

FIX YOUR GADGETS

A Project Report

Submitted by

SYAM KUMAR T S

TVE23MCA-2055

To

the APJ Abdul Kalam Technological University

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

Master of Computer Applications



Department of Computer Applications

College of Engineering

Trivandrum-695016

NOVEMBER 2024

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING

TRIVANDRUM



CERTIFICATE

This is to certify that the report entitled **FIX YOUR GADGETS** submitted by **SYAM KUMAR T S** to the APJ Abdul Kalam Technological University in Partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Guide
Name: Prof. Sunitha D
Sign:

Head of the Dept
Name: Prof Liji P.I
Sign:

DECLARATION

Me **Syam Kumar T S**, do hereby declare that the project titled "**Fix Your Gadgets**" is a record of work carried out under the guidance **Prof.Sunitha D**, Asst. Professor, Department of Computer Applications, College Of Engineering, Trivandrum as per the requirement of the curriculum of Master of Computer Applications Programme of A P J Abdul Kalam Technological University, Thiruvananthapuram. Further, I also declare that this report has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Trivandrum

Syam Kumar T S

Date:

(TVE23MCA-2055)

Acknowledgement

First and foremost I thank GOD almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to Dr. Savier J S, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to Prof Liji PI, HOD, Dept of Computer Applications, for providing me with the best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to Prof Sunitha D, Asso. Professor, Department of Computer Applications, College of Engineering Trivandrum for his valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank the other Asst. Professors in the department and all other staff of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

SYAM KUMAR T S

SYNOPSIS

"FIX YOUR GADGETS" is an innovative, Python-based web application developed to address the increasing demand for convenient and accessible gadget repair services. In today's world, where we rely heavily on electronic devices like smartphones, laptops, and smartwatches, the need for prompt and efficient repair solutions is essential. This platform allows customers to book doorstep repair services for various electronic devices and also offers e-waste collection services, contributing to environmentally responsible disposal. Designed primarily for the FYG Store, the application centralizes all repair and e-waste management processes, with the store owner serving as the system's admin, who oversees all activities.

The platform supports four distinct user roles: Admin, Technician, E-waste Collector, and Customer, each playing a vital role in the system's functionality. The Admin manages service requests, registers technicians and collectors, and allocates tasks. Technicians can accept or reject assigned repairs, while e-waste collectors handle collection requests and record disposal locations. This user-based structure ensures efficient workflow and transparency in both repair and recycling services. By providing easy-to-access repair options and promoting sustainable e-waste management, FIX YOUR GADGETS makes gadget upkeep more manageable and eco-friendly.

TABLE OF CONTENTS

1. Introduction	1
1.1 About the project.....	2
1.2 Objective of the project.....	2
1.3 Scope of the project.....	3
2. Initial investigation and Feasibility study	4
2.1 Initial Investigation.....	5
2.2 Feasibility study.....	6
2.2.1 Economic Feasibility.....	7
2.2.2 Technical feasibility.....	7
2.2.3 Behavioral feasibility.....	7
2.2.4 Operational feasibility.....	8
3. System Analysis	9
3.1 System Analysis.....	10
3.2 Existing system.....	10
3.3 Proposed System.....	11
3.4 Software Requirement Specification.....	11
4. Modules	16
4.1 Modules.....	17
4.1.1 Admin.....	17
4.1.2 Technician.....	18
4.1.3 Ewaste Collector.....	18
4.1.4 Customer.....	19
5. Flow Diagrams	20
5.1 Data Flow Diagram.....	21
6. System Design	26
6.1 System Design.....	27
6.2 Input Design.....	27
6.3 Output Design.....	28
6.4 Table Design.....	28

7. Tools and Platforms	35
7.1 Front end	36
7.2 Back end	36
7.3 Framework	36
7.4 Platform	36
7.5 Database	37
8. Testing	38
8.1 Unit Testing	39
8.2 Integration Testing	39
8.3 System Testing	39
8.3.1 Acceptance Testing	40
8.3.2 Validation Testing	40
8.3.3 Black Box Testing	40
8.3.4 White Box Testing	40
8.4 Sample Test Cases	41
9. System Implementation	42
9.1 Implementation	43
9.2 Problem Statement	43
9.3 Problem Definition	44
10. Conclusion	45
11. Appendices	47
A. Sample Screens	48
B. Sample Code	55
C. Hardware and Software Requirements	61
D. Bibliography	61

INTRODUCTION

1. INTRODUCTION

1.1 About The Project

"FIX YOUR GADGETS" is an innovative, Python-based web application developed to address the increasing demand for convenient and accessible gadget repair services. In today's world, where we rely heavily on electronic devices like smartphones, laptops, and smartwatches, the need for prompt and efficient repair solutions is essential. This platform allows customers to book doorstep repair services for various electronic devices and also offers e-waste collection services, contributing to environmentally responsible disposal. Designed primarily for the FYG Store, the application centralizes all repair and e-waste management processes, with the store owner serving as the system's admin, who oversees all activities.

The platform supports four distinct user roles: Admin, Technician, E-waste Collector, and Customer, each playing a vital role in the system's functionality. The Admin manages service requests, registers technicians and collectors, and allocates tasks. Technicians can accept or reject assigned repairs, while e-waste collectors handle collection requests and record disposal locations. This user-based structure ensures efficient workflow and transparency in both repair and recycling services. By providing easy-to-access repair options and promoting sustainable e-waste management, Fix Your Gadgets makes gadget upkeep more manageable and eco-friendly.

1.1.1 Objective of The Project

The primary objective of this project is to create an all-encompassing, web-based platform designed to provide seamless repair services for electronic gadgets, paired with responsible e-waste management options. With the increasing reliance on electronic devices in daily life, this platform aims to meet the growing need for accessible and efficient repair solutions by allowing customers to conveniently book doorstep repair services. This feature not only improves the user experience but also encourages the maintenance and longevity of devices, reducing the frequency of replacements. The platform also includes a dedicated e-waste collection function, facilitating sustainable disposal practices to help reduce the environmental impact of electronic waste.

1.1.2 Scope of The Project

The scope of this project includes the creation of a comprehensive online service management system, designed to handle various user roles such as administrators, technicians, e-waste collectors, and customers. This platform allows for streamlined booking, assignment, and tracking of repair services, along with systematic e-waste collection and disposal tracking. Its functionality spans from user registration and task allocation to service management and eco-friendly waste processing, making it a versatile tool for businesses focused on offering convenient repair services and supporting environmental sustainability.

Data entry screens are designed such that they are very user friendly and minimum typing is required from the user,

- Not much training required.
- The new web application is more user friendly.
- It aims on paperless work.
- Fast access information.
- Efficient traceability.
- Duplication of data will be avoided.
- Menu driven interface provides ease to use.
- Availability of previous data for future reference.

INITIAL INVESTIGATION AND FEASIBILITY STUDY

2.1 INITIAL INVESTIGATION

The initial investigation for this project highlights the growing demand for accessible, reliable gadget repair services and responsible e-waste disposal. With electronic devices becoming indispensable in everyday life, users often face challenges when their gadgets malfunction. Traditional repair services may be inconvenient, while doorstep repair offers added ease and customer satisfaction. Additionally, the rise in e-waste poses environmental concerns, creating a need for responsible collection and disposal solutions. This platform aims to address both issues by providing users with a streamlined process to book repair services and request e-waste pickups from their homes.

A market analysis reveals that while many repair services focus on in-store solutions, few combine direct booking and e-waste collection in a single platform. By integrating both services, this project fills a significant gap, catering to users who value convenience and environmental responsibility. It aligns with global trends in sustainability, making it appealing to tech-savvy and eco-conscious users. This unique approach also offers businesses a chance to expand their service offerings, making the platform valuable to both customers and repair stores alike.

The project's technical requirements involve creating a robust system with four distinct user roles: Admin, Technician, E-waste Collector, and Customer. Key functionalities include booking management, real-time updates, and secure user profiles. Python, chosen for its flexibility and support for web development, will ensure the platform is scalable, responsive, and easy to maintain. With attention to usability, scalability, and data security, the project aims to deliver a reliable platform that addresses customer needs, supports e-waste management, and promotes sustainable practices.

2.2 FEASIBILITY STUDY

System feasibility is a test or evaluation of the complete system plan. Such an evaluation is necessary to define the application area along with the extend and capability to provide the scope of computerization together with suggested output and input format and potential benefits. Feasibility study is a proposal according to the work ability, impact on the organization, ability to meet user's needs and efficient use of resources. The feasibility study is conducted to determine if the proposed system is feasible or not. Feasibility analysis evaluates the candidate systems and determines the best system that needs performance requirements. The purpose of feasibility study is to investigate the present system, evaluate the possible application of computer-based methods, select a tentative system, evaluate the cost and effectiveness of the proposed system, evaluate impact of the proposed system on existing personnel and ascertain the need for new personnel. All projects are feasible when given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. A feasibility study is not warranted for systems in which economic justification is obvious, technical risk is low, few legal problems are expected and no reasonable alternative exists. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from the business point of view and if it can be developed in the given existing budgetary constraints. The feasibility study should be relatively cheap and quick. The result should inform the decision of whether to go ahead with a more detailed analysis. Feasibility study may be documented as a separated report to higher officials of the top-level management and can be included as an appendix to the system specification. Feasibility and risk analysis are related in many ways. If there is more project risk then the feasibility of producing the quality software is reduced.

The key combinations are involved in the feasibility study:

- Economic Feasibility.
- Technical Feasibility.
- Behavioral Feasibility.
- Operational Feasibility

2.2.1 Economic Feasibility

The "FIX YOUR GADGETS" application demonstrates strong economic feasibility, with manageable initial investments in software development, database setup, and cloud hosting. Leveraging Python-based open-source tools reduces licensing costs, making the platform cost-effective. Additionally, FYG is positioned to generate a steady revenue stream by attracting a wide customer base that values the convenience of doorstep repair services. Increased service requests, coupled with potential revenue from e-waste collection, enhance the platform's profitability. The automation of administrative tasks, service requests, and tracking mechanisms is likely to yield long-term cost savings by reducing manual errors and administrative overhead, thus providing a positive return on investment.

2.2.2 Technical Feasibility

From a technical standpoint, FYG is highly feasible, leveraging Python's versatility along with frameworks like Django or Flask for a secure and scalable backend infrastructure. Python's vast libraries and robust community support facilitate efficient development and maintenance. The application is designed to be scalable with cloud hosting options, allowing it to expand as service demand grows. Integrations with payment gateways enhance both user experience and operational efficiency. Regular software updates and a secure backup protocol ensure that the platform remains stable, secure, and adaptable to future needs.

2.2.3 Behavioral Feasibility

FYG is expected to achieve high user acceptance, as it aligns with the preferences of modern customers who seek convenience in booking doorstep repair services. The user-friendly interface caters to diverse user roles—customers, technicians, e-waste collectors, and the admin—ensuring smooth onboarding and minimal training requirements. Technicians and e-waste collectors will require only brief tutorials to manage their roles, while the admin, with some training, can effectively oversee the system using an intuitive dashboard. Given the efficient workflows and customer-centric features, user adaptation is anticipated to be smooth, with a positive reception among all stakeholders.

2.2.4 Operational Feasibility

The operational feasibility of FYG is reinforced by its efficient resource management and streamlined workflows. The platform allows the admin to manage all aspects of repair and e-waste service requests, from registration to assignment and task tracking. Designed to scale, FYG can accommodate more technicians and expand service areas as demand grows. Additionally, its integration of e-waste collection aligns with regulatory and environmental standards, making the platform eco-friendly and potentially eligible for green certifications. This operational efficiency ensures that FYG can reliably support both current and future demand.

SYSTEM ANALYSIS

3.1 System Analysis

Analysis is a structured method for identifying and solving problems. Analysis implies breaking something into its parts so that the whole may be understood. The definition of system analysis not only process analysis but also that of synthesis, which implies the process of putting parts together to form a new whole. All the activities relating to the life cycle phase must be performed managed and document. To design a system, we need requirements of the system and the specification document are prepared in this phase. The purpose of this document is to specify the functional requirement of the software that is to build. The specifications are intended to guide the activities, relationships and all other objectives. The main thing is to find what is to be done to solve the problems with the current system. In the phase the problems or drawbacks of the current system is identified and the necessary actions to solve these problems are recommended

3.2 Existing System

The existing system for managing gadget repairs and e-waste disposal is fragmented and often inconvenient for users. Typically, customers in need of repairs must rely on in-store service centers or independent repair providers, which usually require them to either visit a physical location or mail their devices for service. This process can be time-consuming and lacks the flexibility needed in today's busy lifestyle, as customers may face issues with transport, long wait times, and limited access to reliable repair centers. Transparency is often lacking in terms of service pricing and progress tracking, leaving users unsure of the status and quality of the repair. Additionally, the process for e-waste disposal remains separate from repair services. Users who wish to dispose of non-functional or outdated gadgets often must search independently for recycling centers, municipal disposal programs, or specialized e-waste management companies. This disconnection can lead to improper disposal practices, as users may be unaware of or lack convenient access to disposal services, contributing to environmental issues related to e-waste accumulation. In regions where recycling facilities are scarce or inconveniently located, users may resort to storing old gadgets or discarding them improperly, missing out on opportunities for responsible disposal and material recovery. Overall, the existing system lacks a unified solution that could streamline both repair services and e-waste collection, making it difficult for users to manage all their device-related needs

on a single platform. An integrated system that offers easy booking for gadget repairs along with accessible e-waste disposal options would significantly enhance user convenience, encourage eco-friendly practices, and better support customers in maintaining their devices responsibly.

3.3 Proposed System

The proposed system is an integrated web platform designed to streamline gadget repair services and e-waste management into a single, efficient solution. Customers will be able to easily book doorstep repair services for various electronic devices, eliminating the need for in-store visits. The platform will feature user registration and profile management, allowing customers to track their repair requests, view service history, and receive real-time updates on the status of their services, thereby enhancing transparency and trust.

In addition to repair services, the platform will incorporate an e-waste collection feature, enabling users to request pickups for non-functional gadgets directly from their homes. E-waste collectors will manage their collection assignments and log disposal sites, promoting responsible recycling practices. The system will support four user roles—Admin, Technician, E-waste Collector, and Customer—ensuring an efficient workflow and accountability. By leveraging modern web technologies, the proposed system aims to enhance the customer experience and contribute to environmental sustainability through responsible electronic waste disposal.

3.4 Software Requirement Specification(SRS)

This Software Requirements Specification (SRS) document provides a detailed description of the Integrated Repair and E-Waste Management System. It outlines the system's functional and non-functional requirements, including user roles, interfaces, and overall system behavior.

A. Purpose

The purpose of this system is to provide a comprehensive web platform that allows users to book gadget repair services and request e-waste collection from the comfort of their homes. This system aims to enhance user convenience, promote responsible e-waste disposal, and streamline the management of repair services.

B. Scope

The scope of the project includes the development of an online platform with the following features:

- User registration and profile management
- Booking of repair services for various electronic gadgets
- E-waste collection request management
- Role-based access for Admin, Technician, E-waste Collector, and Customer
- Real-time status updates for repair requests and e-waste collections
- Administrative management of users and service requests

C. Overall Description



Product Perspective

The "TECHREVIVE & E-WASTE" web application is a new, independent system designed to streamline the management of electronic repair services and e-waste collection. It is a web-based platform that leverages modern web development technologies such as HTML, CSS, Bootstrap, Python, Django, jQuery, Ajax, and Javascript. The backend uses SQLite for database management.



Product Functions

- **Admin Functions:** User registration, service allocation, system management, report generation, and user feedback management.
- **Technician Functions:** Service request management, status updates, and billing.
- **E-waste Collector Functions:** E-waste request management and yard selection.
- **Customer Functions:** Service booking, e-waste collection requests, complaint submission, feedback provision, and payment processing.



User Characteristics

- **Admin:** Experienced in managing web applications, familiar with store operations.
- **Technician:** Skilled in repairing electronic gadgets, capable of managing online service requests.

- **E-waste Collector:** Knowledgeable about e-waste management, capable of handling collection logistics.
- **Customer:** General users with basic knowledge of online booking systems.



Constraints

- The system must be compatible with Windows 11.
- The application should ensure data security and privacy for all users.
- The system must handle multiple user requests simultaneously without performance



Assumptions and Dependencies

- The system assumes a stable internet connection for optimal performance.
- The application is dependent on the availability of technicians and e-waste collectors for timely service delivery.
- Future enhancements may include integration with third-party payment gateways or expansion to other operating systems.

D. Functional Requirements



User Registration and Authentication

- Users should be able to register with their basic details (name, email, password, contact number).
- Users must be able to log in and log out securely.



Profile Management

- Users can view and update their profile information.
- Users can view their repair request history and e-waste collection requests.



Repair Service Booking

- Customers can browse available repair services and book a service by selecting a date and time for the technician's visit.
- Customers will receive a confirmation of their booking and a unique request ID.



E-Waste Collection Request

- Customers can request e-waste pickups, specifying the type and quantity of items.
- Customers will receive a confirmation of their e-waste collection request.

Role-Based Access Control

- **Admin:** Manage user accounts, assign technicians, oversee e-waste collections, and generate reports.
- **Technician:** View assigned repair requests, accept or reject requests, and update the status of repairs.
- **E-Waste Collector:** View assigned collection requests, log the disposal location, and update the status of collections.
- **Customer:** Book repair services, request e-waste pickups, and view their service history.

**Notifications and Updates**

- Users should receive notifications via email or within the application regarding the status of their repair requests and e-waste collections.

E. Non-Functional Requirements**Performance**

- The system should handle up to 500 concurrent users without performance degradation.
- Response time for user actions should not exceed 2 seconds under normal load.

**Usability**

- The interface should be intuitive and user-friendly, allowing users to navigate easily through the platform.
- The system should provide help and support resources for users.

**Security**

- User data must be securely stored and encrypted.
- The system should implement role-based access controls to protect sensitive information.

**Scalability**

- The system should be designed to accommodate future growth, allowing for the addition of more services, users, and features without major redesign.

F. User Interface Requirements

- The application should have a responsive design suitable for both desktop and mobile devices.
- Clear and consistent navigation should be maintained across all pages.

System Design

The system architecture follows a client-server model, where the client interacts with the application through a web browser, and the server processes requests using Python and Django. The database layer is managed by SQLite.

- **Front-end:** HTML, CSS, Bootstrap, JavaScript, jQuery, Ajax.
- **Back-end:** Python, Django.
- **Database:** SQLite.



Data Flow

1. **User Interaction:** Users interact with the web application via a browser.
2. **Request Handling:** The Django server processes the requests, interacting with the database.
3. **Response Generation:** The server sends back the processed data to the client for display.

G. Conclusion

This SRS document outlines the key requirements for the Integrated Repair and E-Waste Management System. The successful implementation of these requirements will ensure that the system meets user needs, provides reliable services, and promotes environmentally responsible practices.

MODULES

4.1 MODULES

In a software project, a module is a collection of source files and build settings that divides a project into distinct units of functionality. Modules can be used to: Make software easier to use, Define program boundaries, Implement separate modules for different areas of a program, and Partition the system design or code.

Modules can be designed to be reusable and can be called by the program as needed. They can also be imported from others, which may be called a library

4.1.1 Admin Module

Administrator is the person who manages the software. He/she is the person who focuses on the data and reports of the software. He/she is the person who add brands, categories, districts, places, products, yards. He/she also performs the allocation of service bookings and e-waste collection requests.

- a) The admin has supreme power over the system.
- b) The admin enter valid email id and password; log on to the admin home page.
- c) The admin shall able to view all the registered users.
- d) The admin can register technicians and e-waste collectors.
- e) The admin should have the permission to view the details of service booking and provision to allocate service to technicians.
- f) The admin is responsible for maintaining and updating the whole system.
- g) Admin should have the provision to view and allocate e waste collection requests.
- h) Admin should have the provision to add brands,category,yard,district,place,product.
- i) Admin should have the provision to view reports on working of the system.
- j) Admin should have the provision to view feedbacks regarding the system.
- k) Admin should have provision to view complaints regarding services and reply to them.
- l) The system should have the provision to logout.

4.1.2 Technician Module

The Technician module enables technicians to view and update their own information. In addition, they can view the work assigned by the administration and can complete the work.

- a) The system should have a provision to login the technician by entering email id and password.
- b) After logging in the technician have the permission to view and edit their profile.
- c) The technician should have the permission to edit the password that is already given by the admin.
- d) The technician should have the permission to view the service bookings allocated to do that are assigned by the admin.
- e) Technician should have the provision to accept and reject the assigned service.
- f) Technician should have the provision to update work status.
- g) Technician should have the provision to generate bill.
- h) System should have the provision to logout.

4.1.3 E-waste Collector

The E-Waste Collector module is designed for collectors to manage e-waste collection requests and ensure responsible disposal. They can also view and update their own information. In addition, they can view the work assigned by the administration and can complete the work of collecting e-waste. Also have a provision to select the yard where the collector dumped the e-waste.

- a) The system should have a provision to login the e-waste collector by entering email id and password.
- b) After logging in the collector have the permission to view their profile details.
- c) The collector should have the permission to edit the password that is already given by the admin.
- d) The collector should have the provision to view the allocated e-waste collection requests.
- e) Collector should have the provision to confirm and reject the collection request allocated to them.
- f) Collector should have the provision to select the yard where the e-waste is dumped

after collecting.

- g) Collector should have the provision to view the confirmed collection requests.
- h) System should have the provision to log out.

4.1.4 Customer

The Customer module allows users to easily access repair services and e-waste collection through a straightforward interface. They are main users of this application. They can book repair services for their electronic appliances at doorstep. Also can send an e-waste collection request for collecting e-wastes. They can post complaints regarding the service done by technician or send feedback about the web application.

- a) The system should have a provision to register a new user by filling a registration form.
- b) The registered customers can log into the system by entering email id and password.
- c) After logging in customer must have a provision to view their profile and edit profile.
- d) Customers should have the permission to change password.
- e) There should a provision for customers in order to book service by submitting a form.
- f) Customer should have a provision to view their service request and its status.
- g) Customer should have the permission to view bill details of the service and a provision for payment.
- h) Customer should have the provision to request e-waste collection.
- i) Customer should have the provision to view their given e-waste collection requests and its status.
- j) Customer should have the provision to give complaints against technicians and view reply.
- k) Customer should have the provision to give feedback regarding the web application.

FLOW DIAGRAMS

5.1 Data Flow Diagram(DFD)

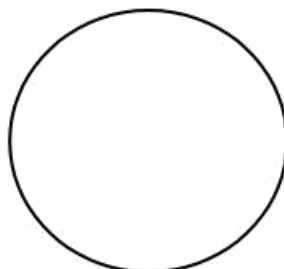
DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

Data Flow Diagram Symbols

External entity: An outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.



Process: Input to output transformation in a system takes place because of process function. The symbols of a process are rectangular with rounded corners, oval, rectangle or a circle. The process is named a short sentence, in one word or a phrase to express its essence.



Data Flow: Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to

the flow to determine the information which is being moved. Data flow also represent material along with information that is being moved. Material shifts are modeled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bi-directional.



Data Store: Also known as warehouse. The data is stored in the warehouse for later use. Two horizontal lines represent the symbol of the store. The warehouse is simply not restricted to being a data file rather it can be anything like a folder with documents, an optical disc, a filing cabinet. The data warehouse can be viewed independent of its implementation. When the data flow from the warehouse it is considered as data reading and when data flows to the warehouse it is called data entry or data updation.



Levels of DFD

DFD uses hierarchy to maintain transparency thus multilevel DFD's can be created. Levels of DFD are as follows:

- 0-levelDFD
- 1-levelDFD
- 2-levelDFD

Rules for creating DFD

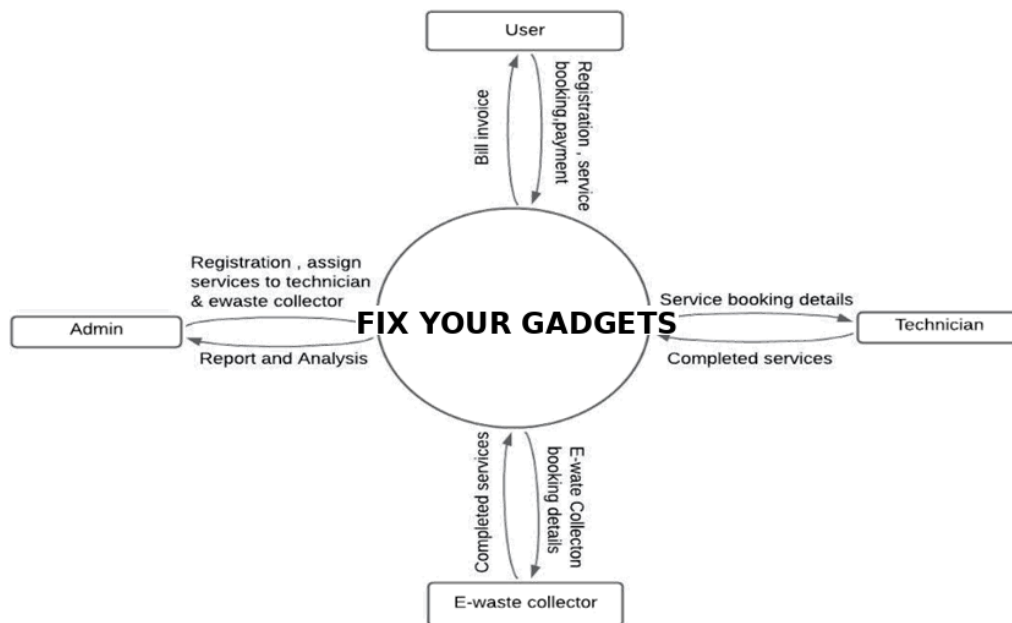
The name of the entity should be easy and understandable without any extra assistance.

The processes should be numbered or put in ordered list to be referred easily.

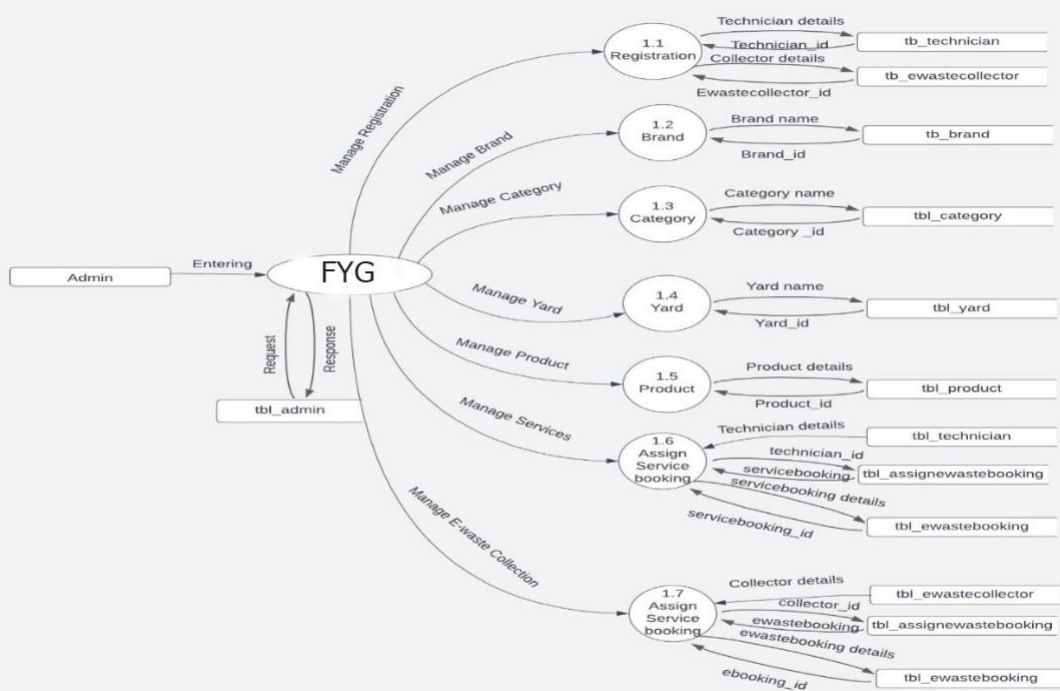
DFD should maintain consistency across all the DFD levels.

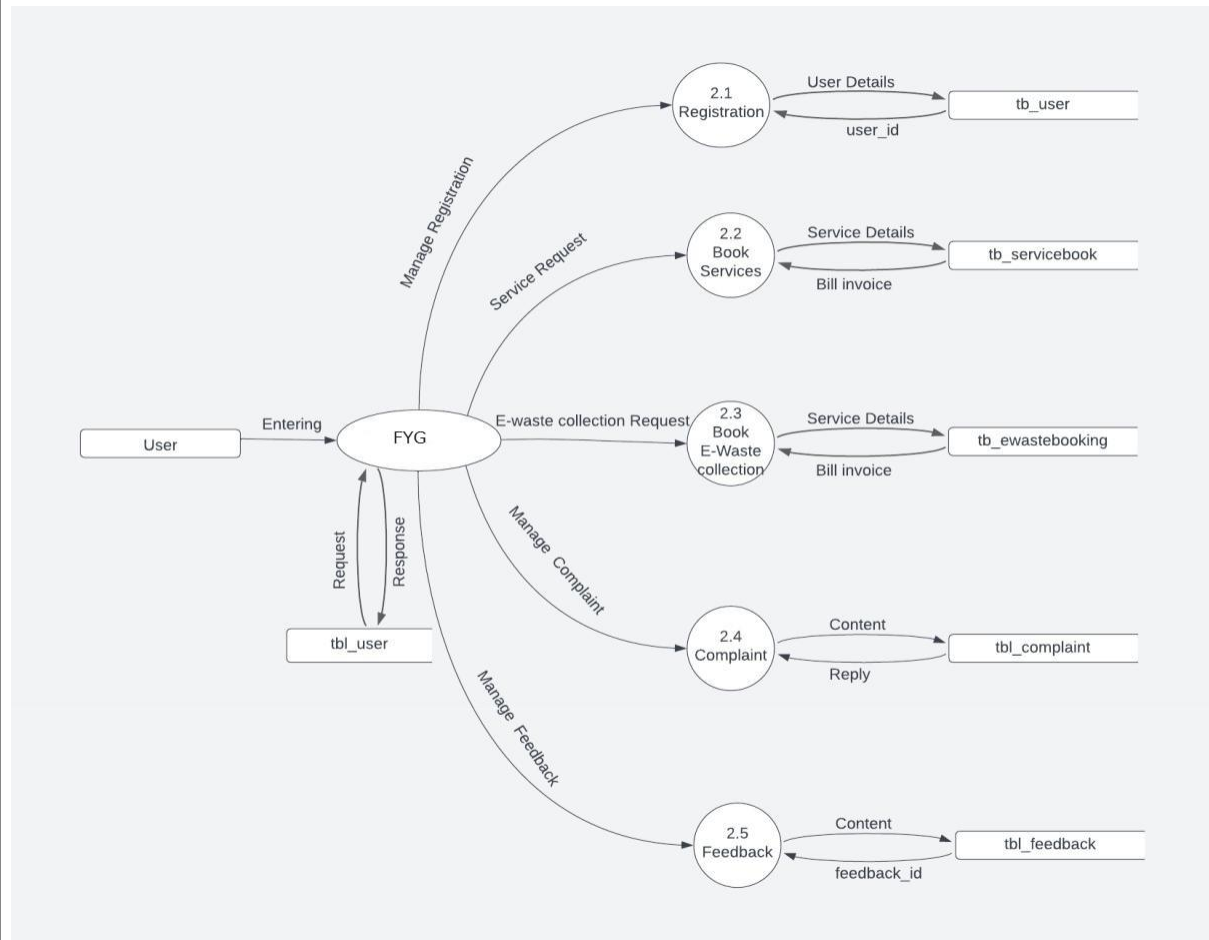
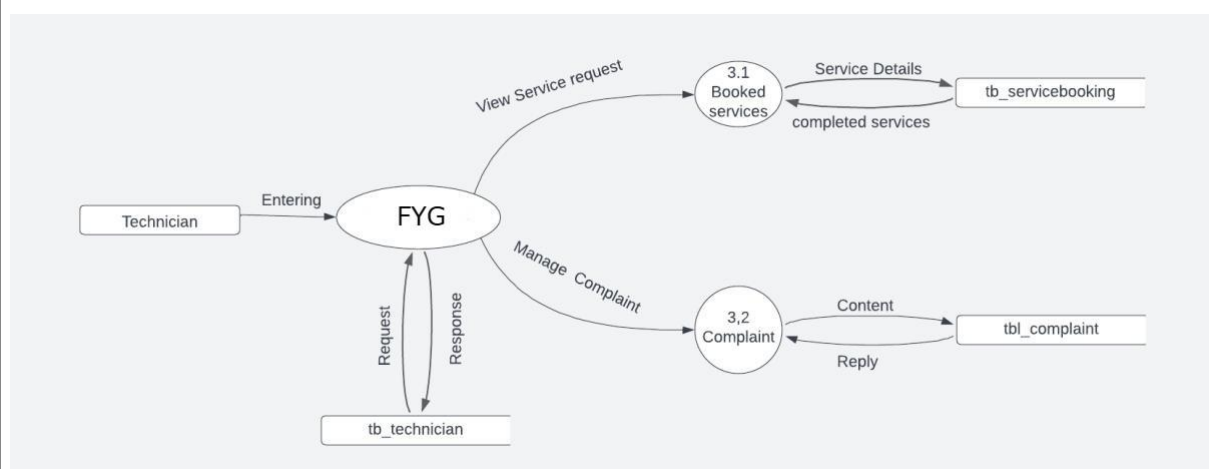
A single DFD can have maximum processes upto 9 and minimum 3 processes.

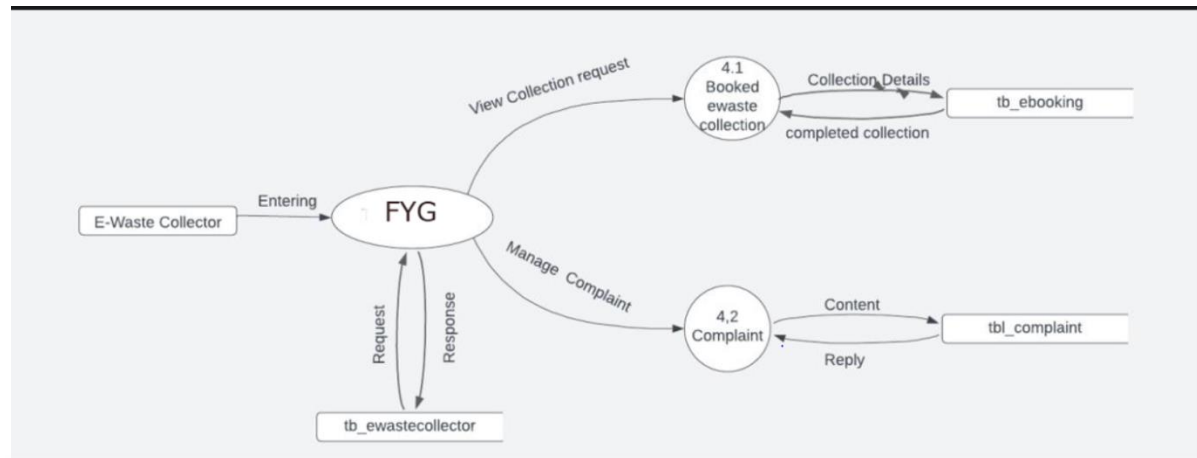
CONTEXT LEVEL-LEVEL 0 DFD



LEVEL 1-ADMIN



LEVEL 1-USER**LEVEL 1-TECHNICIAN**

LEVEL 1-E-WASTE COLLECTOR

SYSTEM DESIGN

6.1 SYSTEM DESIGN

System designing is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. it is a solution to a “how to” approach compared to system analysis which is a “what is” orientation. it translates the system requirements into ways of making them operational. The design phase focuses on the detailed implementation of the system recommended in the feasibility study. The system which is in making is developed by working on two different modules and combining them to work as a single unit. That single unit is the one which is known as the new software. We go through the different design strategies to design the system we are talking about. In the input design we decide which type of input screens are going to be used for the system in making. In the output design we decide the output screens and the reports that will be used to give the output and in the database design we decide what all tables will be required and what all fields will be there in those tables. Each of them discussed briefly below.

6.2 Input Design

Input design converts user-oriented inputs to computer-based formats, which requires careful attention. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via a map support or links. Inaccurate input data is the most common cause of errors in data processing. The input screens need to be designed more carefully and logically. A set of menus is provided which help for better application navigation. While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered. The objective of input design is to create an input layout that is easy to follow and prevent operator errors. It covers all phases of input from creation of initial data into actual entry of the data to the system for processing. The input design is the link that ties the system into world of its users. The user interface design is very important for any application. The interface design defines how the software communication within itself, to system that interpreted with it and with human who use it. The input design requirements such as user friendliness, consistent format and interaction dialogue for giving the right message and help for the user at right time are also considered for the development of the project.

6.3 Output Design

Outputs are the most important and useful information to the user and to the department. Intelligent output designs will improve systems relationships with the user and help much in decision-making. Outputs are also used to provide a permanent hard copy of the results for later use. The forms used in the system are shown in the appendix. The outputs also vary in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be printed are all considered. The output forms a system in the justification for its existence. If the outputs are inadequate in any way, the system itself is inadequate. The basic requirements of output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner.

6.4 Table Design

The efficiency of an application using SQLITE-3 Server is mainly dependent upon the database tables, the fields in each table and joined using the fields contained in them to retrieve the necessary information. A table is a set of data elements that is organized using a model of vertical columns and horizontal rows. A table has a specified number of columns, but can have any number of rows. Each row is identified by the values appearing in a particular column subset which has been identified as a unique key index. The primary objective of a database design is fast response time to inquiries, more information at low cost, control of redundancy, clarity and ease of use, accuracy and integrity of the system fast recovery and availability of powerful end-user language.

There are mainly 20 tables in the project. They are,

1. Admin
2. User
3. Ewastecollector

4. Technician
5. Brand
6. Category
7. District
8. Place
9. Gallery
10. Type
11. Yard
12. Collectedewaste
13. Servicebill
14. Complaint
15. Ewastebooking
16. Feedback
17. Servicebook
18. Assignewastebooking
19. Assignservicebook
20. Product

1. Table: Admin

Description: This table is used to store the details of admin.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	admin_name	CharField	100	Not Null
3	admin_email	CharField	100	Not Null
4	admin_password	CharField	50	Not Null

2. Table: Brand

Description: This table is used to store brand names.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	brand_name	CharField	50	Not Null

3. Table: User

Description: This table is used to store the details of user.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	user_name	CharField	50	Not Null
3	user_gender	CharField	50	Not Null
4	user_contact	CharField	50	Not Null
5	user_email	CharField	50	Not Null
6	user_photo	FileField		Not Null
7	user_proof	FileField		Not Null
8	user_password	CharField	50	Not Null
9	user_address	CharField	100	Not Null
10	place	IntegerField		ForeignKey

4. Table: Ewastecollector

Description: This table is used to store details of e-waste collector.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	Ewastecollector_name	CharField	50	Not Null
3	Ewastecollector_contact	CharField	50	Not Null
4	Ewastecollector_email	CharField	50	Not Null
5	Ewastecollector_photo	FileField		Not Null
6	Ewastecollector_proof	FileField		Not Null
7	Ewastecollector_password	CharField	50	Not Null
8	Ewastecollector_vehiclenu	CharField	50	Not Null
9	Ewastecollector_vehiclemodel	CharField	50	Not Null
10	Ewastecollector_vehicleimg	FileField		Not Null

5. Table: Technician

Description: This table is used to store details of technician.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	technician_name	CharField	50	Not Null
3	technician_gender	CharField	50	Not Null
4	technician_contact	CharField	50	Not Null
5	technician_email	CharField	50	Not Null
6	technician_password	CharField	50	Not Null
7	technician_photo	FileField		Not Null
8	technician_experience	CharField	100	Not Null
9	technician_address	CharField	100	Not Null
10	category	IntegerField		ForeignKey

6. Table: Category

Description: This table is used to store category names.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	category_name	CharField	50	Not Null

7. Table: District

Description: This table is used to store district names.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	district_name	CharField	50	Not Null

8. Table: Place

Description: This table is used to store place names.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	place_name	CharField	50	Not Null
3	district	IntegerField		ForeignKey

9. Table: Gallery

Description: This table is used to store product images.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	gallery_image	FileField		Not Null
3	product	IntegerField		ForeignKey

10. Table: Type

Description: This table is used to store product types.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	type_name	CharField	50	Not Null

11. Table: Yard

Description: This table is used to store e-waste yard names.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	yard_name	CharField	50	Not Null

12. Table: Collectedewaste

Description: This table is used to store collected e-waste details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	collectedewastedumped_date	DateField		Not Null
3	collectedewaste_weight	CharField	100	Not Null
4	Ewaste	IntegerField		ForeignKey
5	yard	IntegerField		ForeignKey

13. Table: Servicebill

Description: This table is used to store bill details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	servicebill_date	DateField		Not Null
3	servicebill_amount	CharField	100	Not Null
4	servicebill_noofdays	CharField	100	Not Null
5	servicebill_details	CharField	200	Not Null
6	servicebill_additionaldetails	CharField	200	Not Null
7	servicebill_partamount	CharField	100	Not Null
8	assignedservicebooking	IntegerField		ForeignKey

14. Table: Complaint

Description: This table is used to store complaint details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	complaint_title	CharField	50	Not Null
3	complaint_content	CharField	50	Not Null
4	complaint_date	DateField		Not Null
5	complaint_status	IntegerField		Not Null
6	complaint_reply	CharField	100	
7	technician	IntegerField		ForeignKey
8	user	IntegerField		ForeignKey

15. Table: Ewastebooking

Description: This table is used to store e-waste collection request booking details.

Sl. no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	ewastebooking_details	CharField	50	Not Null
3	ewastebooking_date	DateField		Not Null
4	ewastebooking_status	IntegerField		Not Null
5	ewastebooking_collectionpoint	CharField	100	Not Null
6	user	IntegerField		ForeignKey

16. Table: Feedback

Description: This table is used to store user feedbacks.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	feedback_content	CharField	100	Not Null
3	technician	IntegerField		ForeignKey
4	user	IntegerField		ForeignKey

17. Table: Servicebook

Description: This table is used to store service booking details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	Servicebook_details	CharField	100	Not Null
3	Servicebook_address	CharField	100	Not Null
4	Servicebook_date	DateField		Not Null
5	Servicebook_status	IntegerField		Not Null
6	payment_status	IntegerField		Not Null
7	category	IntegerField		ForeignKey
8	brand	IntegerField		ForeignKey
9	user	IntegerField		ForeignKey

18. Table: Assignewastebooking

Description: This table is used to store e-waste booking assignment details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	aeb_status	IntegerField		Not Null
3	collector	IntegerField		ForeignKey
4	ewastebooking	IntegerField		ForeignKey

19. Table: Assignservicebook

Description: This table is used to store service booking assignment details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	asb_status	IntegerField		Not Null
3	technician	IntegerField		ForeignKey
4	servicebooking	IntegerField		ForeignKey

20. Table: Product

Description: This table is used to store product details.

Sl.no	Field Name	Data type	Size	Constraints
1	id	IntegerField		Primary key
2	Product_name	CharField	50	Not Null
3	product_image	FileField		Not Null
4	Product_description	CharField	100	Not Null
5	type	IntegerField		ForeignKey

TOOLS AND PLATFORM

7. TOOLS AND PLATFORM

7.1 Front End:HTML, CSS, JavaScript,Bootstrap

The front end of a website is what users directly interact with, built using HTML, CSS, JavaScript, and Bootstrap. HTML provides the structure of web pages, laying out elements like headings, paragraphs, and links. CSS styles these elements, adding colors, fonts, and layouts to make the site visually appealing. JavaScript enables interactivity, allowing users to engage with dynamic features like buttons and animations. Finally, Bootstrap offers us various pre-designed components and styles for creating responsive and modern layouts across various devices, making development faster and more consistent. Together, these tools create a visually appealing and user-friendly web experience.

7.2 Back End:Python

The back end of a Django application is developed using Python, a versatile and easy-to-read programming language. In Django, Python handles all server-side logic, including managing requests, processing data, and interacting with the database. Python's readability and extensive libraries make it ideal for creating robust and scalable web applications, enabling Django to provide a smooth, efficient back end experience.

7.3 FrameWork:Django

Django is a high-level web framework for building backend applications using Python. It simplifies web development by providing built-in tools for handling database management, URL routing, authentication, and more. Django follows the Model-View-Template (MVT) architecture, which promotes organized and maintainable code, making it easy to build robust and secure applications quickly.

7.4 Platform:Visual Studio Code

Visual Studio Code (VS Code) is a lightweight and powerful code editor used as the primary platform for developing and managing the project. With support for multiple programming

languages, extensions, and features like debugging, syntax highlighting, and Git integration, VS Code makes coding, testing, and deploying applications convenient and efficient. It's especially helpful for Django development, as it provides tools to streamline coding in Python and managing web files in one workspace.

7.5 Database:SQLite

SQLite is a lightweight, self-contained database engine that Django uses by default. It stores data in a single file on disk, making it simple to set up and ideal for development and small projects. SQLite requires minimal configuration and provides essential features for managing data, like creating tables, querying data, and handling relationships. Although it's not typically used for high-traffic applications, SQLite is highly effective for testing, prototyping, and small-scale applications due to its simplicity and portability.

TESTING

8. TESTING

8.1 Unit Testing

Here we test each module individually and integrated the overall system. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as "module testing". The modules of the "TECHREVIVE" are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project

8.2 Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub functions, when combined they may not perform the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface. This testing is done with simple data and the developed system has run successfully with this simple data. The need for integrated system is to find the overall system performance. After splitting the programs into units, the units were tested together to see the defects between each module and function. It is testing to one or more modules or functions together with the intent of finding interface defects between the modules or functions. Testing completed at as part of unit or functional testing, integration testing can involve putting together of groups of modules and functions with the goal of completing and verifying meets the system requirements.

8.3 System Testing

System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Testing is the first level where the System is tested as a whole. The application/System is tested in an

environment that closely resembles the production environment where the application will be finally deployed. The prerequisites for System Testing are: -

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.
- Testing should be completed in an environment closely resembling the production environment. When necessary, iterations of System Testing are done in multiple environments.

8.3.1 Acceptance Testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was found to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system.

8.3.2 Validation Testing

Data validation is the process of testing the accuracy of data. A set of rules we can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values in to a form. Now performing validation testing in system Centralized Social Welfare by undergoing validation for each tool and the validation succeeded when the software function in a manner that can be reasonably accepted, by the user.

8.3.3 Black box Testing

Testing Knowing the specified function that a product has been designed to perform, test can be conducted that demonstrates each function that is fully operational, at the same time searching for errors in each function. Black Box testing focuses on functional requirement of the software.

8.3.4 White box Testing

Testing Knowing the internal working of a product test can be conducted to ensure that “all gears mesh” that is internal operation performs according to specification and all internal components have been adequately exercised.

8.4 Sample Test Cases

TC No.	Test Steps	Expected Result	Actual Result	Status
1	Run application and navigate to login screen	Login screen is displayed. A field for entering username, a field for entering password and a button to submit should be present	Login screen has been displayed, fields for entering email address and password together with a log in button is available.	Pass
2	Enter an invalid username and invalid password and press the button	A message should be displayed stating that user name and password are invalid	A message has been displayed stating that user name and password are invalid	Pass
3	Enter a valid username and password and press the button	User must successfully login to the webpages.	A message has been displayed stating that the login successful and navigate into home page	Pass
4	Enter a valid username and leave password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the username and password	Pass
5	Leave username and password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the username and password	Pass
6	Leave username and enter a valid password and press the button	A message should be displayed stating that please enter the user name and password	A message has been displayed stating that please enter the username and password	Pass

SYSTEM IMPLEMENTATION

9.SYSTEM IMPLEMENTATION

9.1 Implementation

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system, as well as, improper installation will prevent. There are four methods for handling a system conversion. The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase. The implementation phase ends with an evaluation of the system after placing it into operation of time. The validity and proper functionality of all the modules of the developed application is assured during the process of implementation. Implementation is the process of assuring that the information system is operational and then allowing user to take over its operation for use and evaluation. Implementation is the stage in the project where the theoretical design is turned into a working system. The implementation phase constructs, installs and operated the new system. The most crucial stage in achieving a new successful system is that it works effectively and efficiently.

9.2 Problem Statement

In today's technology-driven world, the reliance on electronic gadgets has surged, leading to significant challenges in managing their maintenance and disposal. Customers often struggle to find reliable repair services, resulting in inconvenience and frustration due to lengthy processes, limited availability, and a lack of transparency regarding service status and pricing.

Additionally, improper disposal of electronic waste poses serious environmental risks, as many users are unaware of responsible recycling practices or have limited access to convenient e-waste collection options. Existing solutions are fragmented, forcing users to navigate multiple channels for repairs and disposal, which leads to inefficiencies such as missed appointments and increased electronic waste in landfills. Thus, there is an urgent need for an integrated platform that streamlines the booking of gadget repairs and facilitates efficient e-waste management, offering convenience, real-time tracking, and education on responsible disposal, ultimately promoting sustainability in gadget usage and waste management.

9.3 Problem Definition

The problem at hand is the increasing reliance on electronic gadgets, which, while enhancing convenience and connectivity, also creates significant challenges in their maintenance and responsible disposal. Customers frequently encounter difficulties in accessing reliable repair services, facing issues such as inconvenient booking processes, lack of transparency regarding service availability, and inadequate communication about repair statuses. Furthermore, the improper disposal of e-waste poses severe environmental hazards, with many users lacking awareness of proper recycling practices and struggling to find accessible e-waste collection services. The fragmented nature of existing solutions exacerbates these challenges, resulting in inefficiencies and increased electronic waste in landfills. Thus, there is a critical need for a comprehensive platform that integrates gadget repair services with e-waste management, simplifying the user experience and fostering responsible electronic waste disposal.

CONCLUSION

10 . CONCLUSION

The project was successfully completed within the time span allotted. All the modules are tested separately and put together to form the main system. Finally, the modules are tested with real data and it worked successfully. Thus the system has fulfilled the entire objective defined.

This project will help the customer to book their electronic appliances related services in TECHREVIVE. Our goal of developing this “TECHREVIVE” has come to get a good result without many defects.

The main motive for developing this system is for the welfare of the society by giving respect to the time of each person and to protect the environment from electronic wastes related pollution. Since our daily live depends on electronic gadgets it is required to provide services related to it at any time in best manner.

FUTURE ENHANCEMENTS

The system has been designed in such a way that it can be modified with very little effort when such needs arise in the future. New features can be added with slight modifications of software which make it easy to expand the scope of this project. Though the system is working on various assumptions, it can be modified easily to any kind of requirements.

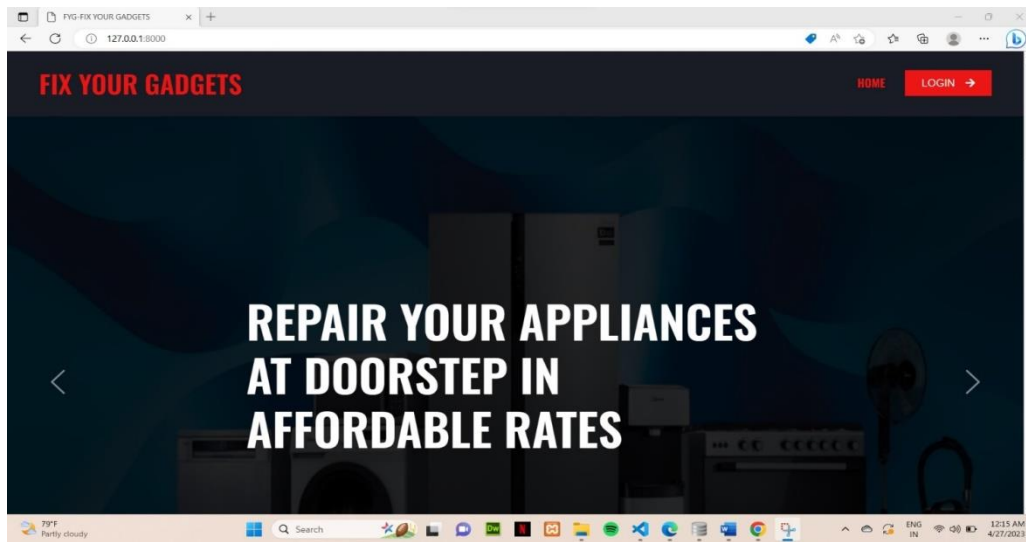
Even though we have tried our best to present the information effectively and efficiently, yet there can be further enhancement in the application. We have taken care of all the critical aspects, which were needed to be taken care of. Because of fast changes in the world of programming this system will gradually get outdated and less effective. For the time being it's possible to overcome problems by amendments and minor modifications to acknowledge the need of fundamental design. Though the new system provides base for improving the efficiency of operations, there are a lot of future enhancements that can be added to this project. Keeping this in view, a provision has been made in the system to facilities easy modification updating in the future. Any modification will not affect the normal working of the system. It can also be converted into a mobile application.

APPENDICES

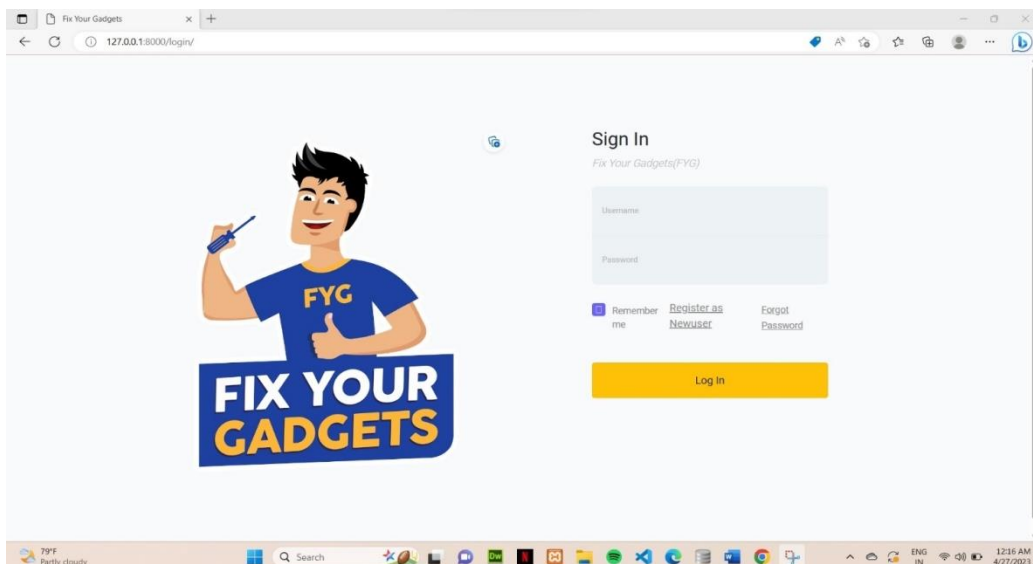
11. APPENDICES

APPENDICES A: Sample Screens

Home Page



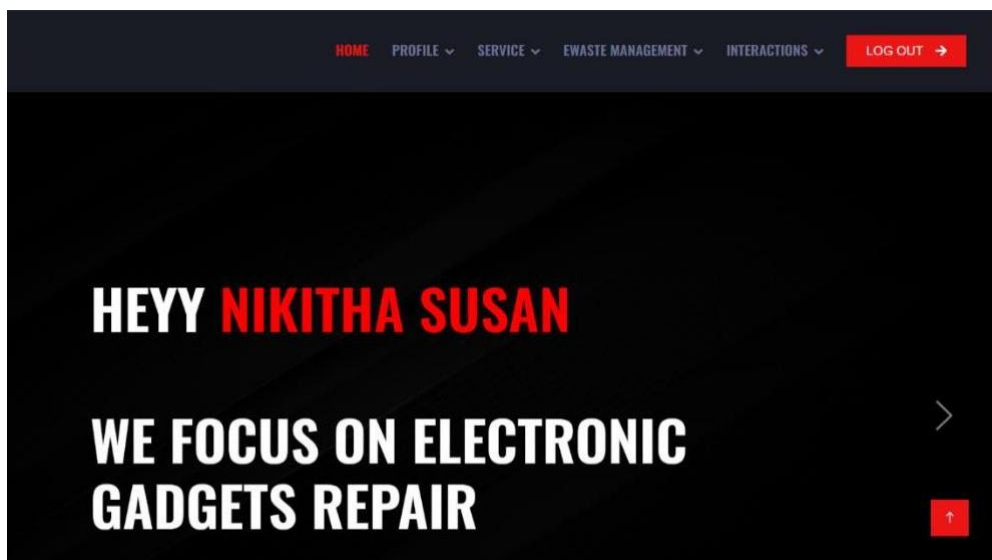
Login Page



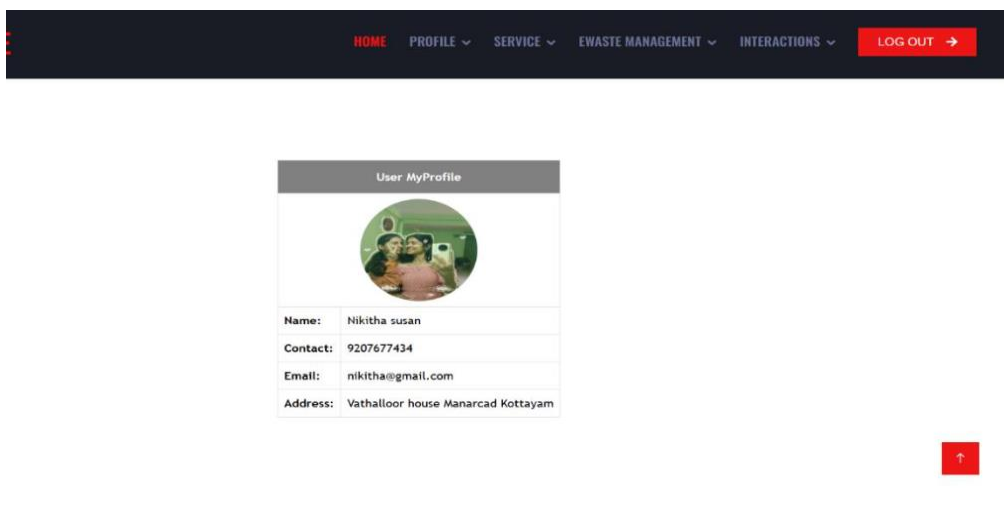
User Registration

User Registration	
Name:	<input type="text" value="Nikitha susan"/>
Gender:	<input type="radio"/> Male <input checked="" type="radio"/> Female
Contact:	<input type="text" value="9207677434"/>
Email:	<input type="text" value="nikitha@gmail.com"/>
District:	<input type="text" value="Kottayam"/>
Place:	<input type="text" value="Manarcad"/>
Photo:	<input type="button" value="Choose File"/> IMG_202404...085524.jpg
Proof:	<input type="button" value="Choose File"/> download (3).jpg

User Homepage



User Profile



Book Service Request

TECHREVIVE[HOME](#) [PROFILE](#) [SERVICE](#) [EWASTE MANAGEMENT](#) [INTERACTIONS](#) [LOG OUT](#)

Book Service At Your Doorstep

Category:

Mobile Phones

Brand:

Samsung

Details:

Display Crack Issue

Address:

Vathalloor (H)
Manarcad P.O
Kottayam
Mob:9207677321

Submit

Cancel

Book EwasteCollection Request

Book Service At Your Doorstep

Category:

Mobile Phones

Brand:

Samsung

Details:

Display Crack Issue

Address:

Vathalloor (H)
Manarcad P.O
Kottayam
Mob:9207677321

Submit

Cancel

User Complaint

User Complaints

Complaint Title:

Service Issue

Technician:

Joseph Thampi

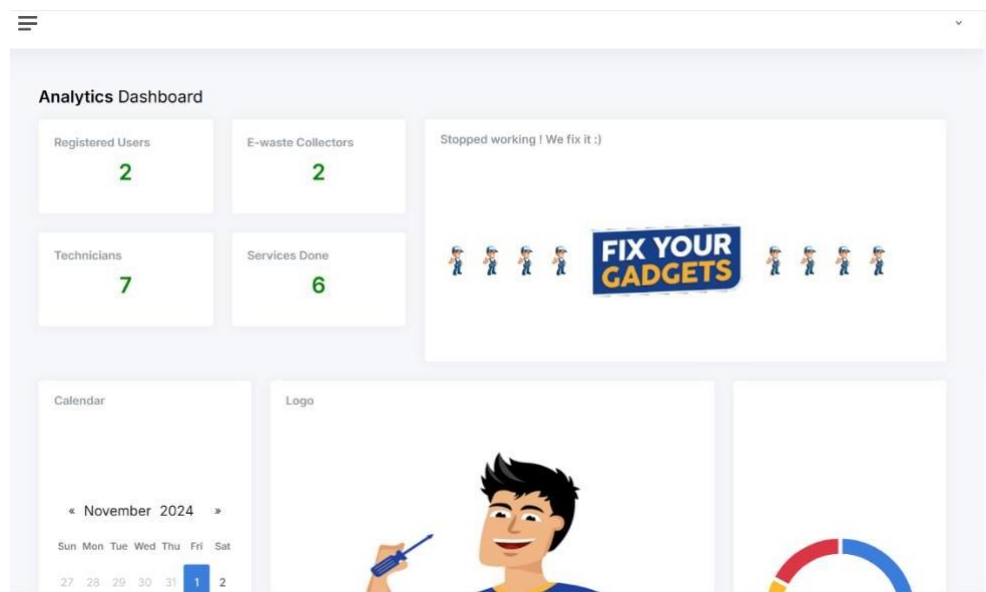
Content:

Not get a proper service for my problem.

Submit

Cancel

Admin Dashboard



Add Yards

The "Add Yards" form includes a text input for "Yard name" and a "Submit" button. Below the form is a table listing existing yards:

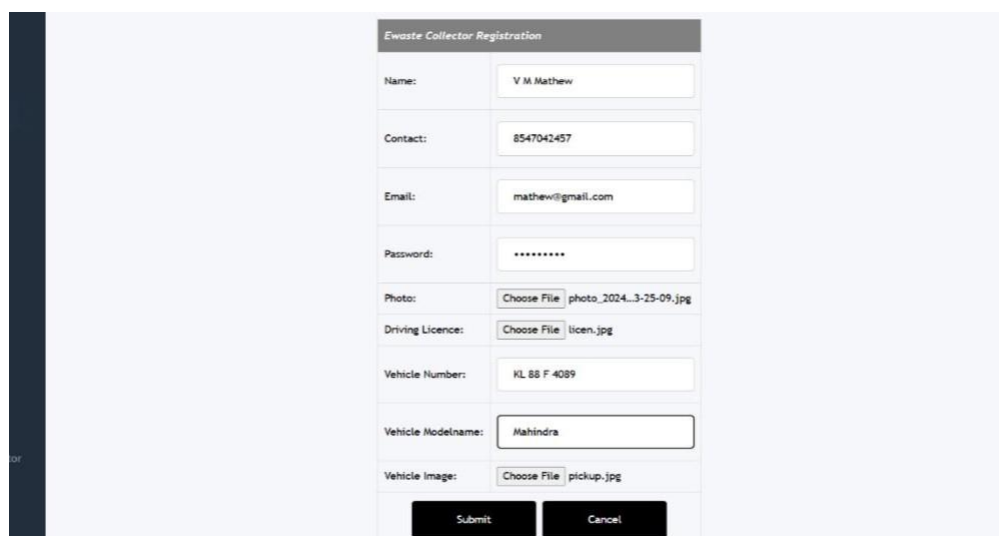
Sl.No	Yard	Action
1	Yard Phase I	Delete
2	Yard Phase II	Delete
3	Yard Phase III	Delete
4	Yard Phase IV	Delete
5	Yard Phase V	Delete
6	Yard Phase VI	Delete
7	Yard Phase VII	Delete
8	Yard Phase VIII	Delete

Technician Registration

The "Technician Registration" form includes the following fields:

- Name:** Jeril Joy
- Gender:** ☒ Male ☐ Female
- Contact:** 7558802241
- Email:** jeril@gmail.com
- Password:** [Masked]
- Photo:** Choose File | photo_2024_3-25-09.jpg
- Skill On:** Air Conditioners
- Experience:** 4 years
- Address:** Churathil H, Cherappukal P.O, Kottayam

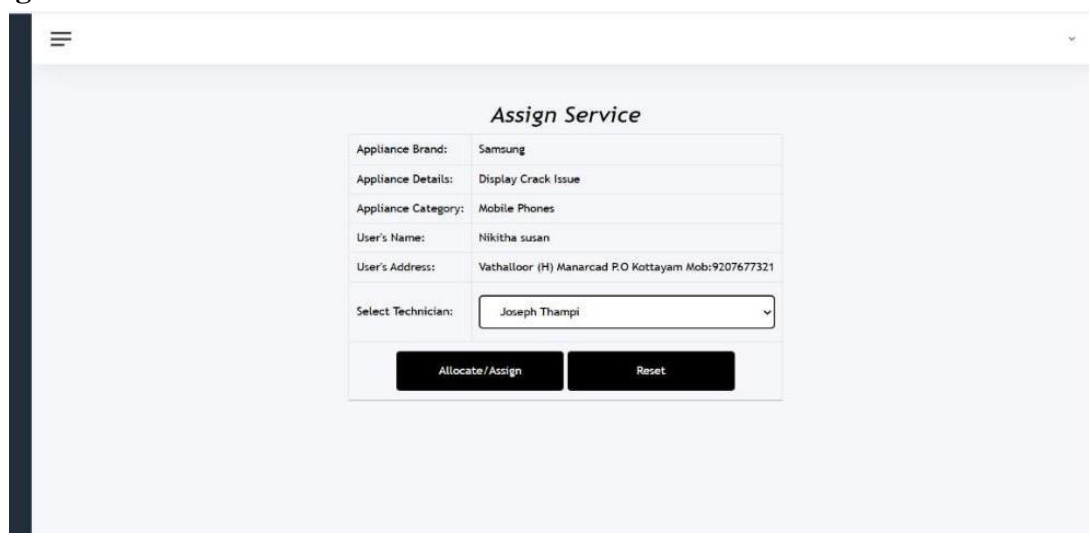
Ewaste Collector Registration



The screenshot shows a web form titled "Ewaste Collector Registration". It contains several input fields for personal and vehicle information, each with a "Choose File" button for uploads. The fields are: Name (V M Mathew), Contact (8547042457), Email (mathew@gmail.com), Password (masked with asterisks), Photo (photo_2024_3-25-09.jpg), Driving Licence (licen.jpg), Vehicle Number (KL 88 F 4089), Vehicle Modelname (Mahindra), and Vehicle Image (pickup.jpg). At the bottom are "Submit" and "Cancel" buttons.

Ewaste Collector Registration	
Name:	V M Mathew
Contact:	8547042457
Email:	mathew@gmail.com
Password:	*****
Photo:	Choose File photo_2024_3-25-09.jpg
Driving Licence:	Choose File licen.jpg
Vehicle Number:	KL 88 F 4089
Vehicle Modelname:	Mahindra
Vehicle Image:	Choose File pickup.jpg
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>	

Assign Booked Services



The screenshot shows a web form titled "Assign Service". It contains a table with service details and a dropdown for selecting a technician. The table has rows for Appliance Brand (Samsung), Appliance Details (Display Crack Issue), Appliance Category (Mobile Phones), User's Name (Nikitha susan), and User's Address (Vathalloor (H) Manarcad P.O Kottayam Mob:9207677321). The Select Technician dropdown is set to "Joseph Thampi". At the bottom are "Allocate/Assign" and "Reset" buttons.

Assign Service	
Appliance Brand:	Samsung
Appliance Details:	Display Crack Issue
Appliance Category:	Mobile Phones
User's Name:	Nikitha susan
User's Address:	Vathalloor (H) Manarcad P.O Kottayam Mob:9207677321
Select Technician:	Joseph Thampi
<input type="button" value="Allocate/Assign"/> <input type="button" value="Reset"/>	

Assign Booked Ewaste Collection



The screenshot shows a web form titled "Assign Ewaste Collection". It contains a table with collection details and a dropdown for selecting a collector. The table has rows for User's Name (Nikitha susan), Date (Nov. 6, 2024), Details (Waste include tv, mobile phones and small electronic wastes), and Collection Point (Vathalloor H Manarcad P.O Kottayam Mob:9207677324). The Select Collector dropdown is set to "V M Mathew". At the bottom are "Allocate/Assign" and "Reset" buttons.

Assign Ewaste Collection	
User's Name:	Nikitha susan
Date:	Nov. 6, 2024
Details:	Waste include tv, mobile phones and small electronic wastes
Collection Point:	Vathalloor H Manarcad P.O Kottayam Mob:9207677324
Select Collector:	V M Mathew
<input type="button" value="Allocate/Assign"/> <input type="button" value="Reset"/>	

Technician homepage



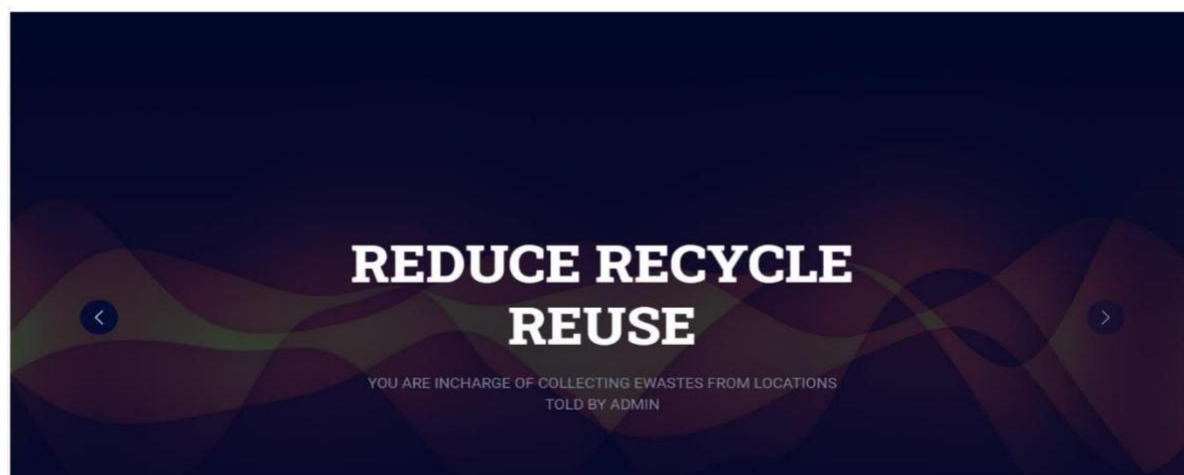
View Assigned Services

Your Allocated Services						
Sl.No	Brand	Category	Details	Username	Address	Action
1	Samsung	Mobile Phones	Display Crack Issue	Nikitha susan	Vathalloor (H) Manarcad P.O Kottayam Mob:9207677321	Confirm Service Reject Service

Generate Bill

Generate Bill	
User's Name:	Nikitha susan
Gadget Category:	Mobile Phones
Gadget Brand:	Samsung
Assumed Issue:	Display Crack Issue
Actual Issue:	Display line Issue
Parts Replaced:	Display
Parts Amount:	7000
Service Charge Per Day:	1500
Number of Days To Perform Service:	1

Ewaste Collector homepage



View Assigned Ewaste Collection

E-Waste Collection Requests						
Sl.No	Date	Username	Contact	Details	Collection Point	Action
1	Nov. 6, 2024	Nikitha susan	9207677434	Waste include tv, mobile phones and small electronic wastes	Vathalloor H Manarcad P.O Kottayam Mob:9207677324	Confirm Reject

Completed Collection List

Confirmed E-Waste Collection Requests						
Sl.No	Date	Username	Contact	Details	Collection Point	Action
1	Nov. 6, 2024	Nikitha susan	9207677434	Waste include tv, mobile phones and small electronic wastes	Vathalloor H Manarcad P.O Kottayam Mob:9207677324	Already Completed

APPENDICES B:Sample Code

```
from django.shortcuts import render,redirect
from Admin.models import *
from Guest.models import *
from User.models import *
from Technician.models import Servicebill
# Create your views here.

def hompagetechnician(request):
    if 'tid' in request.session:
        techn=Technician.objects.get(id=request.session["tid"])
        return render(request,"Technician/Technicianhomepage.html",{ 'data':techn })
    else:
        return redirect('guest:Home')

def TechnicianMyProfile(request):
    if 'tid' in request.session:
        techn=Technician.objects.get(id=request.session["tid"])
        return render(request,"Technician/TechnicianMyProfile.html",{ 'data':techn })
    else:
        return redirect('guest:Home')

def TechnicianEditProfile(request):
    if 'tid' in request.session:
        techn=Technician.objects.get(id=request.session["tid"])
        if request.method=="POST":
            techn.technician_name=request.POST.get('txt_name')
            techn.technician_contact=request.POST.get('txt_contact')
            techn.technician_address=request.POST.get('txt_address')
            techn.save()
            return redirect("technician:technicianprofile")
        else:
            return render(request,"Technician/TechnicianEditProfile.html",{ 'data':techn })
    else:
        return redirect('guest:Home')

def TechnicianChangePass(request):
    if 'tid' in request.session:
        if request.method=="POST":
```

```
techcount=Technician.objects.filter(id=request.session["tid"],technician_password=request.P
OST.get('txt_curr')).count()
    if techcount>0:
        techn=Technician.objects.get(id=request.session["tid"],technician_password=request
t.POST.get('txt_curr'))
        techn.technician_password=request.POST.get('txt_new')
        techn.save()
        return redirect("technician:technicianhome")
    else:
        return render(request,"Technician/TechnicianChangePassword.html")
else:
    return redirect('guest:Home')

def TechnicianAssignedService(request):
    if 'tid' in request.session:
        serv=Assignservicebook.objects.filter(technician=request.session["tid"])
        return render(request,"Technician/ViewAllocatedServices.html",{ 'res':serv })
    else:
        return redirect('guest:Home')

def ConfirmServ(request,cid):
    if 'tid' in request.session:
        serv=Assignservicebook.objects.get(id=cid)
        serv.asb_status=2
        serv.save()
        return redirect('technician:Myassignedservice')
    else:
        return redirect('guest:Home')

def DeleteServ(request,cid):
    if 'tid' in request.session:
        serv=Assignservicebook.objects.get(id=cid)
        bokid=serv.servicebooking_id
        sbok=Servicebook.objects.get(id=bokid)
        sbok.Servicebook_status=1
        sbok.save()

        serv.delete()
        return redirect('technician:Myassignedservice')
    else:
        return redirect('guest:Home')
```

```
def ConfirmedServices(request):
    if 'tid' in request.session:
        serv=Assignservicebook.objects.filter(technician=request.session["tid"])
        return render(request,"Technician/ViewConfirmedServices.html",{ 'res':serv })
    else:
        return redirect('guest:Home')

def ServiceBill(request,bid):
    if 'tid' in request.session:
        serv=Assignservicebook.objects.get(id=bid)
        if request.method=="POST":
            Servicebill.objects.create(servicebill_amount=request.POST.get('txt_amount'),service
            bill_noofdays=request.POST.get('txt_days'),servicebill_partamount=request.POST.get('txt_pa
            rts'),servicebill_additionaldetails=request.POST.get('txt_replaced'),servicebill_details=request .POST.get('txt_issue'),assignedservicebooking=serv)

            serv.asb_status=3
            serv.save()
            bokid=serv.servicebooking_id
            sbok=Servicebook.objects.get(id=bokid)
            sbok.Servicebook_status=4
            sbok.save()
            return redirect('technician:viewconfimedservice')
        else:

            return render(request,"Technician/GenerateBill.html",{ 'data':serv })
        else:
            return redirect('guest:Home')

def logout(request):

    del request.session['tid']
    return redirect('guest:Home')
```


EwasteCollector registration

```
{% extends 'Admin/Head.html' %}
{% load static %}
{% block content %}
<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
    scale=1.0"> <title>Document</title>
</head>
<body>
  <div id="tab">

    <br>
    <br>
    <form method="post" enctype="multipart/form-data">
      {% csrf_token %}
      <table border="1" align="center">
        <tr>
          <th colspan="2"><i><b>Ewaste Collector Registration</b></i></th>
        </tr>
        <tr>
          <td>Name:</td>
          <td><input type="text" name="txt_name" required="required"></td>
        </tr>
        <tr>
          <td>Contact:</td>
          <td><input type="text" name="txt_num" required="required"></td>
        </tr>
        <tr>
          <td>Email:</td>
          <td><input type="email" name="txt_email" required="required"></td>
        </tr>
        <tr>
          <td>Password:</td>
          <td><input type="password" name="txt_pass" required="required"></td>
        </tr>
      </table>
    </form>
  </div>
</body>
</html>
```

```

</tr>
<tr>
  <td>Photo:</td>
  <td><input type="file" name="file_photo" required="required"></td>
</tr>
<tr>
  <td>Driving Licence:</td>
  <td><input type="file" name="file_lphoto"
required="required"></td> </tr>

<tr>
  <td>Vehicle Number:</td>
  <td><input type="text" name="txt_vnumber" required="required"></td>
</tr>
<tr>
  <td>Vehicle Modelname:</td>
  <td><input type="text" name="txt_vname" required="required"></td>
</tr>
<tr>
  <td>Vehicle Image:</td>
  <td><input type="file" name="file_vphoto" required="required"></td>
</tr>
<tr>
  <td colspan="2"><center><input type="submit"
name="btn_submit">&nbsp;&nbsp;&nbsp;<input type="reset" name="btn_cancel"
value="Cancel"></center></td>
</tr>
</table>
<br>
<br>
<br>
<table border="1" align="center">
  <tr><th colspan="10"><i><center>Ewaste
Collectors</center></i></th></tr> <tr>
  <th>Sl No</th>
  <th>Name</th>
  <th>Contact</th>
  <th>Email</th>
  <th>Photo</th>
  <th>Driving Licence</th>
  <th>Vehicle Number</th>
  <th>Vehicle Modelname</th>

```

```

        <th>Vehicle Image</th>
        <th>Action</th>
    <tr>
    {% for i in result %}
    <tr>
        <td>{{ forloop.counter }}</td>
        <td>{{ i.EwasteCollector_name }}</td>
        <td>{{ i.EwasteCollector_contact }}</td>
        <td>{{ i.EwasteCollector_email }}</td>
        <td></td>

        <td></td> <td>{{ i.EwasteCollector_vehiclenu }}</td>
        <td>{{ i.EwasteCollector_vehiclemodel }}</td>
            <td></td>
        <td><a href="{% url 'webadmin:dele wastec' i.id
            %}">Delete</a></td> </tr>
    {% endfor %}
    </table>
</form>
</div>
<br>
<br>
<br><br>
<br>
<br>
<script>
    {% if msg %}
    alert("{{ msg }}")
    // window.location=""
    {% endif %}
</script>
</body>
</html>
{% endblock content %}

```

APPENDICES C: Hardware and Software Requirements

● Hardware Requirements

Processor	:	Intel Core i3
Random Access Memory	:	4GB or above
Hard Disk/SSD	:	240GB
Monitor	:	Color Monitor
Keyboard	:	Standard
Video	:	800X600 256 colors

● Software Requirements

Operating System	:	Windows 11 or Higher
Front End	:	HTML,CSS,Javascript,Bootstrap
Environment	:	Django
Database	:	SQLite
Operating System	:	Windows 11
Documentation Tool	:	MS Word

APPENDICES D: Bibliography

Book References:

Taming Python By Programming by Dr. Jeeva Jose

Publications:

1. **Dauzon, S., Bendoraitis, A., & Ravindran, A.** (2016). *Django: web development with Python*. Packt Publishing Ltd . [Publisher Site](#) | Google Scholar
2. **Shaw, B., Badhwar, S., Bird, A., KS, B. C., & Guest, C.** (2021). *Web Development with Django: Learn to build modern web applications with a Python-based framework*. Packt Publishing Ltd. [Publisher Site](#) | Google Scholar
3. **Vincent, W. S.** (2022). *Django for Beginners: Build websites with Python and Django*. WelcomeToCode.. [Publisher Site](#) | Google Scholar

4. **Kiddee, P., Naidu, R., & Wong, M. H.** (2013). Electronic waste management approaches: An overview. *Waste management*, 33(5), 1237-1250. [Publisher Site](#) | Google Scholar
5. **Mundada, M. N., Kumar, S., & Shekdar, A. V.** (2004). E-waste: a new challenge for waste management in India. *International journal of environmental studies*, 61(3), 265-279. [Publisher Site](#) | Google Scholar
6. **Sonego, M., Echeveste, M. E. S., & Debarba, H. G.** (2022). Repair of electronic products: Consumer practices and institutional initiatives. *Sustainable Production and Consumption*, 30, 556-565. [Publisher site](#) | Research gate

Website References:

Retrieved from www.stackoverflow.com

Retrieved from www.w3schools.com

Retrieved from www.gihub.com

Retrieved from www.python.org

Retrieved from www.tutorialspoint.com

Retrieved from www.javatpoint.com