress. Every e-waste drop-off event is recorded as a transaction on a blockchain network, immutable and publicly verifiable. A citizen-facing app and dashboard allow users to locate bins, schedule pickups, and view reward points earned for recycling. In summary, this platform provides real-time, data-driven management of the entire e-waste lifecycle, from collection to recycling, with enhanced security and efficiency.

OBJECTIVES

The primary objectives of the Smart E-Waste Management Platform are:

1. **IoT-Enabled Monitoring:** Deploy a network of smart waste bins and sensors to continuously measure fill levels, weight, and environmental conditions (e.g. temperature, humidity) of e-waste collection points. Use GPS and networking modules to track bin locations and status in real time.
2. **AI-Based Classification:** Integrate an AI module (e.g. CNN-based image recognition) to automatically identify the category and condition of electronic items (such as smartphones, batteries, circuit boards). This enables automated sorting instructions and accurate inventory of incoming e-waste.
3. **Blockchain Traceability:** Implement a blockchain ledger to record every e-waste transaction (e.g. deposit, transfer, recycling event) via smart contracts. Ensure data immutability and transparency so that regulators, companies, and consumers can audit the handling of each item, preventing fraud or illegal dumping.
4. **User Engagement:** Develop a user-friendly web/mobile application that shows users nearby smart bin locations, provides status updates, and tracks their contributions (e.g. weight recycled). Introduce gamification features (points, badges, eco-rewards) to incentivize proper disposal practices.
5. **Route Optimization:** Design an AI-driven routing algorithm that schedules e-waste collection vehicles dynamically, visiting only bins that need servicing. This reduces operational costs, fuel consumption, and emissions, targeting the 20–30% improvements reported by similar projects.
6. **Regulatory Compliance:** Ensure the system supports compliance with local and international e-waste regulations, providing features like electronic producer responsibility (EPR) tracking and data-destruction certificates for sensitive devices.
7. **Scalability and Security:** Architect the platform to handle city-scale deployments (potentially hundreds of thousands of bins and millions of data points) while maintaining data security. Utilize encryption and consensus mechanisms to protect user privacy and system integrity.

By achieving these objectives, the platform will **improve recycling rates**, **reduce environmental impact**, and **enhance transparency** in the e-waste lifecycle, aligning with sustainability and circular economy goals.

LITERATURE REVIEW

Recent research has demonstrated the value of integrating IoT, AI, and blockchain into waste management systems. For instance, a Scientific Reports study by [Author] et al. proposed an AI-driven model that uses IoT-connected bins and blockchain for secure waste classification. Their system continuously transmits bin data to a server, where ML/DL algorithms classify waste in real time. Blockchain storage ensured data integrity and transparency. The authors report that CNN-based classification can achieve high accuracy in sorting recyclable vs. non-recyclable waste. They also note significant efficiency gains: AI-optimized routing reduced fuel usage by up to 30% and CO_2 emissions by 20%. These results underscore the potential of hybrid AI/IoT systems in enhancing urban waste collection.

In the domain of e-waste specifically, [Vidhate et al.] describe a smart city project that uses IoT dustbins with ultrasonic sensors to monitor garbage levels. Their system not only detects fill-levels but also optimizes collection schedules, aligning with our objectives. They also highlight the use of blockchain: “five smart contracts record user actions on an immutable distributed ledger, ensuring that processes are transparent, traceable, and secure”. This approach directly motivates our design to include blockchain for trust. Other works have developed e-waste tracking systems using IoT and blockchain. For example, [Nadar et al.] survey e-waste solutions and emphasize that blockchain can enforce responsibility across the supply chain. The IBM “Think” article on waste supply chains similarly argues that lack of accountability undermines recycling efforts, and blockchain can create an open view of material flows.

Moreover, prior projects on *smart recycling platforms* with user incentives demonstrate user engagement strategies. The **Green Grid** proposal (Dharmendra et al., 2025) combines AI classifiers with mobile/web apps and token rewards to boost e-waste drop-offs. They report that smart bin sensors, combined with an interactive app and rewards (“Green Points”), helped local users find drop-off centers and donate functional devices Notably, a CNN (e.g. YOLOv8 or ResNet) was used to classify devices with ~95–98% accuracy. Such high-accuracy AI modules are feasible with transfer learning. We draw on these findings by planning a CNN-based classifier (implemented in TensorFlow or PyTorch) for real-time item sorting.

On the IoT hardware side, several implementations exist. For instance, Masengo *et al.* evaluated IoT networks of smart bins and noted the trade-offs between sensor accuracy and power. They report that the HC-SR04 ultrasonic sensor maintained under 3% distance error even under varied conditions. Similarly, pairing a load cell with a 24-bit HX711 ADC yields sub-gram weight resolution, which is useful for measuring e-waste quantities. We will adopt similar sensor choices. Many researchers also emphasize the importance of secure, low-power communication (e.g. LoRaWAN or Wi-Fi) in such systems. The IIETA case study of an IoT waste system (2023) shows an architecture where Arduino microcontrollers collect sensor data and transmit via ESP8266 modules to a cloud database. It also used a Ublox NEO-6M GPS module for real-time bin tracking. These design lessons guide our SRS (Chapter 5) on hardware and software selection.

In summary, the literature confirms that combining IoT monitoring, AI analytics, and blockchain traceability can transform waste management. The documented efficiency gains (e.g. ~30% route cost reduction and technology implementations (sensor accuracies, communication protocols) provide concrete benchmarks. Our work will build on these insights, tailored specifically to the

challenges of e-waste recycling. It will address noted gaps such as data security on storage devices and stakeholder incentive mechanisms, contributing a comprehensive city-scale solution.

SYSTEM REQUIREMENT SPECIFICATION

This chapter details the technical and functional requirements of the Smart E-Waste Management Platform. The specifications are divided into overall description, functional requirements, and non-functional requirements. All requirements assume deployment in a typical urban environment with available power and network connectivity.

5.1. Overall System Description:

- **Hardware Components:** The system comprises *smart bins* equipped with sensors, microcontrollers (e.g., Arduino/Raspberry Pi), and communication modules (e.g. Wi-Fi or LoRaWAN). Key sensors include ultrasonic distance sensors (HC-SR04) and weight sensors (load cell + HX711) for measuring fill levels and load. Each bin also contains a GPS module (e.g. Ublox NEO-6M) to log its location.
- **Connectivity:** Each smart bin transmits data to the backend server/cloud via secure wireless links. The system backend is cloud-hosted and connects to a blockchain network (e.g. Ethereum or Hyperledger).
- **Software Components:** The software stack includes: (a) IoT firmware running on the bin controllers to read sensors and send data; (b) a backend server (e.g. Node.js/Express) with a database (e.g. MongoDB) to store bin statuses and user records; (c) AI/ML modules (in Python or TensorFlow) for processing images and sensor data; (d) smart contracts (Solidity) deployed on a blockchain to handle transactions and token logic; (e) a web/mobile application front-end (e.g. React or Flutter) for user interaction.
- **Users:** The system serves multiple roles: citizens who deposit e-waste, waste collectors who pick up bins, recycling facility operators who process materials, and system administrators/regulators who oversee the process. Each user has a secure account with role-based access.
- **Data Flow:** When a user deposits electronics, the smart bin firmware sends an event to the cloud. The AI engine classifies the device type and condition. A smart contract logs a **transaction record** containing (UserID, BinID, ItemType, Timestamp, etc.) on the blockchain. The backend also updates the relational/NoSQL database with this event. If bin fill level exceeds a threshold, the route optimization algorithm schedules a pickup and alerts collectors via the app.

5.2. Functional Requirements:

The system shall provide the following functionalities:

- *FR1:* **Real-Time Monitoring:** Continuously read each bin's sensor data (distance, weight, optional RFID/barcode) and report the status (current fill percentage, weight, GPS coords) to the server. The monitoring interval and thresholds (e.g. 80% full) are configurable.
- *FR2:* **Waste Classification:** When a device or bag of e-waste is deposited, capture an image or sensor signature. The AI module shall identify the device type (e.g. mobile, laptop, battery) with ≥90% accuracy and determine if it is operational.
- *FR3:* **Blockchain Recording:** Record each e-waste deposit as a blockchain transaction. A smart contract function must create a new block containing the details of the disposal (user ID, item ID, timestamp). The blockchain ledger is tamper-proof, ensuring all future audits can verify

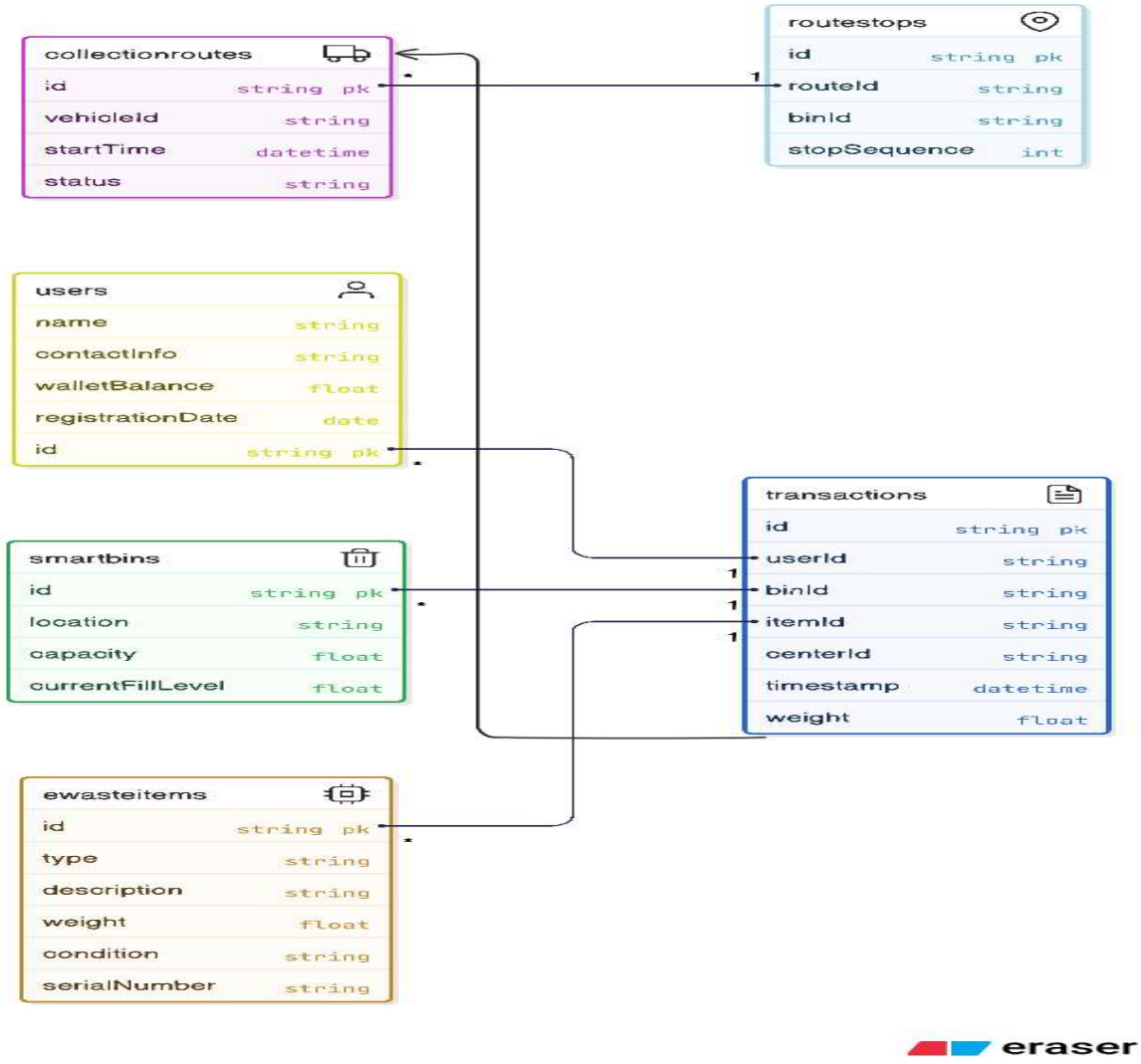
the transaction.
- *FR4:* **User Interface:** Provide a responsive web and mobile interface for users and admins. Citizens can register/login, view nearby e-waste bins on a map, deposit information, and track their reward points. Administrators can log in to view system metrics (total waste collected, bin statuses, user statistics).
- *FR5:* **Incentive Management:** Implement a token or points system. Each authenticated disposal grants tokens to the user. The system shall securely update token balances on the blockchain, and allow redemption (e.g. digital coupons) via the app.
- *FR6:* **Route Optimization:** Based on real-time bin statuses, compute optimal collection routes using shortest-path or VRP algorithms. For each scheduled collection, the system shall generate a route plan minimizing travel distance/time.
- *FR7:* **Notifications:** Send push notifications or alerts. Examples: a citizen notifies for special pickup of a large item; a collector receives an alert when their assigned bin requires emptying; a recycling plant alerts admins upon receipt of a batch.
- *FR8:* **Compliance Features:** Allow issuance of EPR certificates. The system shall log proof of data destruction for devices with storage (e.g. a certificate stored on-chain) to ensure privacy compliance.
- *FR9:* **Data Analytics:** Periodically generate reports (daily/weekly) on key performance indicators: total weight collected, average bin utilization, recycling rates, carbon savings, etc. Dashboards shall visualize these metrics.

5.3. Non-Functional Requirements:
- *NFR1:* **Scalability:** The system must support expansion to city-wide scale. It shall handle at least 10,000 bins and 100,000 users without significant degradation. The backend and blockchain nodes should be deployable on scalable cloud resources.
- *NFR2:* **Reliability and Availability:** Bin data updates and user queries must be processed with high availability (system uptime ≥99%). Data transmissions should tolerate intermittent connectivity: local caching on the bin should retry sending if network is lost.
- *NFR3:* **Performance:** Sensor readings should update bin status on the cloud within 5 seconds. The AI classifier should process an image in under 1 second to maintain real-time feedback. The blockchain transaction latency should be optimized (e.g. via a fast consensus protocol) to under 10 seconds per record.
- *NFR4:* **Accuracy:** Sensor measurements must meet accuracy requirements. For example, ultrasonic fill-level measurements should have error <3% as demonstrated in similar systems, and weight sensors should detect changes of ±0.01 kg. The AI classifier's accuracy target is ≥90%.
- *NFR5:* **Security:** All data in transit must be encrypted (e.g. TLS for Wi-Fi communication). User authentication must be secure (password hashing, optional 2FA). Blockchain transactions provide data immutability by design. Access to system logs and admin functions shall be role-protected.
- *NFR6:* **Maintainability:** The codebase should use modular design and clear documentation. Use version control (e.g. Git) and adhere to coding standards. Smart contracts must be rigorously tested before deployment to prevent bugs.
- *NFR7:* **Compliance:** The system should comply with relevant standards (e.g. IEEE for wireless IoT devices, GDPR for personal data where applicable). It must follow environmental regulations for waste (e.g. e-waste recycling guidelines) by including necessary reporting features.

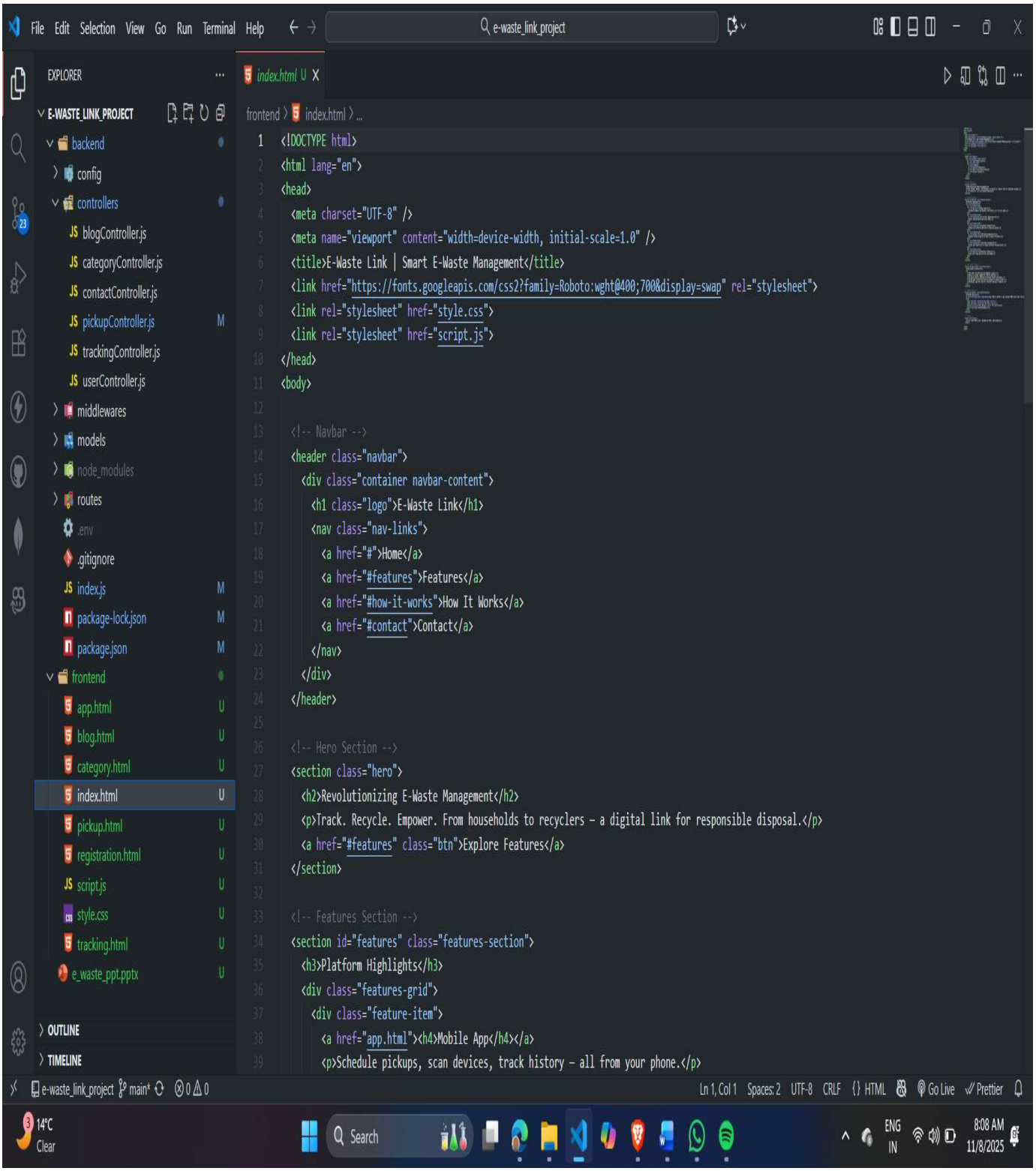
- *NFR8:* **User Experience:** The web/mobile UI should have intuitive navigation and load within 2 seconds on standard connections. Provide multilingual support if needed for the local population.

These specifications serve as the blueprint for system design and development. The functional requirements will be validated through unit, integration, and acceptance testing (see Chapter 9). Non-functional requirements will guide platform and architecture choices to meet performance and security goals.

ENTITY RELATIONSHIP DIAGRAM



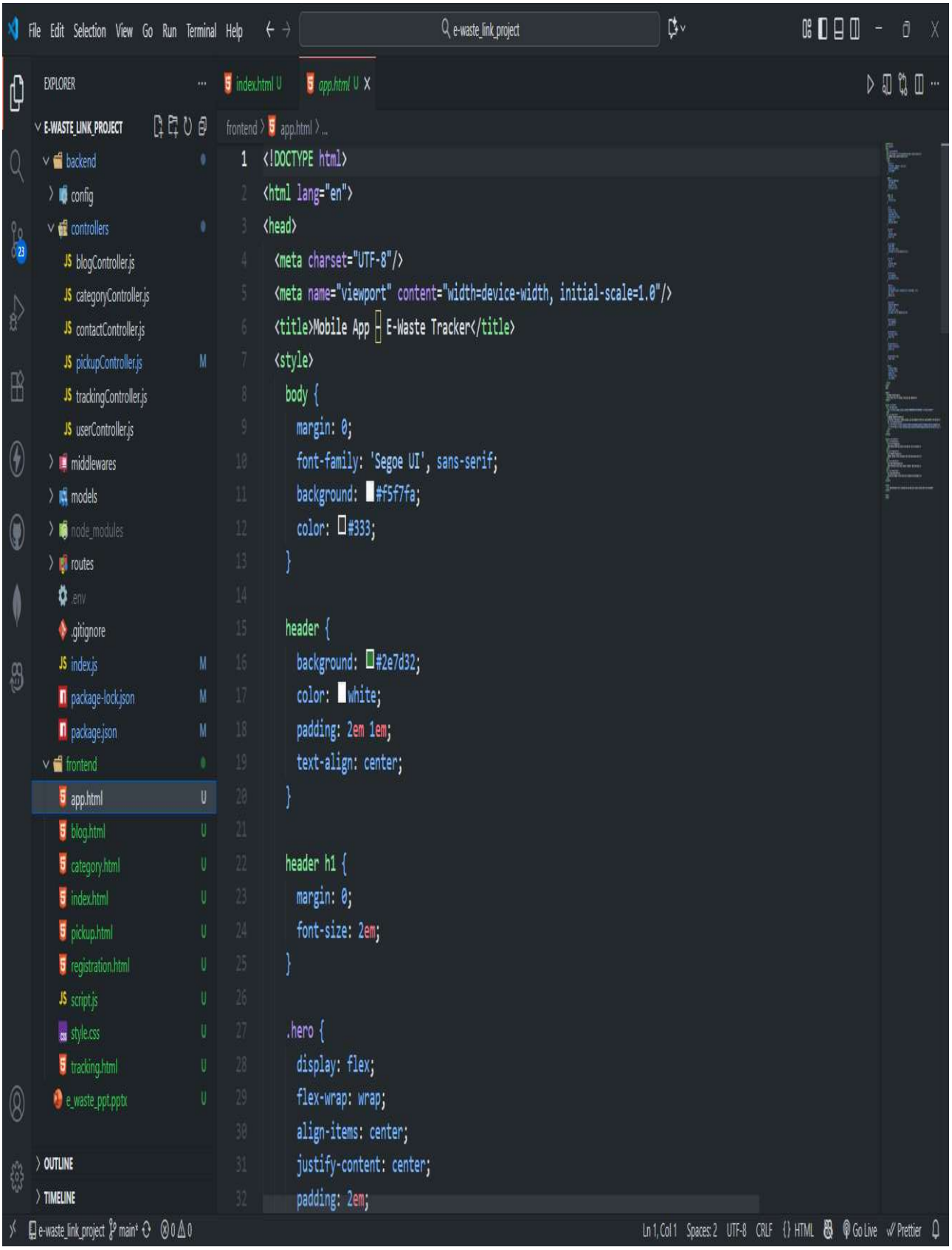
CODE

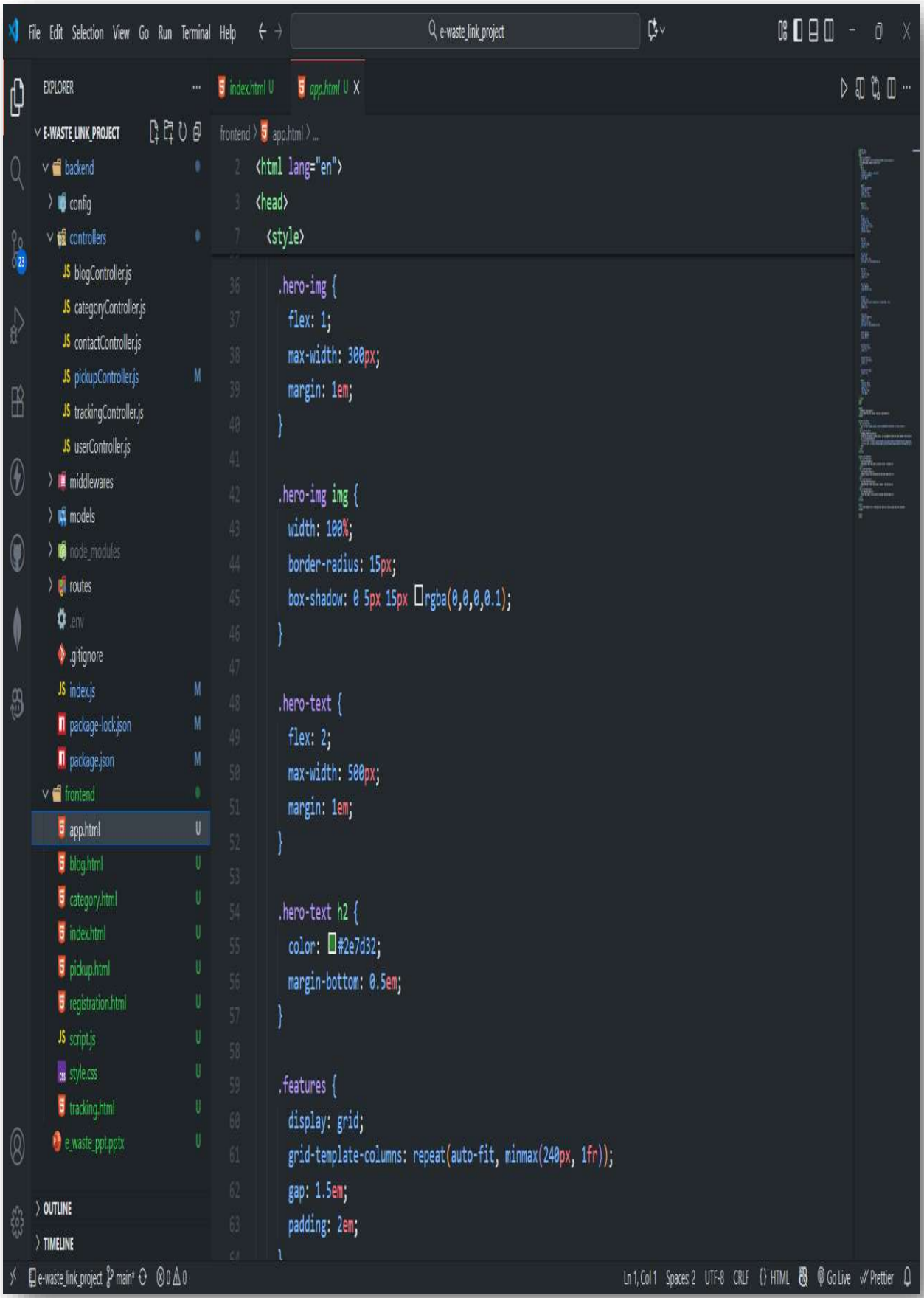

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
index.html U
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
frontend > index.html > ...
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>E-Waste Link | Smart E-Waste Management</title>
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
<link rel="stylesheet" href="style.css">
<link rel="stylesheet" href="script.js">
</head>
<body>
<!-- Navbar -->
<header class="navbar">

<h1 class="logo">E-Waste Link</h1>
<nav class="nav-links">
Home
Features
How It Works
Contact
</nav>

</header>
<!-- Hero Section -->
<section class="hero">
<h2>Revolutionizing E-Waste Management</h2>
Track. Recycle. Empower. From households to recyclers – a digital link for responsible disposal.
Explore Features
</section>
<!-- Features Section -->
<section id="features" class="features-section">
<h3>Platform Highlights</h3>

<h4>Mobile App</h4>
Schedule pickups, scan devices, track history – all from your phone.
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier
14°C Clear
Search
ENG IN
8:08 AM 11/8/2025


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
index.html U
app.html U
E-WASTE_LINK_PROJECT
frontend > app.html > ...
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8"/>
5 <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6 <title>Mobile App – E-Waste Tracker</title>
7 <style>
8 body {
9 margin: 0;
10 font-family: 'Segoe UI', sans-serif;
11 background: #f5f7fa;
12 color: #333;
13 }
14
15 header {
16 background: #2e7d32;
17 color: white;
18 padding: 2em 1em;
19 text-align: center;
20 }
21
22 header h1 {
23 margin: 0;
24 font-size: 2em;
25 }
26
27 .hero {
28 display: flex;
29 flex-wrap: wrap;
30 align-items: center;
31 justify-content: center;
32 padding: 2em;
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
index.html U
app.html U
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
frontend > app.html > ...
<html lang="en">
<head>
<style>
.hero-img {
flex: 1;
max-width: 300px;
margin: 1em;
}
.hero-img img {
width: 100%;
border-radius: 15px;
box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}
.hero-text {
flex: 2;
max-width: 500px;
margin: 1em;
}
.hero-text h2 {
color: #2e7d32;
margin-bottom: 0.5em;
}
.features {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
gap: 1.5em;
padding: 2em;
}
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
blog.html U
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
frontend > blog.html > html > body
<html lang="en">
<body>

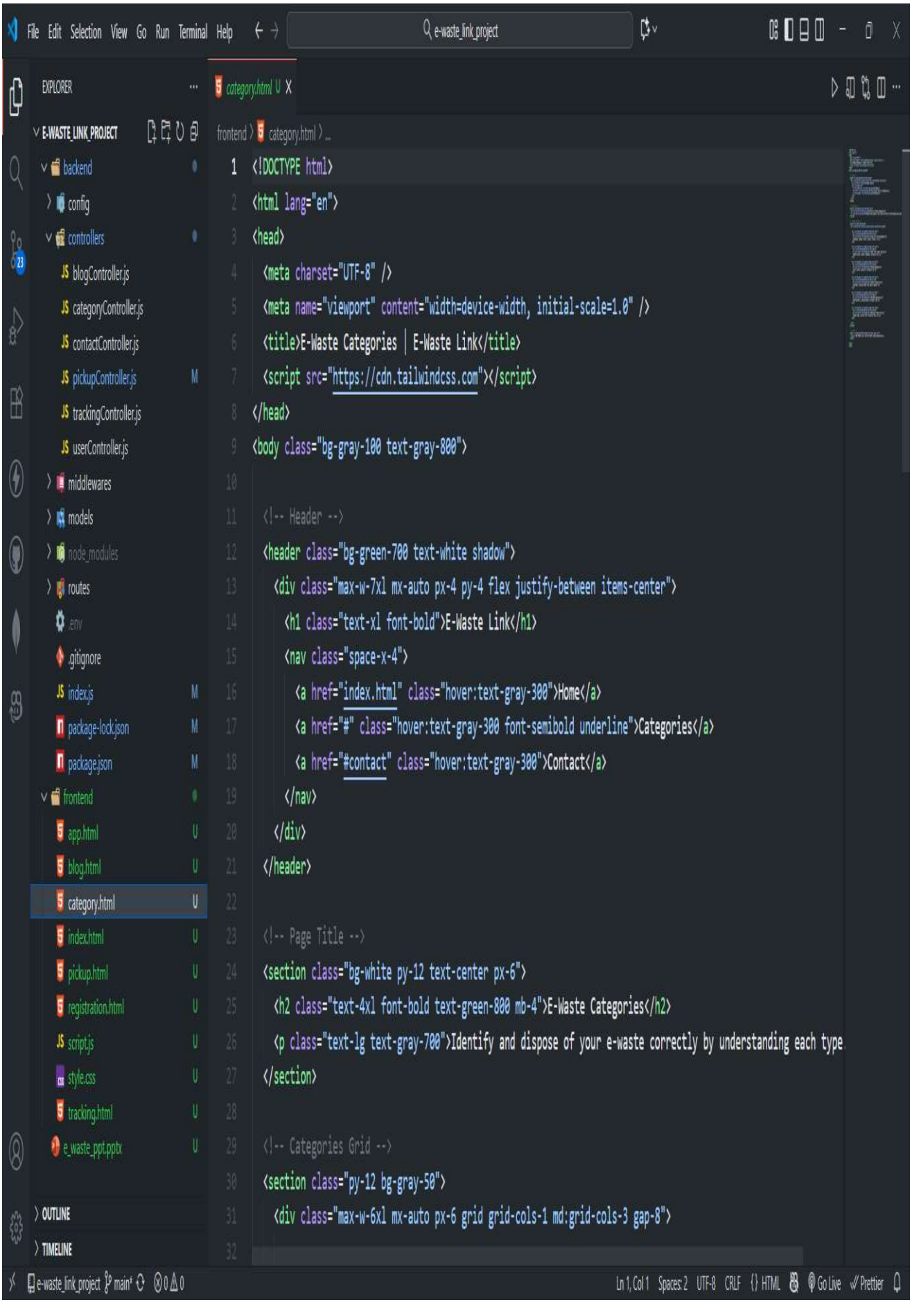
<div class="blog-image" style="background-image: url('https://images.unsplash.com/photo-1604697222767-1c

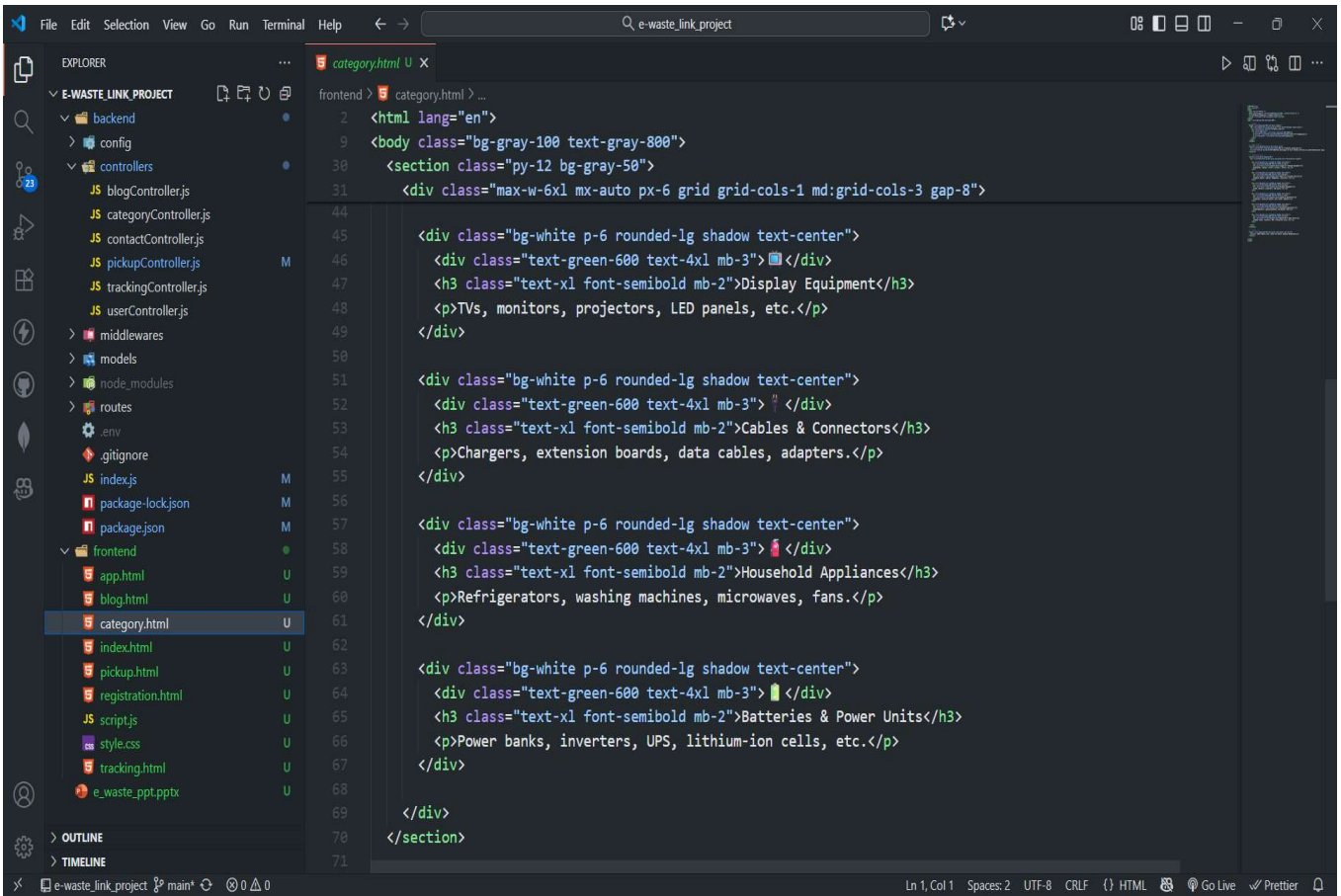
<h3 class="blog-title">Inside India's E-Waste Crisis</h3>
<p class="blog-snippet">A deep dive into how unregulated e-waste disposal is affecting health and envi

<div class="blog-image" style="background-image: url('https://images.unsplash.com/photo-1581092335379-0c

<h3 class="blog-title">How Schools Can Lead the E-Waste Movement</h3>
<p class="blog-snippet">From awareness programs to device donations, schools can be a hub of sustainab

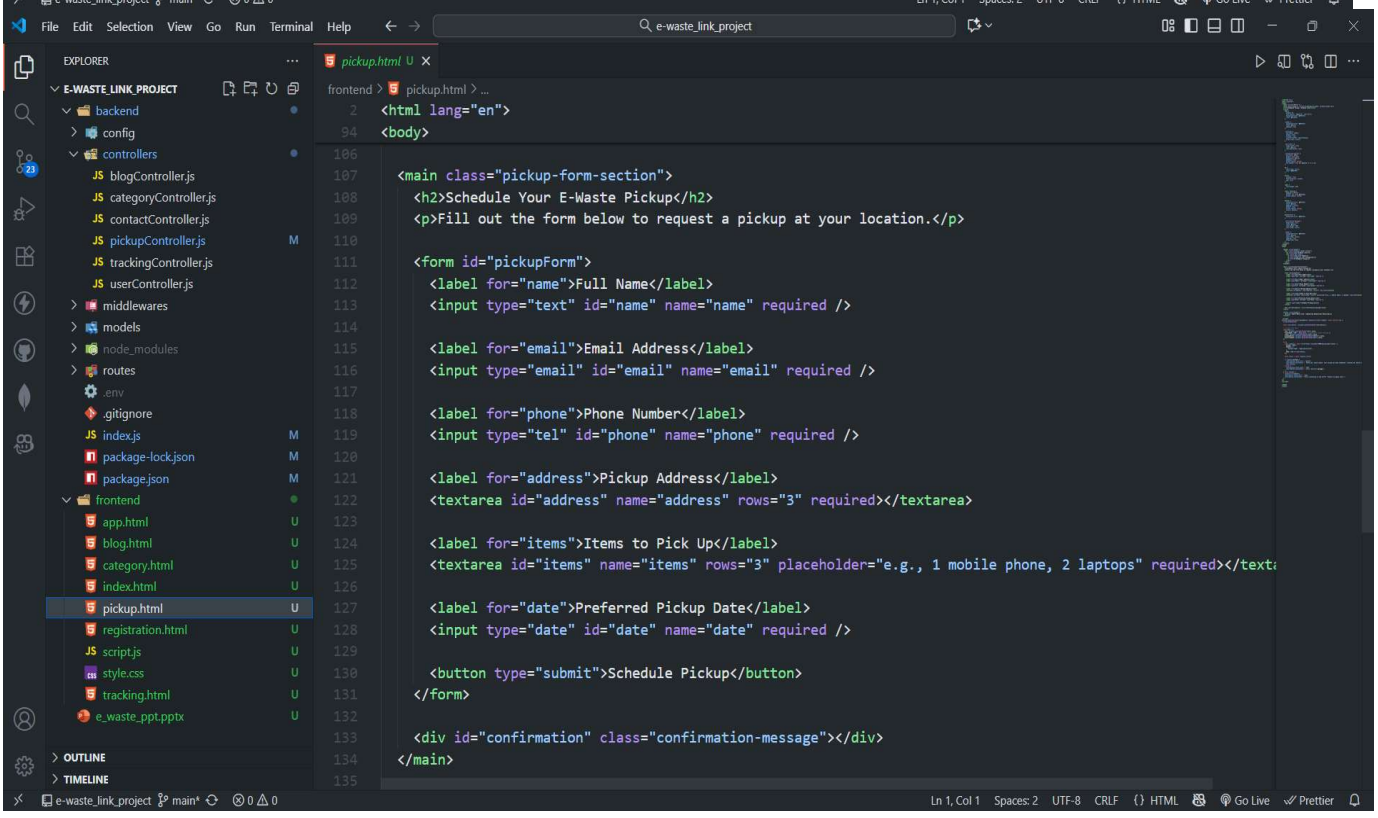
<footer>
© 2025 GreenCycle Tech | Spreading Awareness, Sparking Action.
</footer>
</body>
</html>
e-waste_link_project main* 0 0
Ln 101, Col 1 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
category.html U
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
frontend > category.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <title>E-Waste Categories | E-Waste Link</title>
7 <script src="https://cdn.tailwindcss.com"></script>
8 </head>
9 <body class="bg-gray-100 text-gray-800">
10
11 <!-- Header -->
12 <header class="bg-green-700 text-white shadow">
13
14 <h1 class="text-xl font-bold">E-Waste Link</h1>
15 <nav class="space-x-4">
16 Home
17 Categories
18 Contact
19 </nav>
20
21 </header>
22
23 <!-- Page Title -->
24 <section class="bg-white py-12 text-center px-6">
25 <h2 class="text-4xl font-bold text-green-800 mb-4">E-Waste Categories</h2>
26 <p class="text-lg text-gray-700">Identify and dispose of your e-waste correctly by understanding each type.
27 </section>
28
29 <!-- Categories Grid -->
30 <section class="py-12 bg-gray-50">
31
32
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier

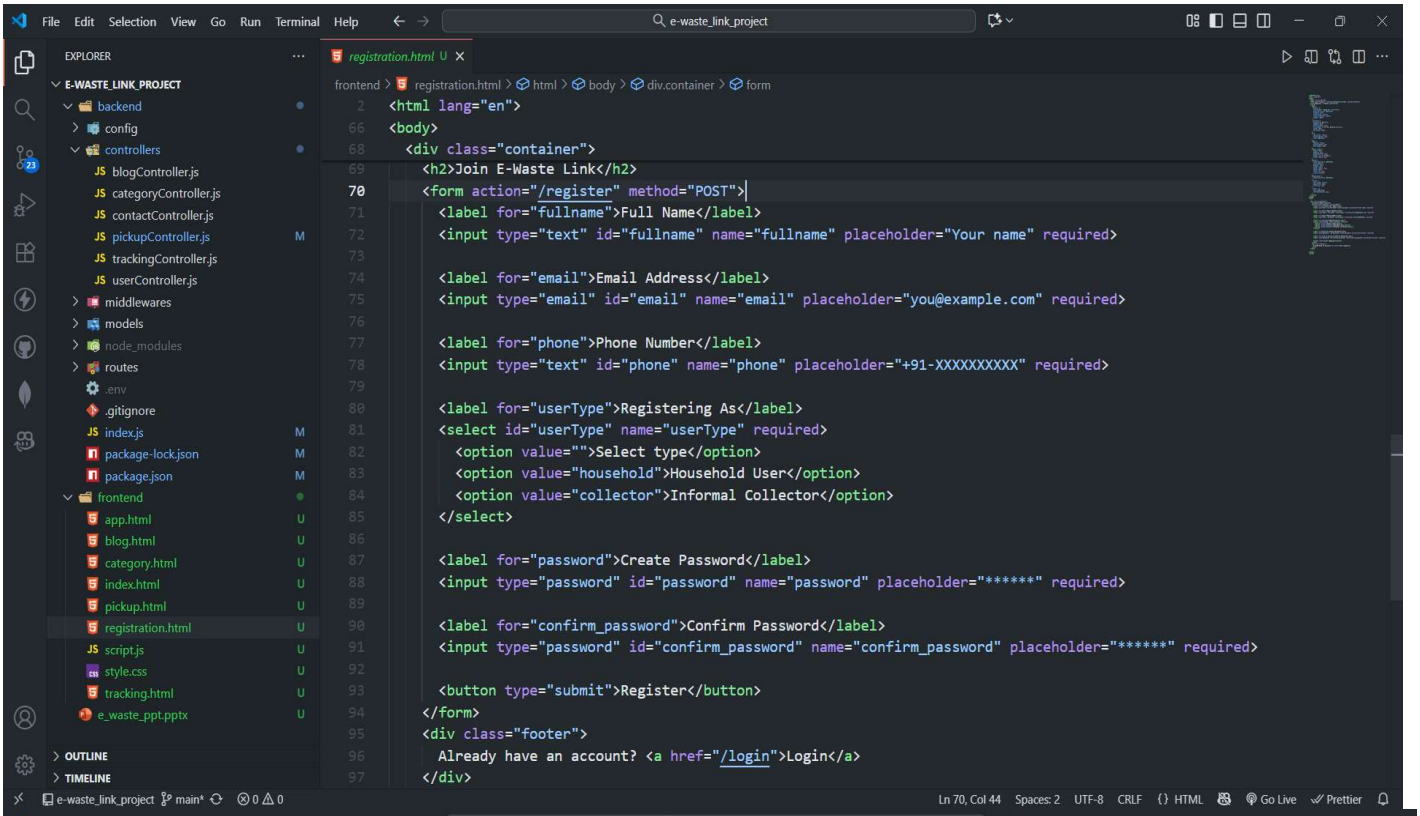

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
category.html U
frontend > category.html > ...
<html lang="en">
<body class="bg-gray-100 text-gray-800">
<section class="py-12 bg-gray-50">

<h3 class="text-xl font-semibold mb-2">Display Equipment</h3>
TVs, monitors, projectors, LED panels, etc.
<h3 class="text-xl font-semibold mb-2">Cables & Connectors</h3>
Chargers, extension boards, data cables, adapters.
<h3 class="text-xl font-semibold mb-2">Household Appliances</h3>
Refrigerators, washing machines, microwaves, fans.
<h3 class="text-xl font-semibold mb-2">Batteries & Power Units</h3>
Power banks, inverters, UPS, lithium-ion cells, etc.

</section>
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
pickup.html U
frontend > pickup.html > ...
<html lang="en">
<body>
<main class="pickup-form-section">
<h2>Schedule Your E-Waste Pickup</h2>
Fill out the form below to request a pickup at your location.
<form id="pickupForm">
<label for="name">Full Name</label>
<input type="text" id="name" name="name" required />
<label for="email">Email Address</label>
<input type="email" id="email" name="email" required />
<label for="phone">Phone Number</label>
<input type="tel" id="phone" name="phone" required />
<label for="address">Pickup Address</label>
<textarea id="address" name="address" rows="3" required></textarea>
<label for="items">Items to Pick Up</label>
<textarea id="items" name="items" rows="3" placeholder="e.g., 1 mobile phone, 2 laptops" required></texta
<label for="date">Preferred Pickup Date</label>
<input type="date" id="date" name="date" required />
<button type="submit">Schedule Pickup</button>
</form>

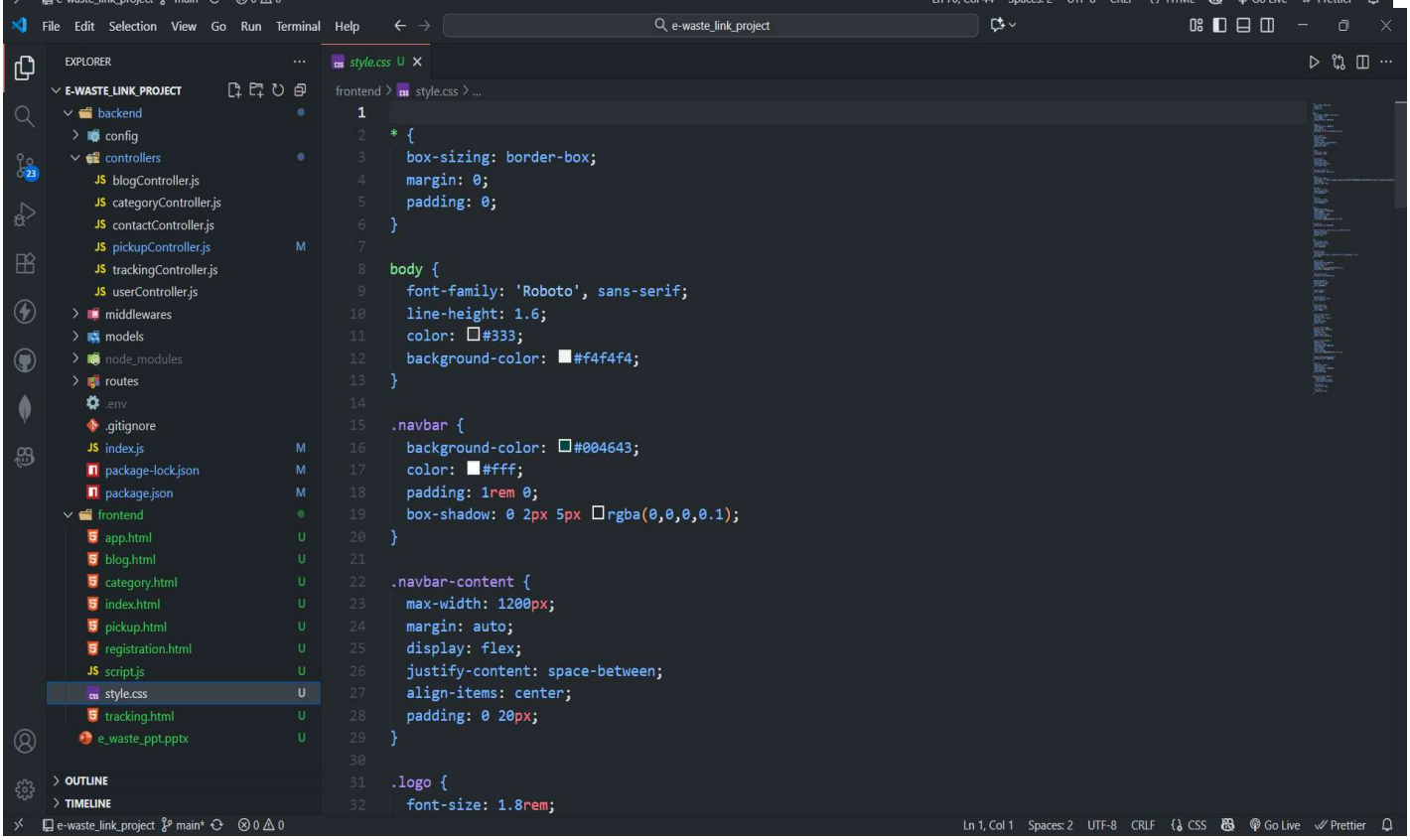
</main>
e-waste_link_project main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier

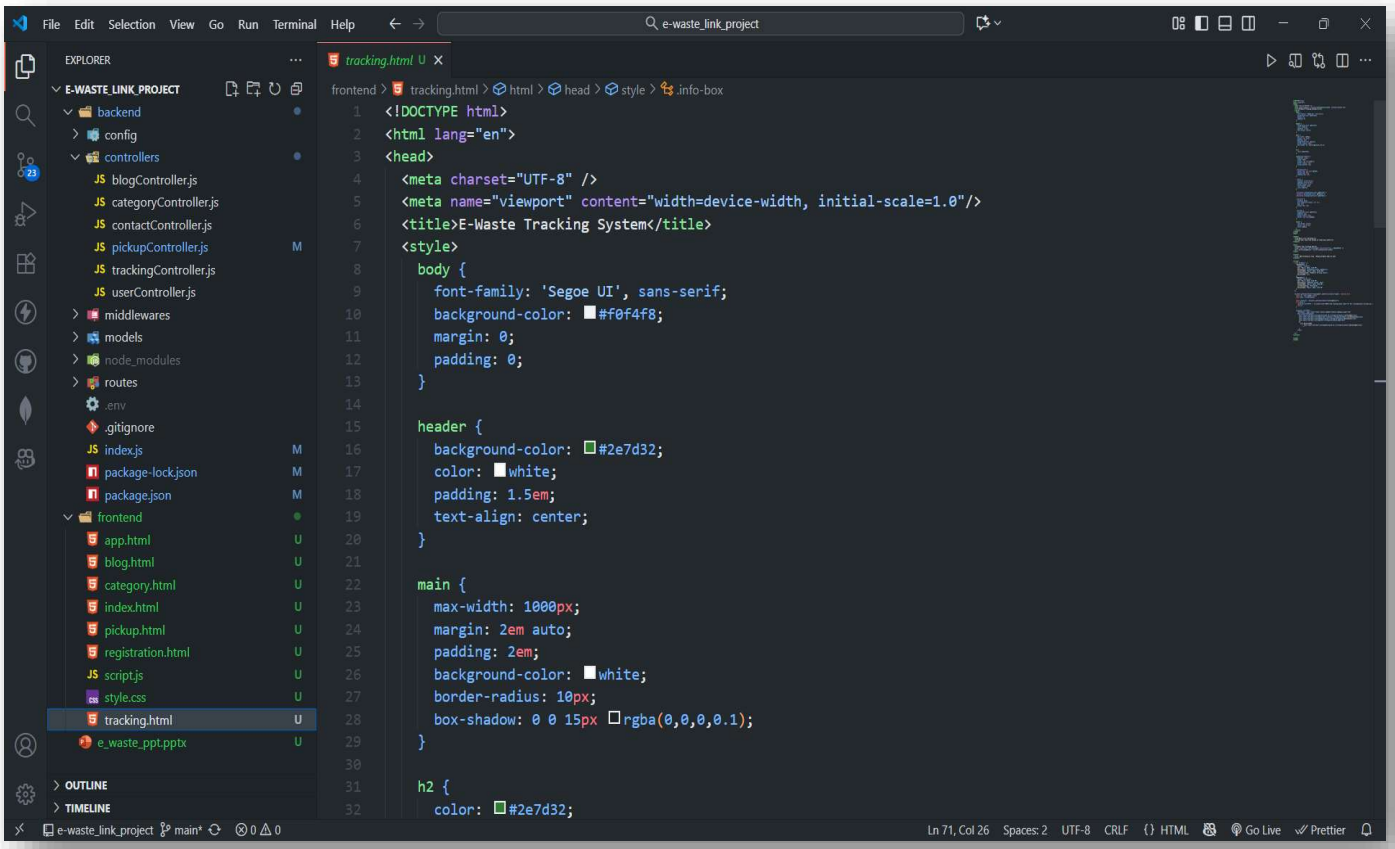

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
registration.html U
frontend > registration.html > html > body > div.container > form
<html lang="en">
<body>

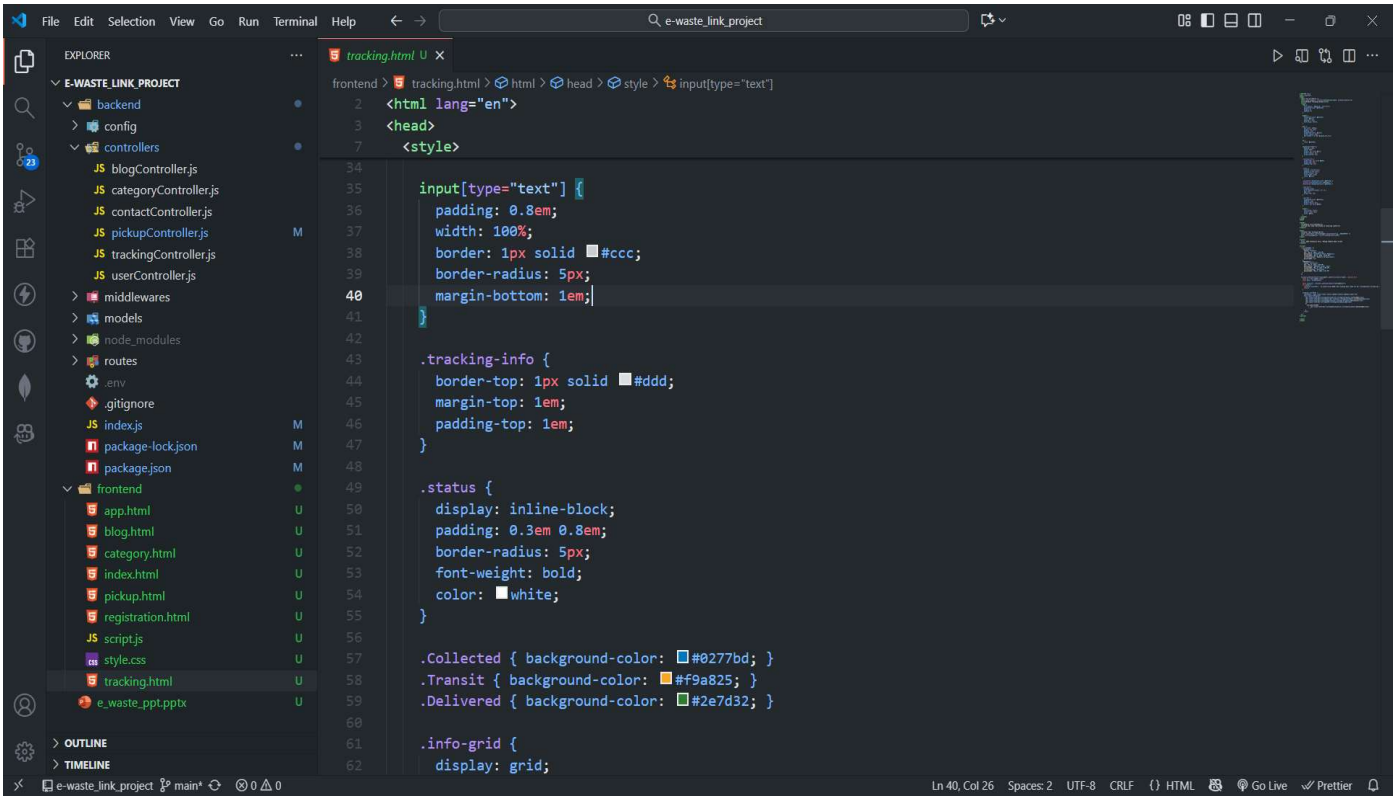
<h2>Join E-Waste Link</h2>
<form action="/register" method="POST">
<label for="fullname">Full Name</label>
<input type="text" id="fullname" name="fullname" placeholder="Your name" required>
<label for="email">Email Address</label>
<input type="email" id="email" name="email" placeholder="you@example.com" required>
<label for="phone">Phone Number</label>
<input type="text" id="phone" name="phone" placeholder="+91-XXXXXXXXXX" required>
<label for="userType">Registering As</label>
<select id="userType" name="userType" required>
<option value="">Select type</option>
<option value="household">Household User</option>
<option value="collector">Informal Collector</option>
</select>
<label for="password">Create Password</label>
<input type="password" id="password" name="password" placeholder="******" required>
<label for="confirm_password">Confirm Password</label>
<input type="password" id="confirm_password" name="confirm_password" placeholder="******" required>
<button type="submit">Register</button>
</form>

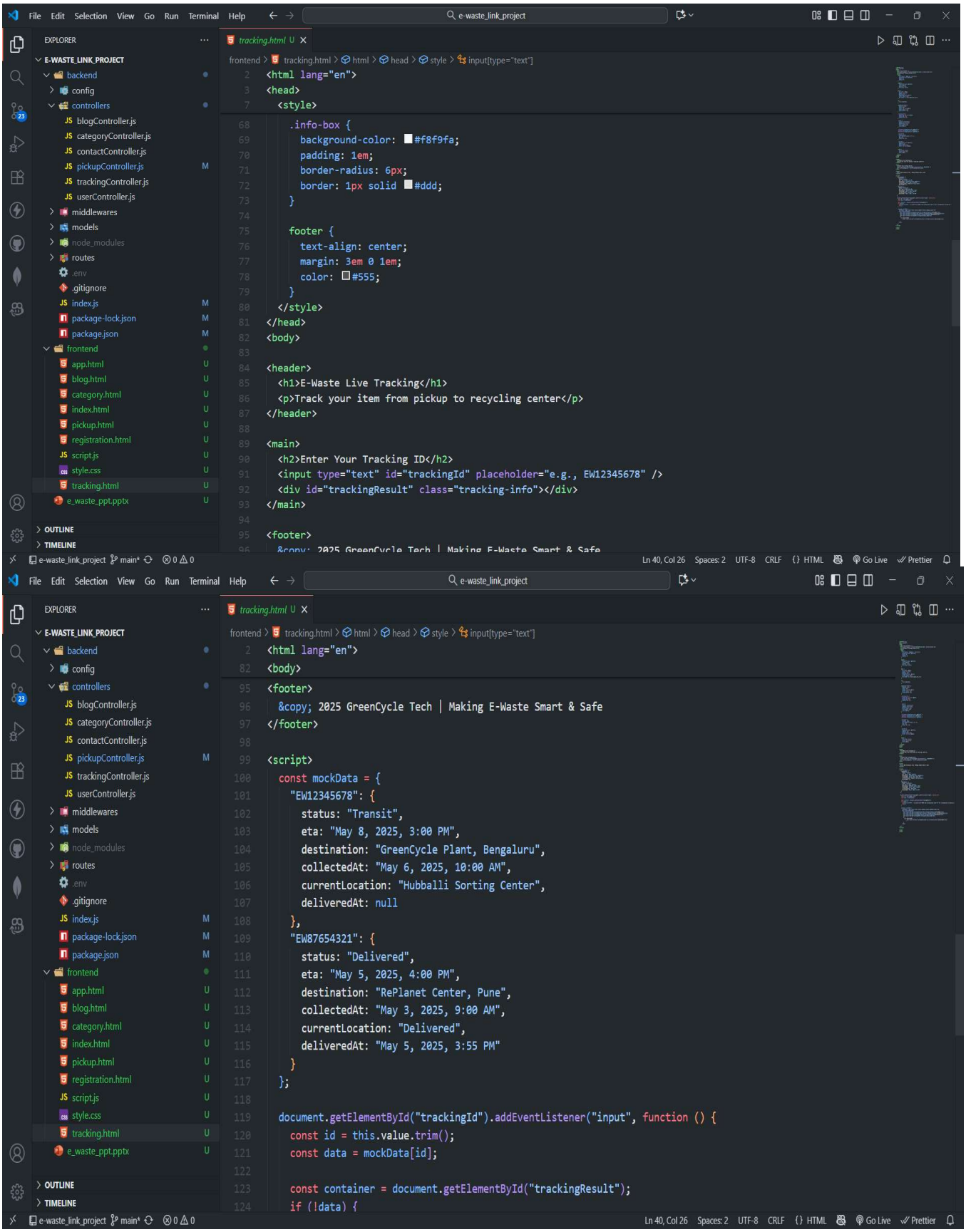
Already have an account? Login

Ln 70, Col 44 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
style.css U
frontend > style.css > ...
* {
box-sizing: border-box;
margin: 0;
padding: 0;
}
body {
font-family: 'Roboto', sans-serif;
line-height: 1.6;
color: #333;
background-color: #f4f4f4;
}
.navbar {
background-color: #004643;
color: #fff;
padding: 1rem 0;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
.navbar-content {
max-width: 1200px;
margin: auto;
display: flex;
justify-content: space-between;
align-items: center;
padding: 0 20px;
}
.logo {
font-size: 1.8rem;
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF CSS Go Live Prettier

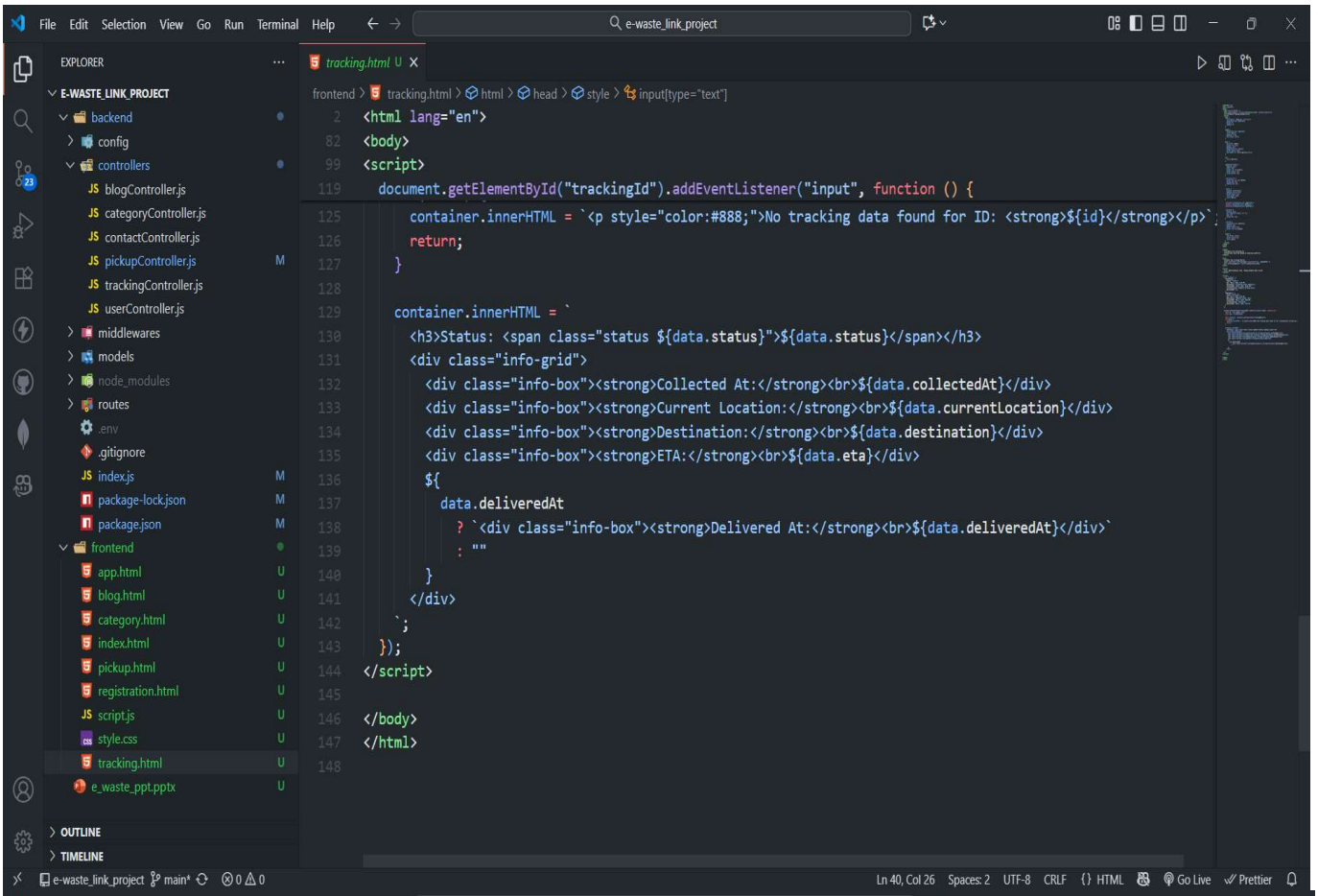

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
tracking.html U
frontend > tracking.html > html > head > style > .info-box
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<title>E-Waste Tracking System</title>
<style>
body {
font-family: 'Segoe UI', sans-serif;
background-color: #f0f4f8;
margin: 0;
padding: 0;
}
header {
background-color: #2e7d32;
color: white;
padding: 1.5em;
text-align: center;
}
main {
max-width: 1000px;
margin: 2em auto;
padding: 2em;
background-color: white;
border-radius: 10px;
box-shadow: 0 0 15px rgba(0,0,0,0.1);
}
h2 {
color: #2e7d32;
e-waste_link_project main* 0 0
Ln 71, Col 26 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
tracking.html U
frontend > tracking.html > html > head > style > input[type="text"]
<html lang="en">
<head>
<style>
input[type="text"] {
padding: 0.8em;
width: 100%;
border: 1px solid #ccc;
border-radius: 5px;
margin-bottom: 1em;
}
.tracking-info {
border-top: 1px solid #ddd;
margin-top: 1em;
padding-top: 1em;
}
.status {
display: inline-block;
padding: 0.3em 0.8em;
border-radius: 5px;
font-weight: bold;
color: white;
}
.Collected { background-color: #0277bd; }
.Transit { background-color: #f9a825; }
.Delivered { background-color: #2e7d32; }
.info-grid {
display: grid;
e-waste_link_project main* 0 0
Ln 40, Col 26 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
tracking.html U
frontend > tracking.html > html > head > style > input[type="text"]
<html lang="en">
<head>
<style>
.info-box {
background-color: #f8f9fa;
padding: 1em;
border-radius: 6px;
border: 1px solid #ddd;
}
footer {
text-align: center;
margin: 3em 0 1em;
color: #555;
}
</style>
</head>
<body>
<header>
<h1>E-Waste Live Tracking</h1>
Track your item from pickup to recycling center
</header>
<main>
<h2>Enter Your Tracking ID</h2>
<input type="text" id="trackingId" placeholder="e.g., EW12345678" />

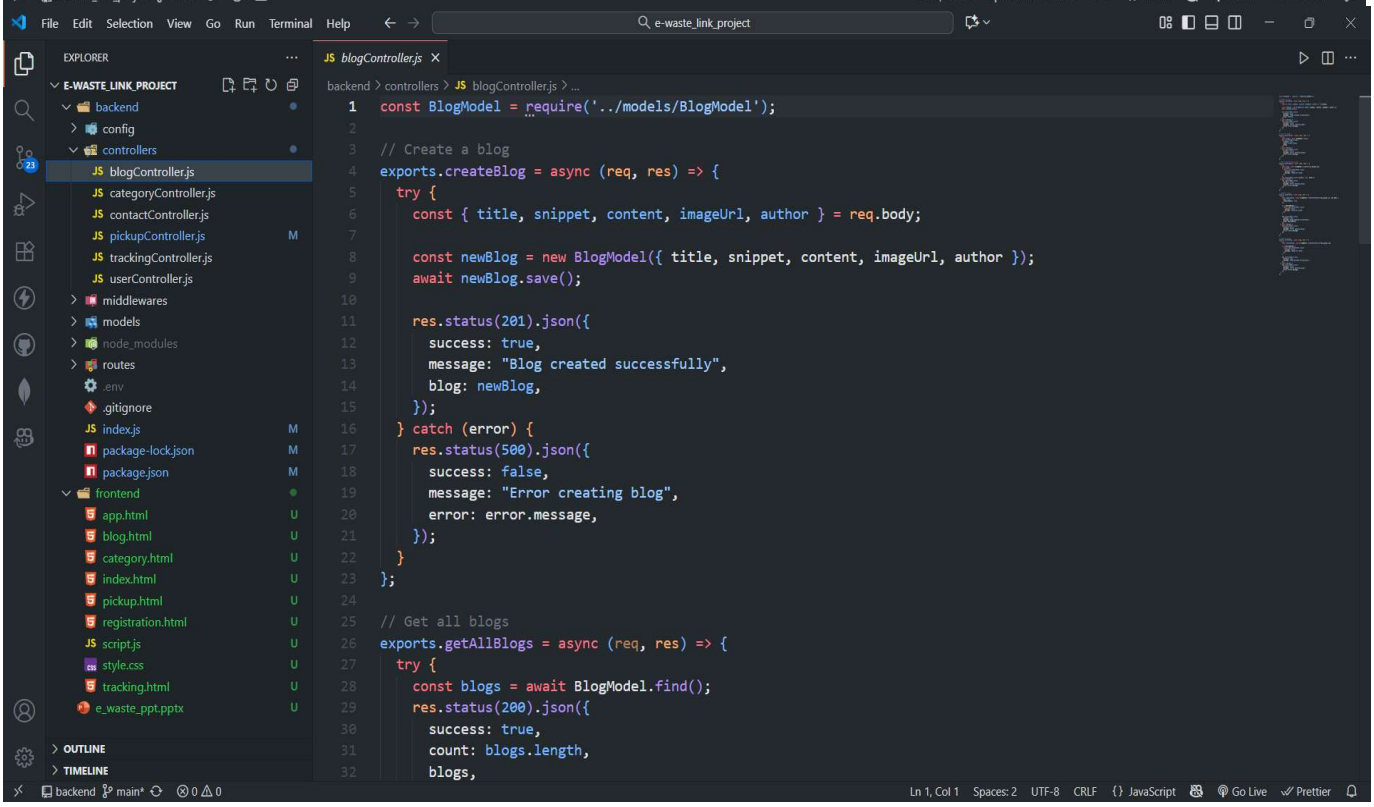
</main>
<footer>
Ln 40, Col 26 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier

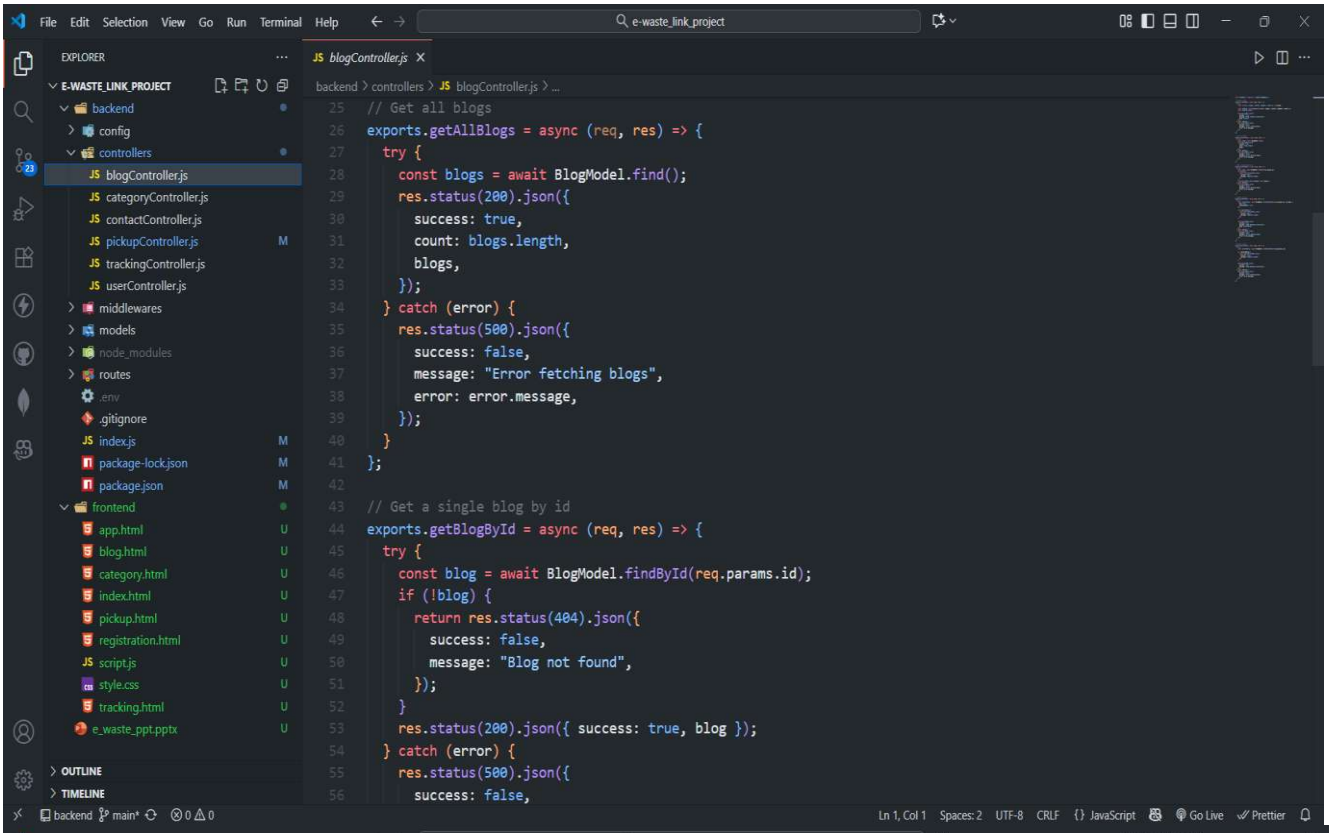
File Edit Selection View Go Run Terminal Help
e-waste_link_project
tracking.html U
frontend > tracking.html > html > head > style > input[type="text"]
<html lang="en">
<body>
<footer>
© 2025 GreenCycle Tech | Making E-Waste Smart & Safe
</footer>
<script>
const mockData = {
"EW12345678": {
status: "Transit",
eta: "May 8, 2025, 3:00 PM",
destination: "GreenCycle Plant, Bengaluru",
collectedAt: "May 6, 2025, 10:00 AM",
currentLocation: "Hubballi Sorting Center",
deliveredAt: null
},
"EW87654321": {
status: "Delivered",
eta: "May 5, 2025, 4:00 PM",
destination: "RePlanet Center, Pune",
collectedAt: "May 3, 2025, 9:00 AM",
currentLocation: "Delivered",
deliveredAt: "May 5, 2025, 3:55 PM"
}
};
document.getElementById("trackingId").addEventListener("input", function () {
const id = this.value.trim();
const data = mockData[id];
const container = document.getElementById("trackingResult");
if (!data) {
Ln 40, Col 26 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier

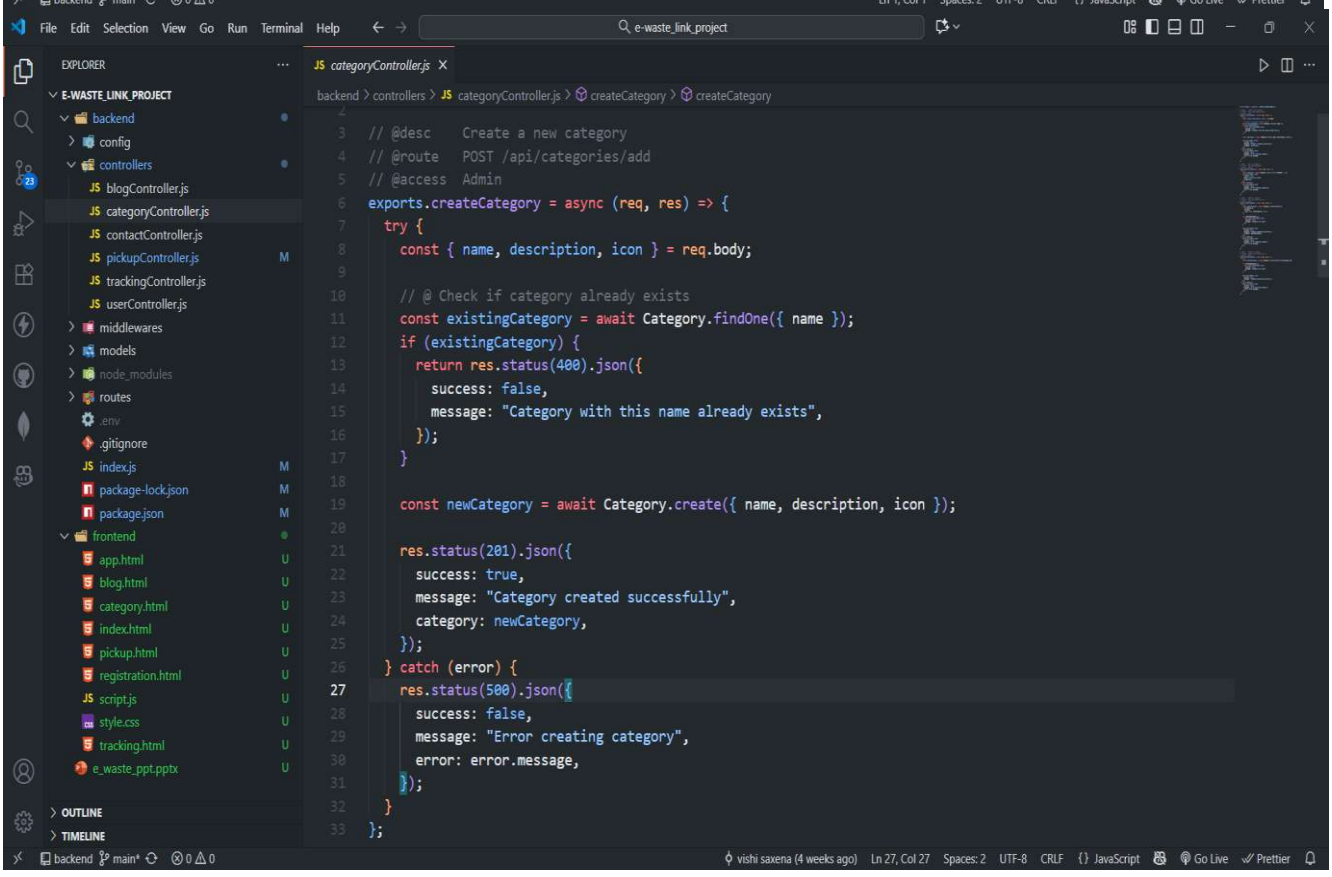

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
tracking.html U
frontend > tracking.html > html > head > style > input[type="text"]
<html lang="en">
<body>
<script>
document.getElementById("trackingId").addEventListener("input", function () {
container.innerHTML = `<p style="color:#888;">No tracking data found for ID: <strong>${id}</strong></p>`;
return;
}
container.innerHTML = `
<h3>Status: <span class="status ${data.status}">${data.status}</h3>

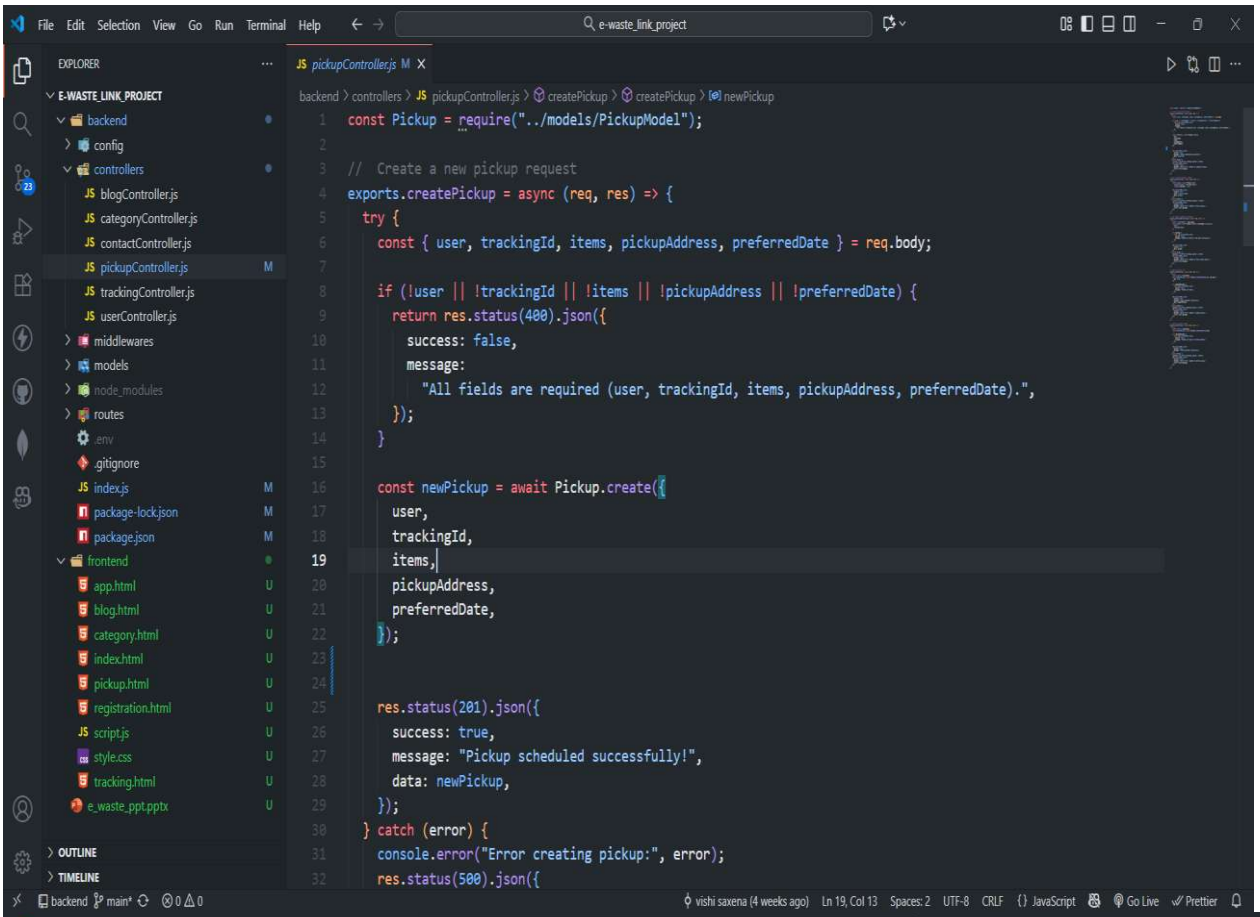
Collected At:
${data.collectedAt}
Current Location:
${data.currentLocation}
Destination:
${data.destination}
ETA:
${data.eta}
${
data.deliveredAt
? `<div class="info-box"><strong>Delivered At:</strong><br>${data.deliveredAt}</div>`
: ""
}

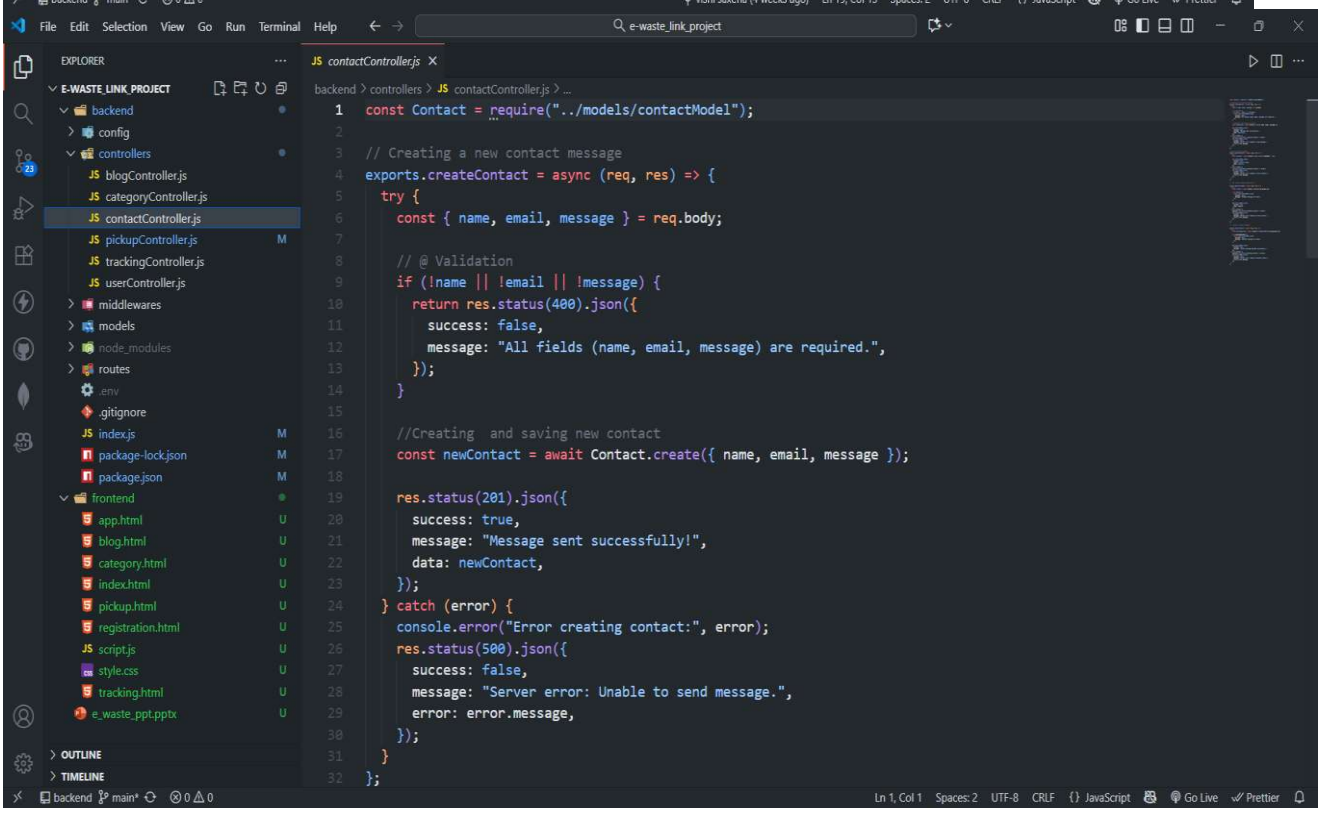
`;
});
</script>
</body>
</html>
e-waste_link_project main* 0 0
Ln 40, Col 26 Spaces: 2 UTF-8 CRLF {} HTML Go Live Prettier

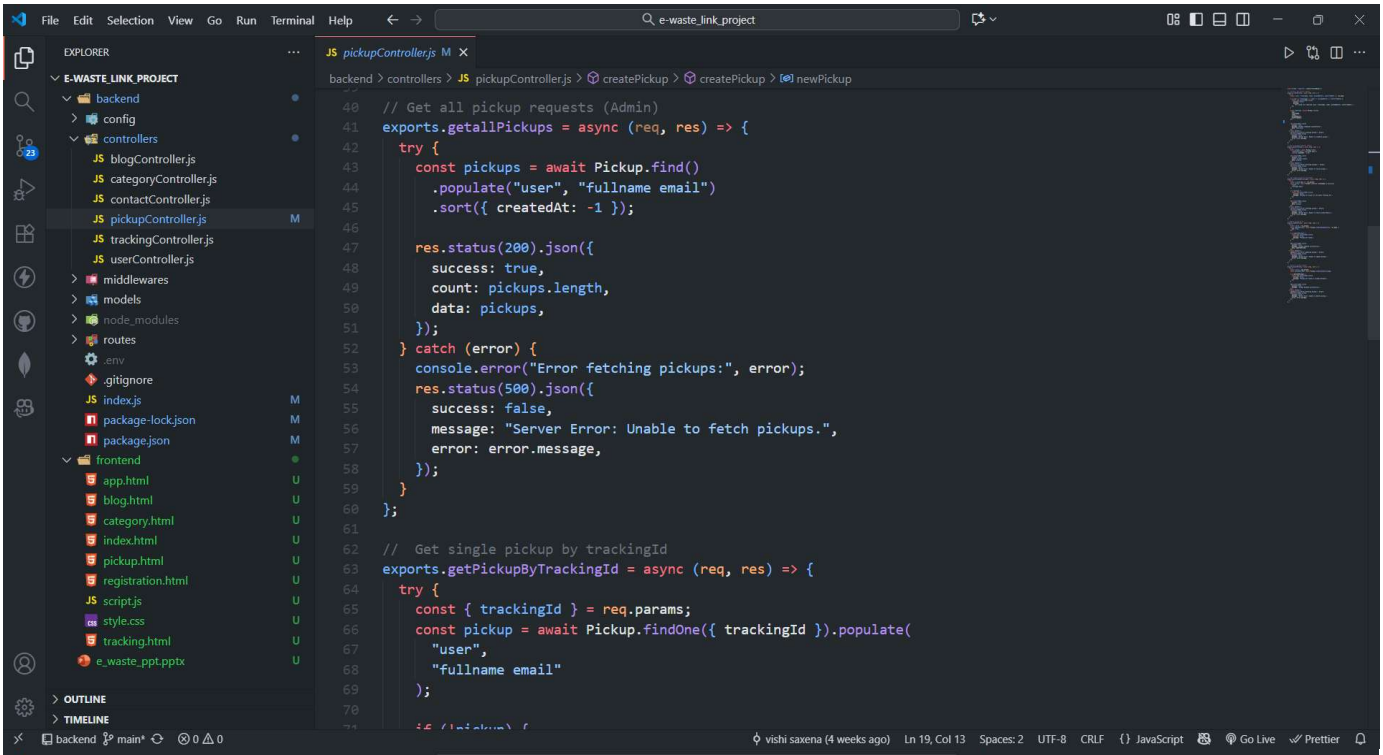

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
blogController.js
backend > controllers > blogController.js > ...
const BlogModel = require('../models/BlogModel');
// Create a blog
exports.createBlog = async (req, res) => {
try {
const { title, snippet, content, imageUrl, author } = req.body;
const newBlog = new BlogModel({ title, snippet, content, imageUrl, author });
await newBlog.save();
res.status(201).json({
success: true,
message: "Blog created successfully",
blog: newBlog,
});
} catch (error) {
res.status(500).json({
success: false,
message: "Error creating blog",
error: error.message,
});
}
};
// Get all blogs
exports.getAllBlogs = async (req, res) => {
try {
const blogs = await BlogModel.find();
res.status(200).json({
success: true,
count: blogs.length,
blogs,
backend main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier

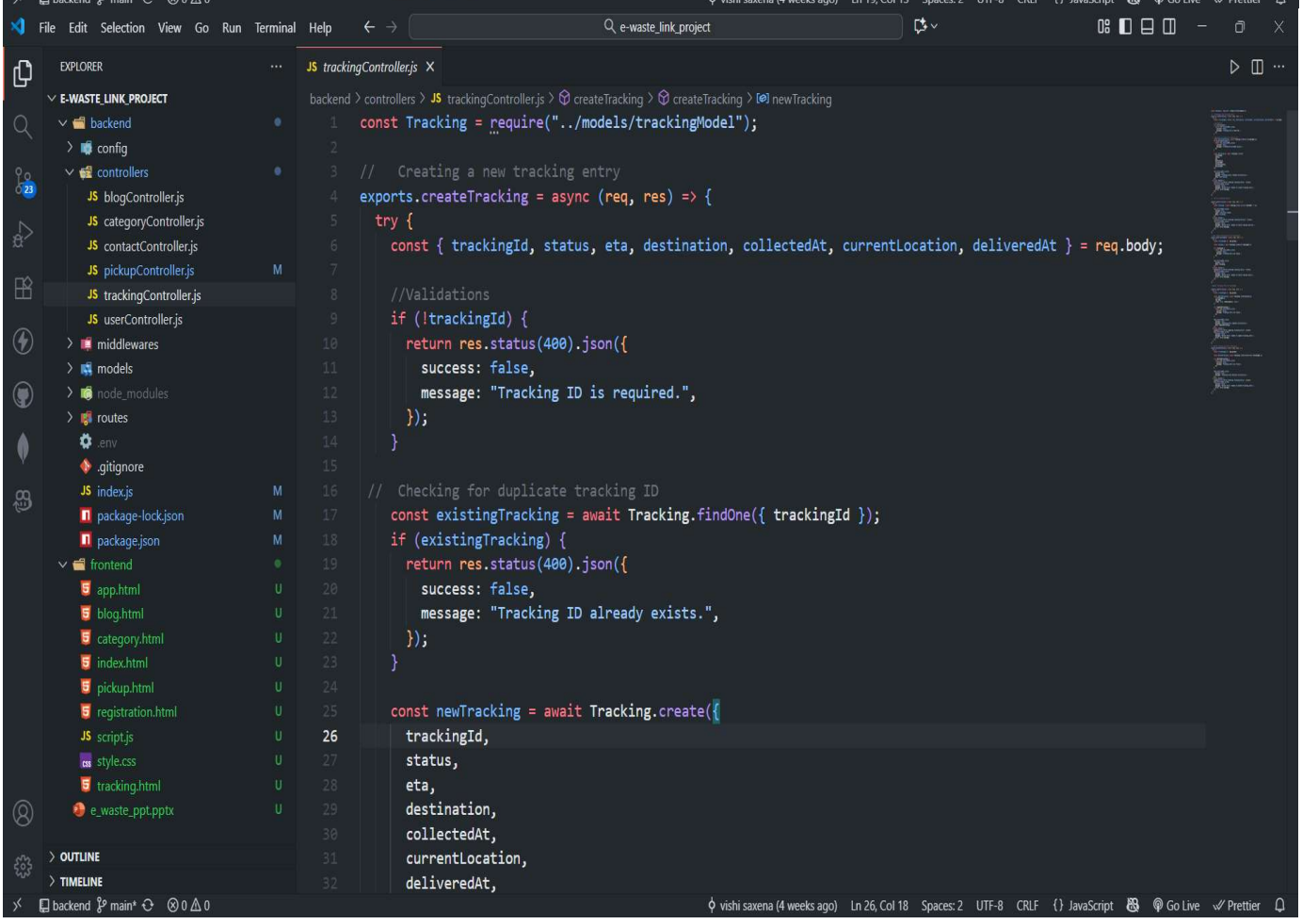

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
blogController.js
backend > controllers > blogController.js > ...
// Get all blogs
exports.getAllBlogs = async (req, res) => {
try {
const blogs = await BlogModel.find();
res.status(200).json({
success: true,
count: blogs.length,
blogs,
});
} catch (error) {
res.status(500).json({
success: false,
message: "Error fetching blogs",
error: error.message,
});
}
};
// Get a single blog by id
exports.getBlogById = async (req, res) => {
try {
const blog = await BlogModel.findById(req.params.id);
if (!blog) {
return res.status(404).json({
success: false,
message: "Blog not found",
});
}
res.status(200).json({ success: true, blog });
} catch (error) {
res.status(500).json({
success: false,
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier

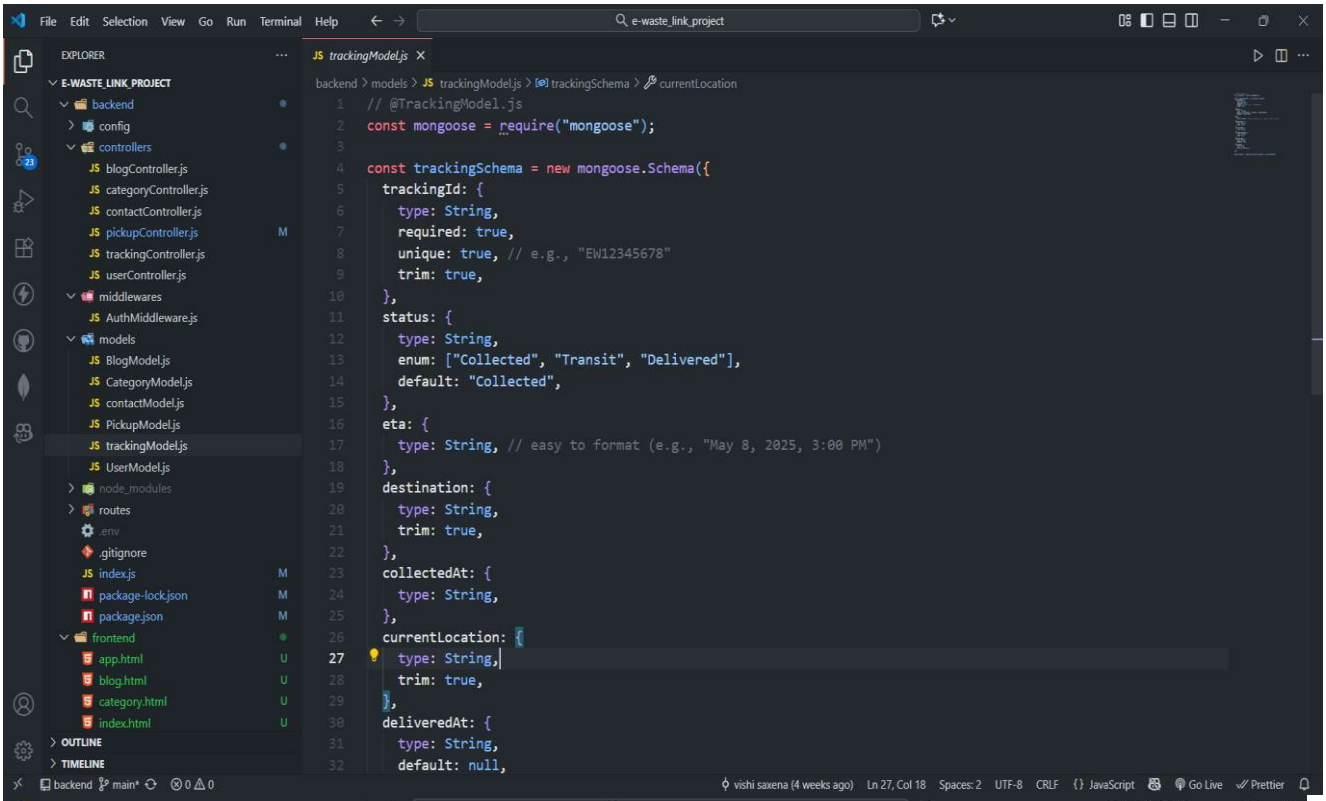

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
categoryController.js
backend > controllers > categoryController.js > createCategory > createCategory
// @desc Create a new category
// @route POST /api/categories/add
// @access Admin
exports.createCategory = async (req, res) => {
try {
const { name, description, icon } = req.body;
// @ Check if category already exists
const existingCategory = await Category.findOne({ name });
if (existingCategory) {
return res.status(400).json({
success: false,
message: "Category with this name already exists",
});
}
const newCategory = await Category.create({ name, description, icon });
res.status(201).json({
success: true,
message: "Category created successfully",
category: newCategory,
});
} catch (error) {
res.status(500).json({
success: false,
message: "Error creating category",
error: error.message,
});
}
};
backend main* 0 0 vishi saxena (4 weeks ago) Ln 27, Col 27 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier

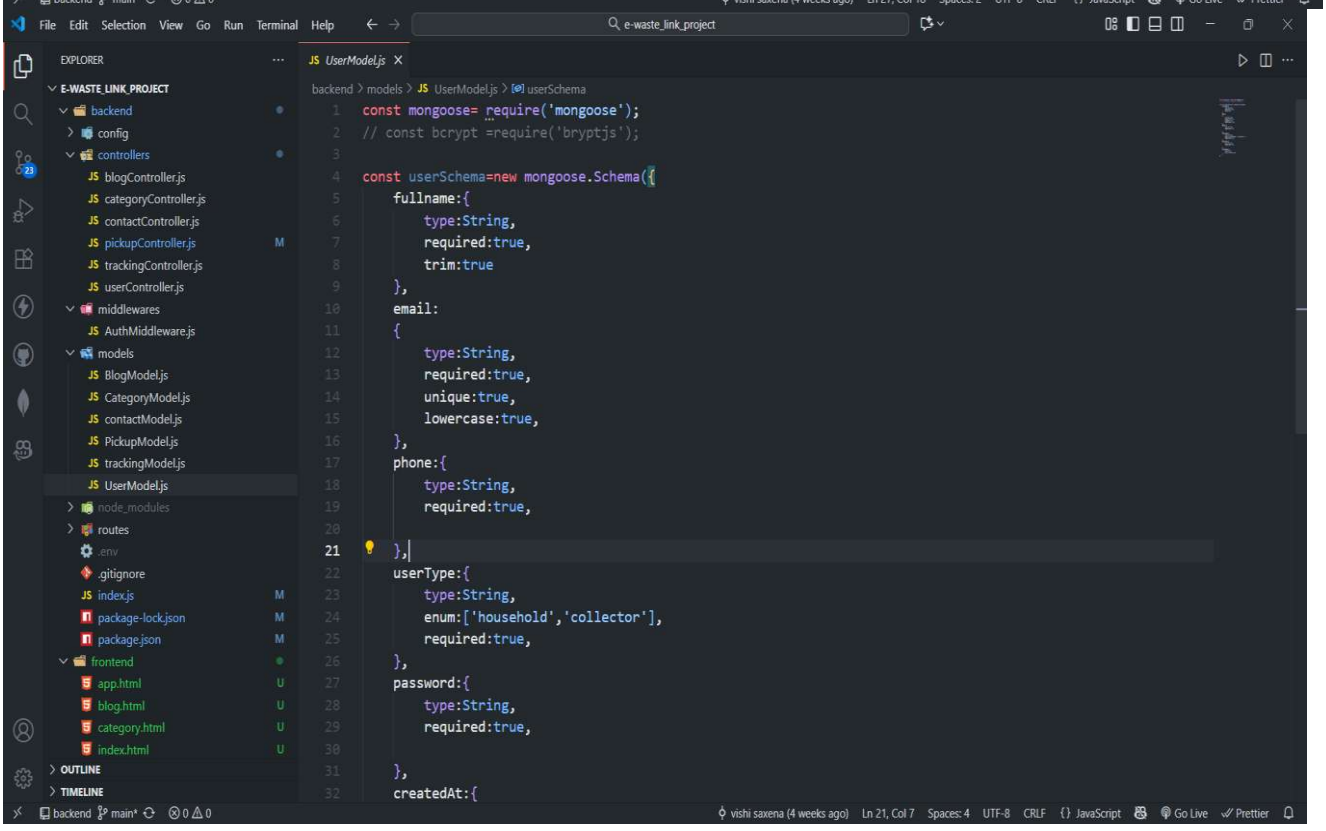

File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
pickupController.js M
backend > controllers > pickupController.js > createPickup > createPickup > newPickup
const Pickup = require("../models/PickupModel");
// Create a new pickup request
exports.createPickup = async (req, res) => {
try {
const { user, trackingId, items, pickupAddress, preferredDate } = req.body;
if (!user || !trackingId || !items || !pickupAddress || !preferredDate) {
return res.status(400).json({
success: false,
message:
"All fields are required (user, trackingId, items, pickupAddress, preferredDate).",
});
}
const newPickup = await Pickup.create({
user,
trackingId,
items,
pickupAddress,
preferredDate,
});
res.status(201).json({
success: true,
message: "Pickup scheduled successfully!",
data: newPickup,
});
} catch (error) {
console.error("Error creating pickup:", error);
res.status(500).json({
backend main* 0 0
vishi saxena (4 weeks ago) Ln 19, Col 13 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
pickup.html
registration.html
script.js
style.css
tracking.html
e_waste_ppt.pptx
OUTLINE
TIMELINE
contactController.js
backend > controllers > contactController.js > ...
const Contact = require("../models/contactModel");
// Creating a new contact message
exports.createContact = async (req, res) => {
try {
const { name, email, message } = req.body;
// @ Validation
if (!name || !email || !message) {
return res.status(400).json({
success: false,
message: "All fields (name, email, message) are required.",
});
}
//Creating and saving new contact
const newContact = await Contact.create({ name, email, message });
res.status(201).json({
success: true,
message: "Message sent successfully!",
data: newContact,
});
} catch (error) {
console.error("Error creating contact:", error);
res.status(500).json({
success: false,
message: "Server error: Unable to send message.",
error: error.message,
});
}
};
backend main* 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
pickupController.js M
backend > controllers > pickupController.js > createPickup > createPickup > newPickup
// Get all pickup requests (Admin)
exports.getallPickups = async (req, res) => {
try {
const pickups = await Pickup.find()
.populate("user", "fullname email")
.sort({ createdAt: -1 });
res.status(200).json({
success: true,
count: pickups.length,
data: pickups,
});
} catch (error) {
console.error("Error fetching pickups:", error);
res.status(500).json({
success: false,
message: "Server Error: Unable to fetch pickups.",
error: error.message,
});
}
};
// Get single pickup by trackingId
exports.getPickupByTrackingId = async (req, res) => {
try {
const { trackingId } = req.params;
const pickup = await Pickup.findOne({ trackingId }).populate(
"user",
"fullname email"
);
backend main* 0 0
vishi saxena (4 weeks ago) Ln 19, Col 13 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js M
trackingController.js
userController.js
middlewares
models
node_modules
routes
.env
.gitignore
index.js M
package-lock.json M
package.json M
frontend
app.html U
blog.html U
category.html U
index.html U
pickup.html U
registration.html U
script.js U
style.css U
tracking.html U
e_waste_ppt.pptx U
OUTLINE
TIMELINE
trackingController.js
backend > controllers > trackingController.js > createTracking > createTracking > newTracking
const Tracking = require("../models/trackingModel");
// Creating a new tracking entry
exports.createTracking = async (req, res) => {
try {
const { trackingId, status, eta, destination, collectedAt, currentLocation, deliveredAt } = req.body;
//Validations
if (!trackingId) {
return res.status(400).json({
success: false,
message: "Tracking ID is required.",
});
}
// Checking for duplicate tracking ID
const existingTracking = await Tracking.findOne({ trackingId });
if (existingTracking) {
return res.status(400).json({
success: false,
message: "Tracking ID already exists.",
});
}
const newTracking = await Tracking.create({
trackingId,
status,
eta,
destination,
collectedAt,
currentLocation,
deliveredAt,
backend main* 0 0
vishi saxena (4 weeks ago) Ln 26, Col 18 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
AuthMiddleware.js
models
BlogModel.js
CategoryModel.js
contactModel.js
PickupModel.js
trackingModel.js
UserModel.js
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
OUTLINE
TIMELINE
trackingModel.js
backend > models > trackingModel.js > trackingSchema > currentLocation
// @TrackingModel.js
const mongoose = require("mongoose");
const trackingSchema = new mongoose.Schema({
trackingId: {
type: String,
required: true,
unique: true, // e.g., "EW12345678"
trim: true,
},
status: {
type: String,
enum: ["Collected", "Transit", "Delivered"],
default: "Collected",
},
eta: {
type: String, // easy to format (e.g., "May 8, 2025, 3:00 PM")
},
destination: {
type: String,
trim: true,
},
collectedAt: {
type: String,
},
currentLocation: {
type: String,
trim: true,
},
deliveredAt: {
type: String,
default: null,
backend main* 0 0
vishi saxena (4 weeks ago) Ln 27, Col 18 Spaces: 2 UTF-8 CRLF {} JavaScript Go Live Prettier


File Edit Selection View Go Run Terminal Help
e-waste_link_project
EXPLORER
E-WASTE_LINK_PROJECT
backend
config
controllers
blogController.js
categoryController.js
contactController.js
pickupController.js
trackingController.js
userController.js
middlewares
AuthMiddleware.js
models
BlogModel.js
CategoryModel.js
contactModel.js
PickupModel.js
trackingModel.js
UserModel.js
node_modules
routes
.env
.gitignore
index.js
package-lock.json
package.json
frontend
app.html
blog.html
category.html
index.html
OUTLINE
TIMELINE
UserModel.js
backend > models > UserModel.js > userSchema
const mongoose= require('mongoose');
// const bcrypt =require('bryptjs');
const userSchema=new mongoose.Schema({
fullname:{
type:String,
required:true,
trim:true
},
email:
{
type:String,
required:true,
unique:true,
lowercase:true,
},
phone:{
type:String,
required:true,
},
userType:{
type:String,
enum:['household','collector'],
required:true,
},
password:{
type:String,
required:true,
},
createdAt:{
backend main* 0 0
vishi saxena (4 weeks ago) Ln 21, Col 7 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live Prettier

DATABASE TABLES

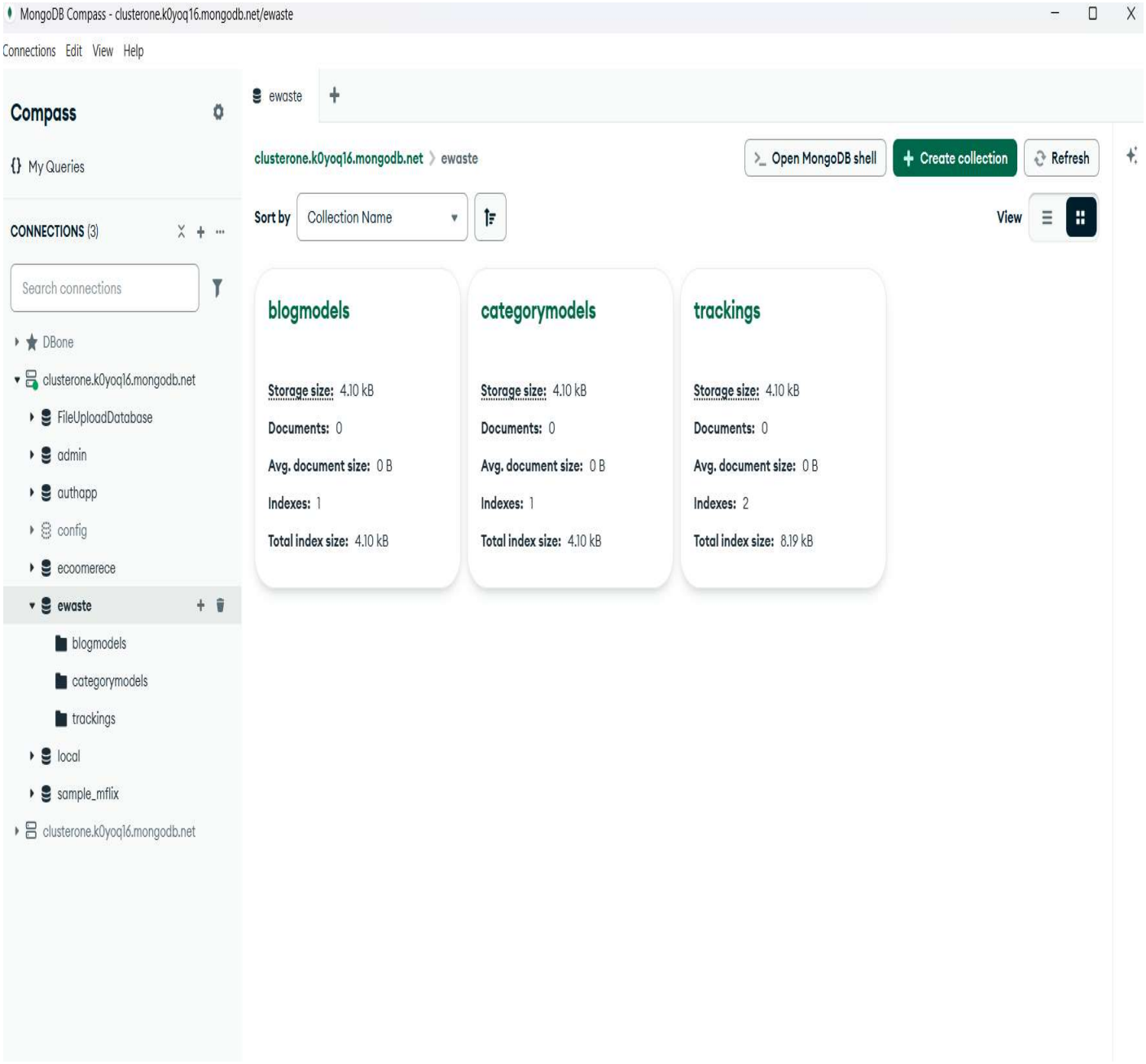
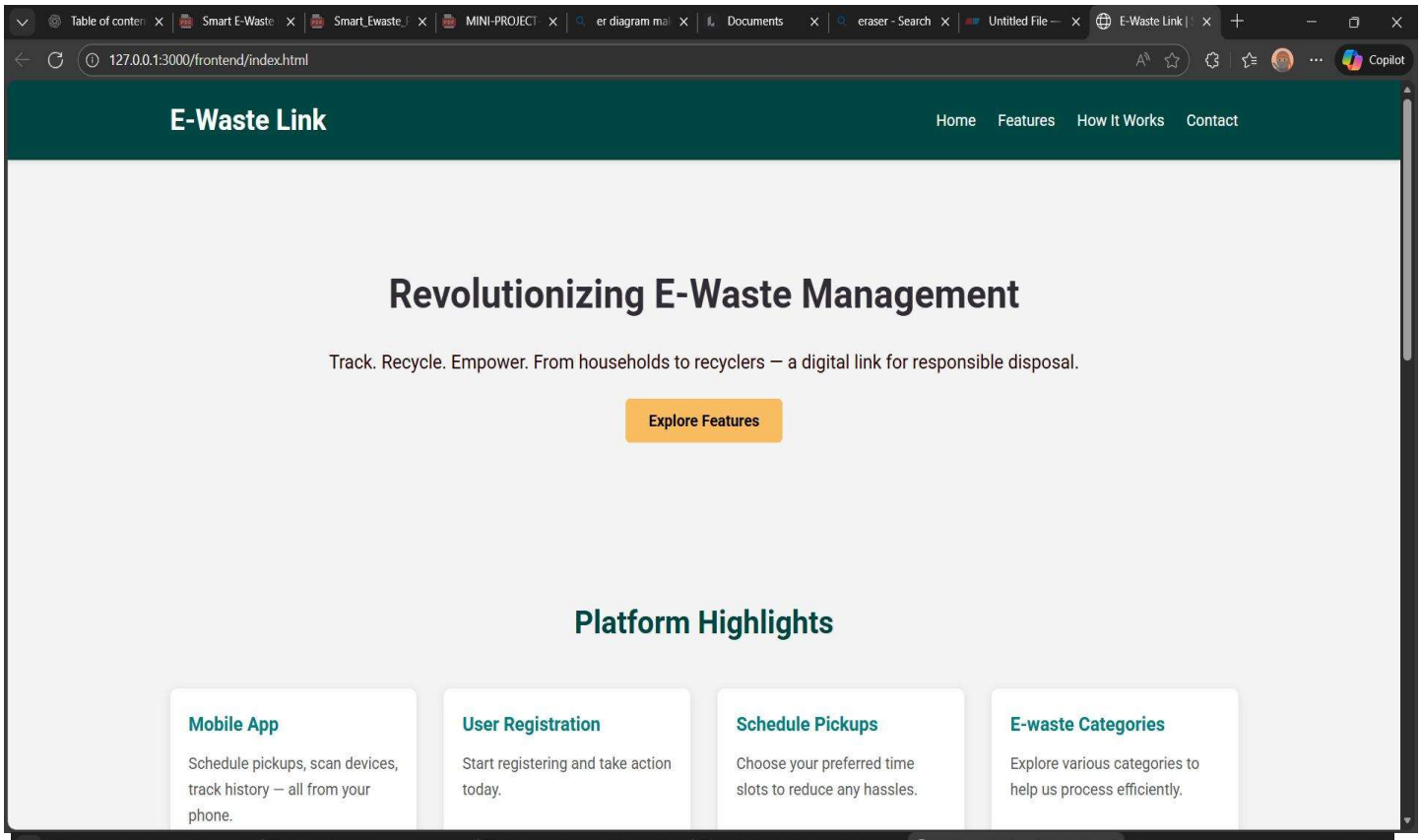
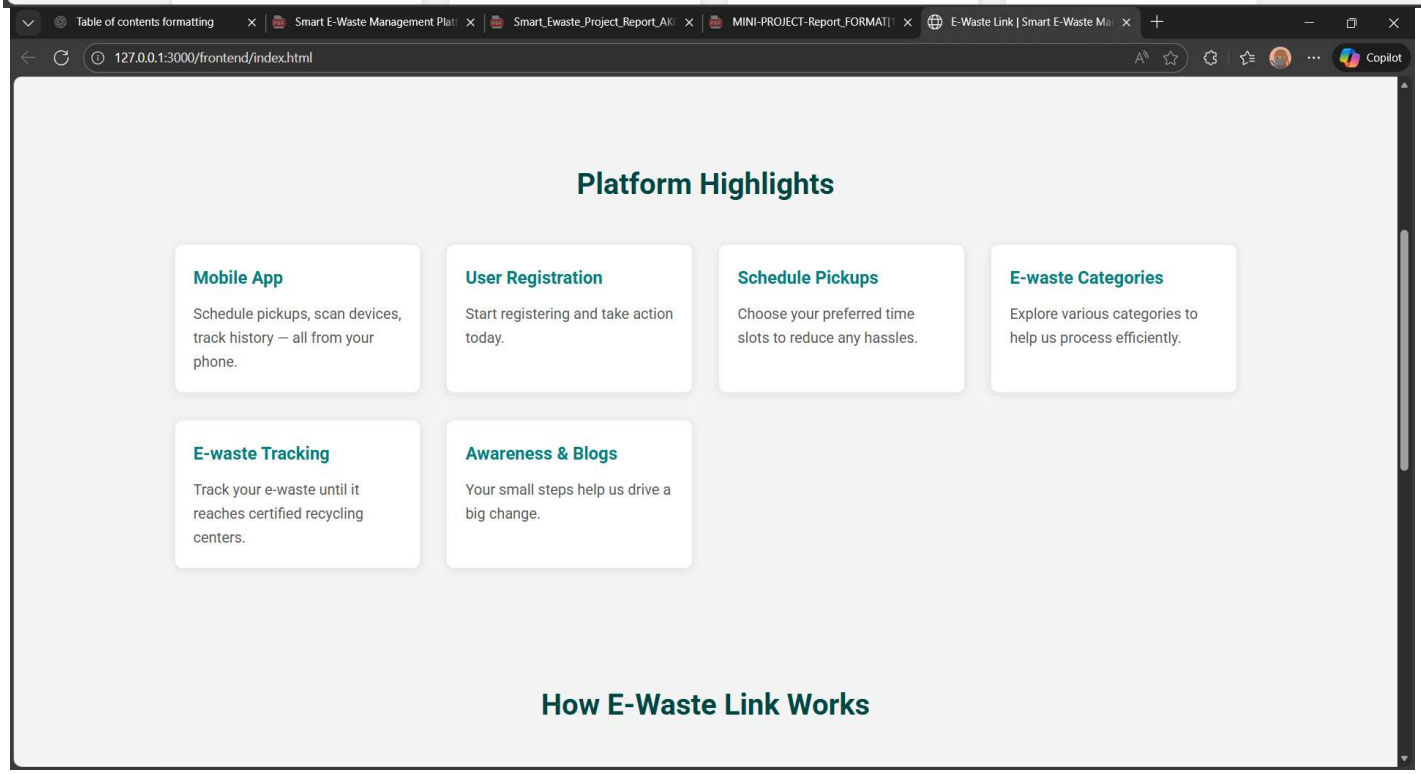
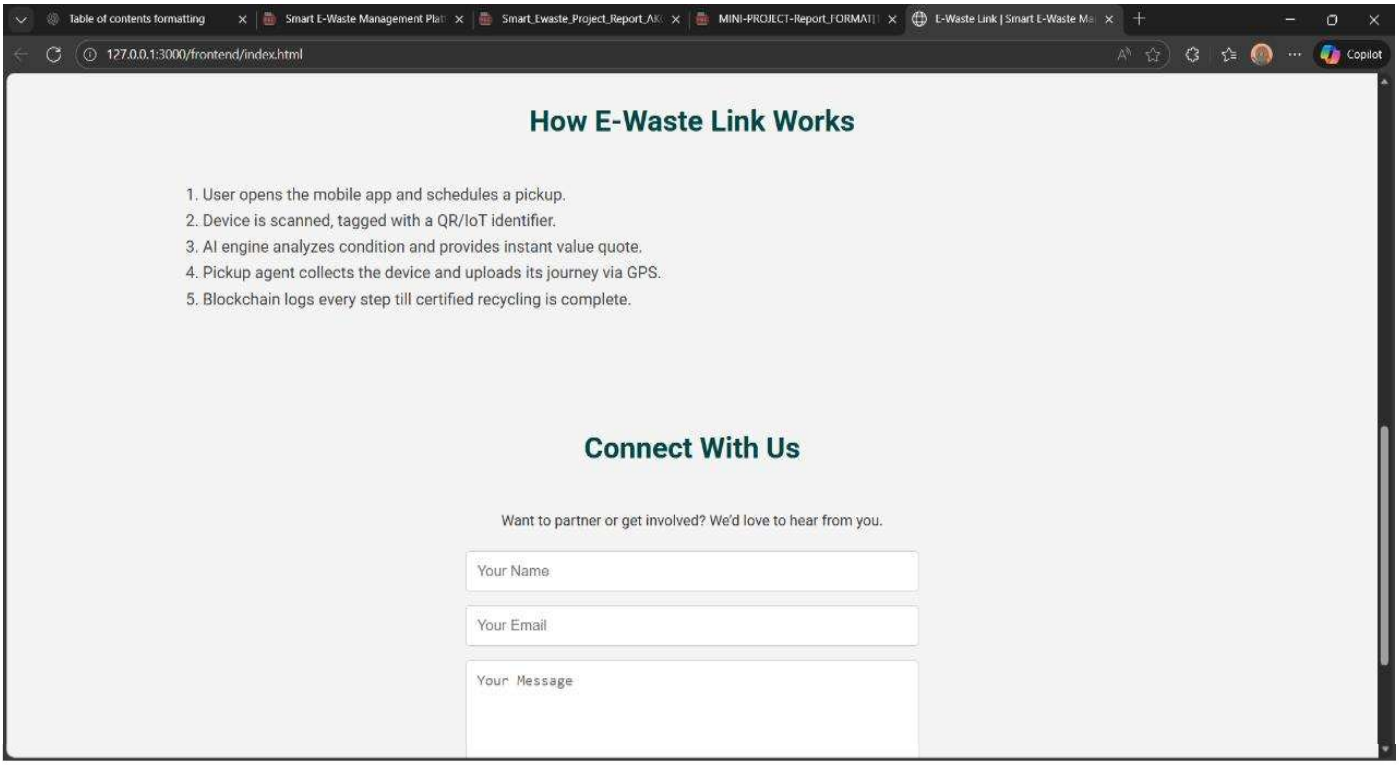


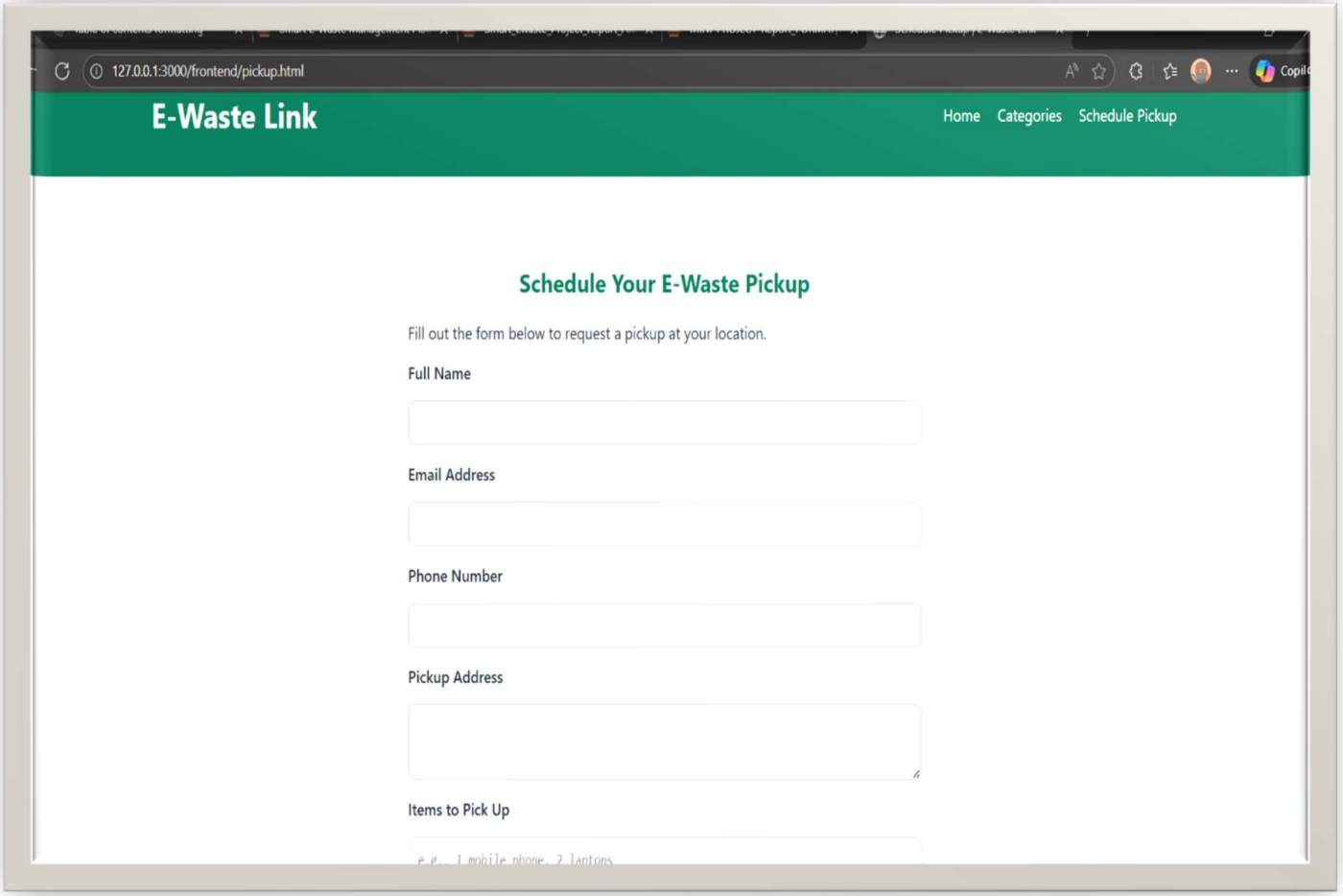
fig -Database tables;

SNAPSHOTS


127.0.0.1:3000/frontend/index.html
E-Waste Link
Home
Features
How It Works
Contact
Revolutionizing E-Waste Management
Track. Recycle. Empower. From households to recyclers — a digital link for responsible disposal.
Explore Features
Platform Highlights
Mobile App
Schedule pickups, scan devices, track history — all from your phone.
User Registration
Start registering and take action today.
Schedule Pickups
Choose your preferred time slots to reduce any hassles.
E-waste Categories
Explore various categories to help us process efficiently.


127.0.0.1:3000/frontend/index.html
Platform Highlights
Mobile App
Schedule pickups, scan devices, track history — all from your phone.
User Registration
Start registering and take action today.
Schedule Pickups
Choose your preferred time slots to reduce any hassles.
E-waste Categories
Explore various categories to help us process efficiently.
E-waste Tracking
Track your e-waste until it reaches certified recycling centers.
Awareness & Blogs
Your small steps help us drive a big change.
How E-Waste Link Works


127.0.0.1:3000/frontend/index.html
How E-Waste Link Works
1. User opens the mobile app and schedules a pickup.
2. Device is scanned, tagged with a QR/IoT identifier.
3. AI engine analyzes condition and provides instant value quote.
4. Pickup agent collects the device and uploads its journey via GPS.
5. Blockchain logs every step till certified recycling is complete.
Connect With Us
Want to partner or get involved? We'd love to hear from you.
Your Name
Your Email
Your Message


127.0.0.1:3000/frontend/pickup.html
E-Waste Link
Home
Categories
Schedule Pickup
Schedule Your E-Waste Pickup
Fill out the form below to request a pickup at your location.
Full Name
Email Address
Phone Number
Pickup Address
Items to Pick Up

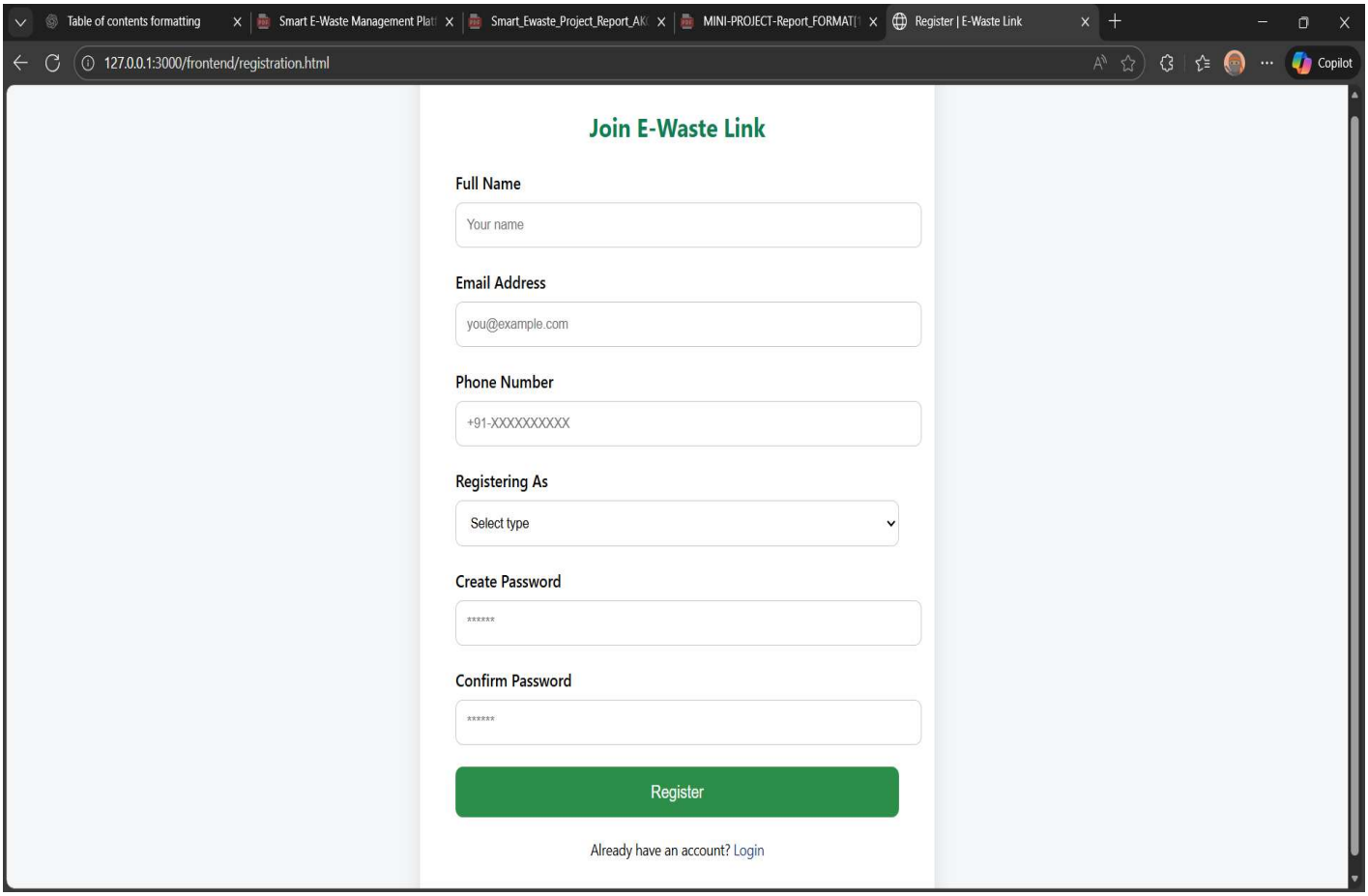


Table of contents formatting
Smart E-Waste Management Plat
Smart_Ewaste_Project_Report_AK
MINI-PROJECT-Report_FORMAT
Register | E-Waste Link
127.0.0.1:3000/frontend/registration.html
Copilot
Join E-Waste Link
Full Name
Your name
Email Address
you@example.com
Phone Number
+91-XXXXXXXXXX
Registering As
Select type
Create Password
Confirm Password
Register
Already have an account? Login

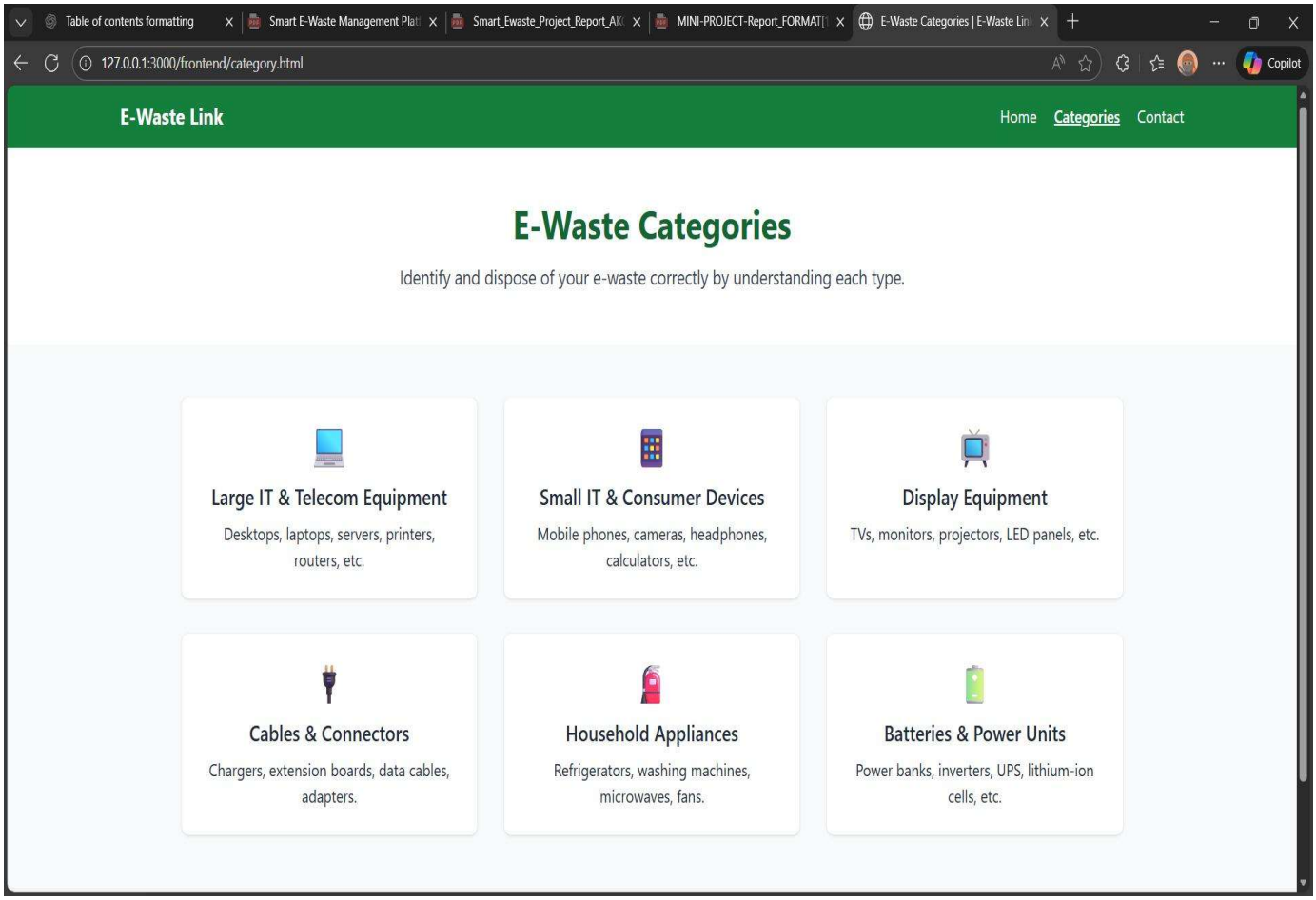


Table of contents formatting
Smart E-Waste Management Plat
Smart_Ewaste_Project_Report_AK
MINI-PROJECT-Report_FORMAT
E-Waste Categories | E-Waste Link
127.0.0.1:3000/frontend/category.html
Copilot
E-Waste Link
Home
Categories
Contact
E-Waste Categories
Identify and dispose of your e-waste correctly by understanding each type.
Large IT & Telecom Equipment
Desktops, laptops, servers, printers, routers, etc.
Small IT & Consumer Devices
Mobile phones, cameras, headphones, calculators, etc.
Display Equipment
TVs, monitors, projectors, LED panels, etc.
Cables & Connectors
Chargers, extension boards, data cables, adapters.
Household Appliances
Refrigerators, washing machines, microwaves, fans.
Batteries & Power Units
Power banks, inverters, UPS, lithium-ion cells, etc.

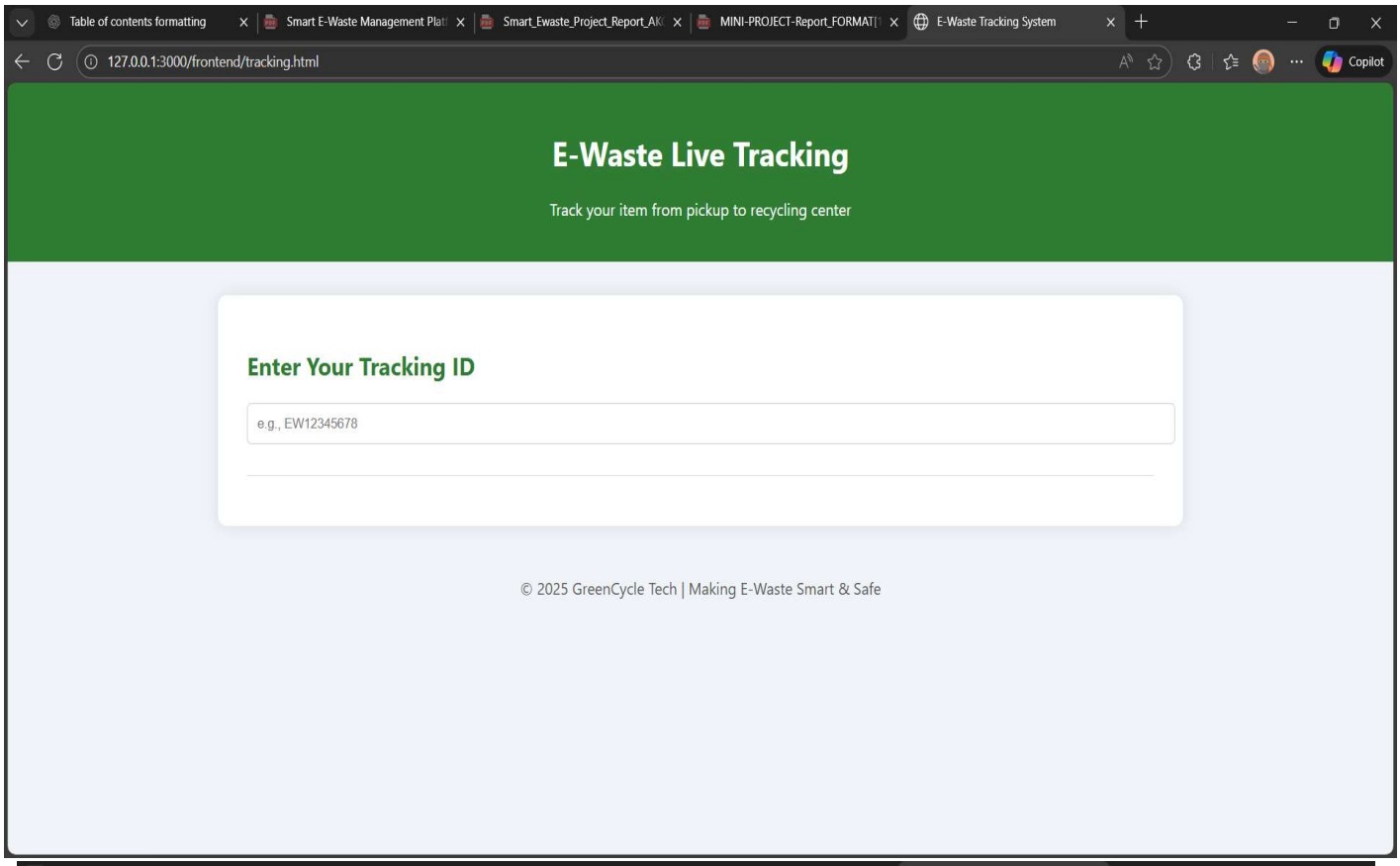


Table of contents formatting
Smart E-Waste Management Plat
Smart_Ewaste_Project_Report_AK
MINI-PROJECT-Report_FORMAT
E-Waste Tracking System
127.0.0.1:3000/frontend/tracking.html
Copilot
E-Waste Live Tracking
Track your item from pickup to recycling center
Enter Your Tracking ID
e.g., EW12345678
© 2025 GreenCycle Tech | Making E-Waste Smart & Safe

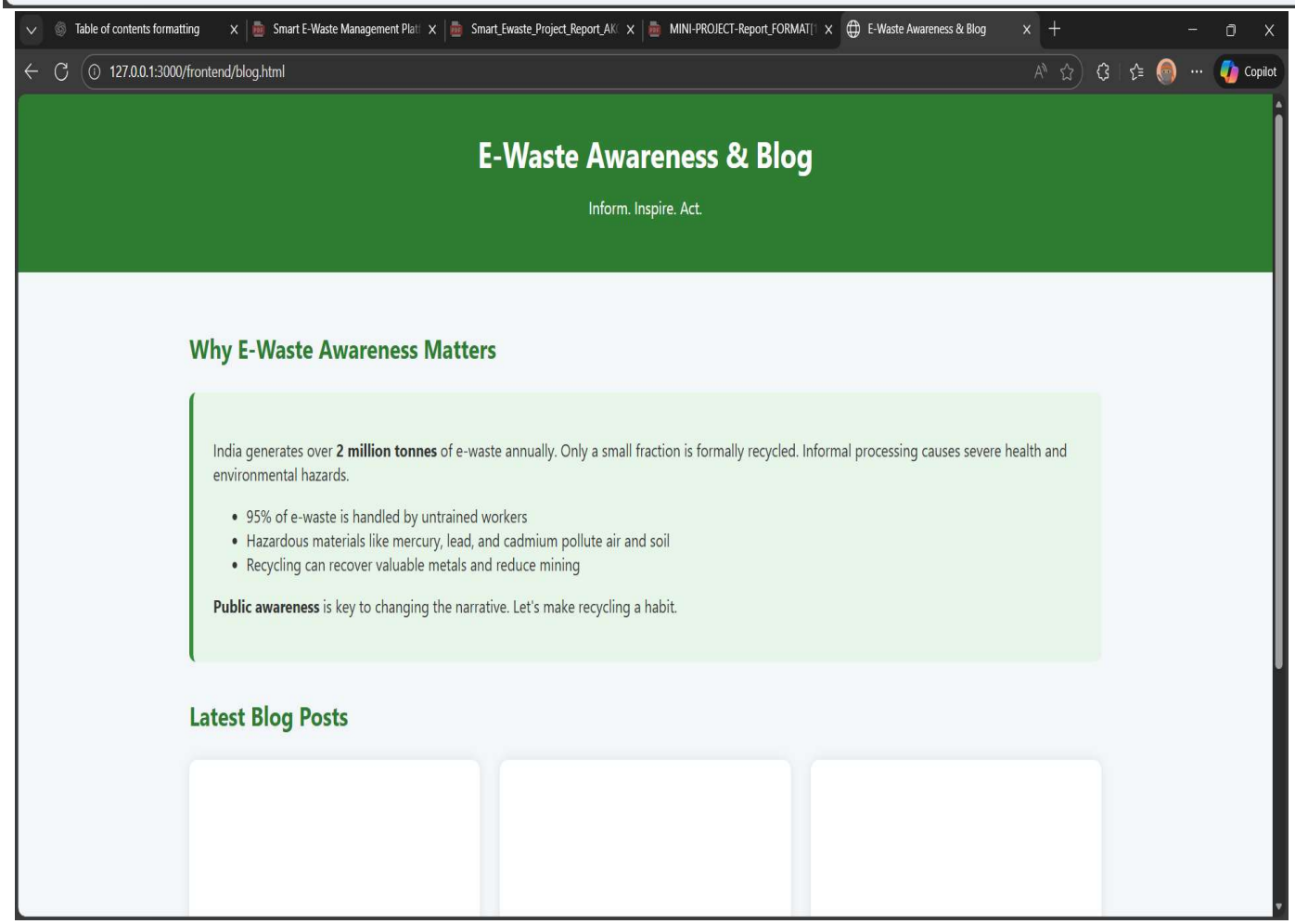


Table of contents formatting
Smart E-Waste Management Plat
Smart_Ewaste_Project_Report_AK
MINI-PROJECT-Report_FORMAT
E-Waste Awareness & Blog
127.0.0.1:3000/frontend/blog.html
Copilot
E-Waste Awareness & Blog
Inform. Inspire. Act.
Why E-Waste Awareness Matters
India generates over 2 million tonnes of e-waste annually. Only a small fraction is formally recycled. Informal processing causes severe health and environmental hazards.
95% of e-waste is handled by untrained workers
Hazardous materials like mercury, lead, and cadmium pollute air and soil
Recycling can recover valuable metals and reduce mining
Public awareness is key to changing the narrative. Let's make recycling a habit.
Latest Blog Posts

CONCLUSION

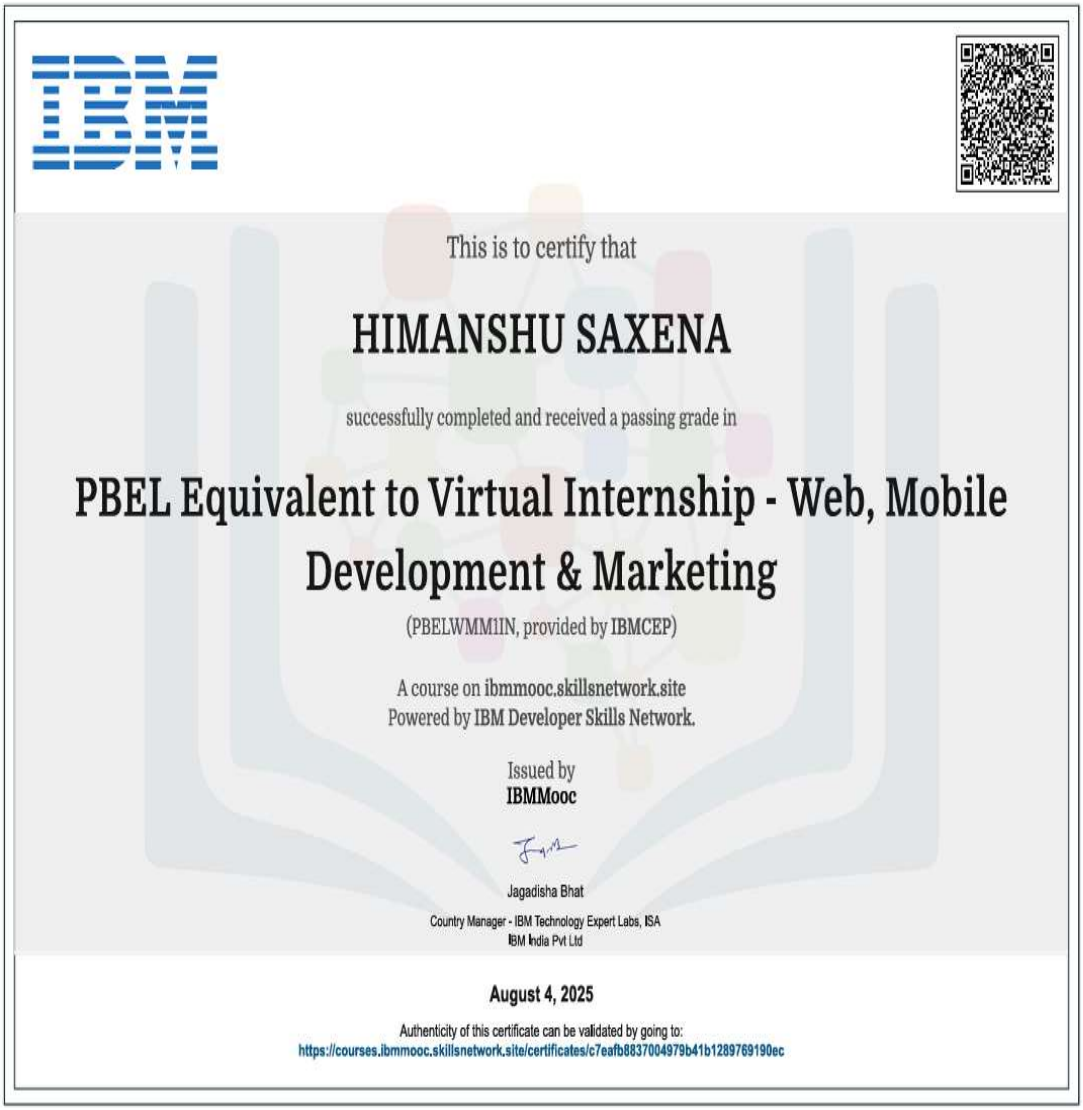
This report has outlined the design of a comprehensive Smart E-Waste Management Platform leveraging AI, IoT, and Blockchain to tackle the e-waste problem. By integrating real-time IoT sensing with intelligent analytics, the system provides fine-grained monitoring of waste levels and automatic classification of disposed items. Blockchain-backed record-keeping adds a tamper-proof audit trail, ensuring accountability and building trust among stakeholders. Our literature survey and requirement analysis indicate that such a system can significantly enhance recycling efficiency: for example, AI-driven route scheduling can reduce collection costs by ~30%, and transparent data sharing can raise compliance rates.

Given the UN's prediction that e-waste will reach *82 million tonnes by 2030*, solutions like this platform are urgently needed. If widely adopted, even a modest increase in recycling (e.g. from 22% to 60%) could yield benefits exceeding **US$38 billion** and greatly reduce environmental harm.The proposed platform aims to contribute toward these goals by making e-waste disposal convenient and reliable for citizens, and by streamlining operations for municipalities and recyclers.

In conclusion, our Smart E-Waste Management Platform offers a prototype for future sustainable cities. It embodies circular economy principles: recovering valuable resources and minimizing waste. Future work would involve implementing the prototype, conducting pilot deployments, and gathering empirical data. Additional enhancements might include machine learning improvements, more robust blockchain privacy schemes, or integration with other city IoT infrastructure. Nonetheless, the core architecture presented here—IoT sensing, AI classification, and blockchain transparency—lays the foundation for smarter, safer, and more efficient e-waste handling, aligning with both technological trends and environmental imperatives.

Sources: Authoritative studies and reports were referenced throughout (e.g., Global E-waste Monitor, Scientific Reports) to inform statistics and technical decisions.The system design builds on proven concepts from recent research in smart waste management.

Each chapter above reflects current best practices and empirical findings in the field, ensuring the platform design is both cutting-edge and grounded in reality.



This is to certify that

HIMANSHU SAXENA

successfully completed and received a passing grade in

PBEL Equivalent to Virtual Internship - Web, Mobile Development & Marketing

(PBELWMM1IN, provided by IBMCEP)

A course on ibmmooc.skillsnetwork.site
Powered by IBM Developer Skills Network.

Issued by
IBMMooc

Jagadisha Bhat
Country Manager - IBM Technology Expert Labs, ISA
IBM India Pvt Ltd

August 4, 2025

Authenticity of this certificate can be validated by going to:
https://courses.ibmmooc.skillsnetwork.site/certificates/c7eafb8837004979b41b1289769190ec

ZIDIO
DEVELOPMENT

CERTIFICATE

OF ACHIEVEMENT

THIS CERTIFICATE IS PRESENTED TO

Harshit Kumar Singh

In recognition of his/her efforts and achievernent in completing 30 days internship program in Web Development.

Conducted From 18th July – 18th August, 2025

Qadism

Managing Director