CHINMAYA COLLEGE OF ARTS, COMMERCE AND SCIENCE

Layam road, Tripunithura – 682301

(Affiliated to Mahatma Gandhi University, Kottayam)

BACHELOR OF COMPUTER APPLICATIONS

MINI PROJECT

ON

CleanWave - LAUNDRY MANAGEMENT SYSTEM

Submitted By,

ASSNA V.S

Register No: 220021084852

CHINMAYA COLLEGE OF ARTS, COMMERCE AND SCIENCE

Layam road, Tripunithura – 682301

(Affiliated to Mahatma Gandhi University, Kottayam)

CERTIFICATE

This is to certify that the Mini Project Report entitled

CleanWave - LAUNDRY MANAGEMENT SYSTEM

has been submitted by

ASSNA V.S

Register No: 220021084852

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

BACHELOR OF COMPUTER APPLICATIONS

MAHATMA GANDHI UNIVERSITY

During the academic year 2024-2025

Submitted for the University Examination held on .....

Principal ***Head of the Department***

External Examiner ***Project Guide***

ACKNOWLEDGEMENT

By blessing and permission of Almighty God, I was able to complete this work successfully. My sincere thanks to our Principal in charge, **Mrs.Rosy Joice Lopez** for her overwhelming and moralsupport extended towards us.

I would like to thank our head of the department **Mrs. Nisha Sanjay** for her constant encouragements and support for the completion of our project.

I would like to express grateful thanks to **Mrs. Remilda Rajan (Project Guide)**, for her valuable guidance and support.

I would also like to express my sincere thanks to all my teachers, **Mr. Vishnu Mohanan, Mrs. Ranimol V.G , Mrs. Sharmila Francis and Mrs. Andal V** for their timely assistance andadvice offered to us to make this project a success.

Finally, I thank my parents for their boundless support and for making our lives so easy and for helping to tackle all those difficulties in life.

DECLARATION

I **ASSNA V.S** , hereby declare that the Mini Project entitled **CleanWave-LAUNDRY MANAGEMENT SYSYTEM** submitted to **Mahatma Gandhi University, Kottayam** in partial fulfillment of the requirements for the **Bachelor's degree in Computer Applications** is a record of original work done by me during the period of study at CHINMAYA COLLEGE OF ARTS, COMMERCE AND SCIENCE Tripunithura under the supervision and guidance of **Mrs. Remilda Rajan (Project Guide)**, Department of Computer Application and that this project work has not formed the basis for the award of any diploma/associates-ship/fellowship or similar title to any candidate of any university.

Place: Tripunithura **ASSNA V.S**

Date: Register:220021084852

\

SYNOPSIS

CleanWave is an innovative web-based laundry management system designed to enhance the efficiency of managing a variety of laundry services, including Wash + Fold, Wash + Iron, Dry Clean, Steam Iron, and Stain Clear. Developed using PHP, CSS, and JavaScript on a WAMP server, CleanWave provides a user-friendly interface that caters to both customers and administrators. Customers can effortlessly explore service offerings, view detailed pricing, and place orders online, ensuring a smooth and convenient experience.The administrative dashboard allows for comprehensive management of services, orders, and customer feedback, enabling seamless oversight of operations. Key features include an item management system that organizes services by categories such as men, women, kids, and household, allowing for easy updates and modifications.A standout aspect of CleanWave is its donation module, which empowers users to donate their old clothes to those in need through the website. This feature simplifies the process of giving back to the community, allowing users to contribute to charitable causes easily and efficiently. By promoting social responsibility and community support, CleanWave not only addresses the logistical challenges of laundry services but also fosters a sense of goodwill among its users.With a robust backend supported by MySQL, the system efficiently handles data storage and retrieval, ensuring quick access to information regarding orders, items, and user feedback. CleanWave stands out as a comprehensive solution for modern laundry management, combining convenience, organization, and community engagement into one effective platform. This project not only streamlines laundry service management but also encourages users to participate in charitable activities through the donation feature, making a positive impact on the lives of those in need

TABLE OF CONTENTS

1. Introduction 1
    1.1.Project Overview 1
2. System Analysis 3
    2.1.Problem Analysis 3
        2.1.1. Existing System 4
        2.1.2. Proposed System 4
        2.1.3. Feasibility Study 4
            2.1.3.1. Economic Feasibility 5
            2.1.3.2. Technical Feasibility 5
            2.1.3.3. Behavioral Feasibility 6
    2.2.Requirement specification 6
        2.2.1. Software requirement Specification 7
    2.3.Hardware and Software selection and justification 8
        2.3.1. Hardware Specification 8
        2.3.2. Software Specification 8
    2.4.Use case diagram 12
    2.5.Data Flow Diagram 13
    2.6.Entity Relationship Diagram 16
3. System Design 18
    3.1.Structured Design methodologies 18
    3.2.User Interface Design 18
    3.3.Output Design 19
    3.4.Database Design 20
        3.4.1. Data and Integrity Constraint 20
    3.5.Table Design 21
4. Coding 23
    4.1.Program code preparation 23
5.Implementation of Security 24
    5.1. Data Security 24
    5.2. Users and Access Rights 24
6. System Testing 25
    6.1.Unit Testing 25
    6.2.Integration Testing 26
    6.3.User Acceptance Testing 26
    6.4.Test Case Design 26
    6.5.Test Report and Debugging 27
7. System Implementation and maintenance 28
    7.1.System maintenance 28
8. Scope of project 29
9. Future enhancement 30
10. Conclusion 31
11. Bibliography 32
    Appendix I – coding 33
    Appendix II – Screenshots – Demo of Project 50

1. INTRODUCTION

1.1 PROJECT OVERVIEW

CleanWave is a comprehensive web-based laundry management system designed to streamline various laundry services while promoting community engagement through its innovative donation module. The project aims to address the challenges of traditional laundry services by providing a user-friendly interface that caters to both customers and administrators. By leveraging modern web technologies like PHP, CSS, and JavaScript, and utilizing a robust backend powered by MySQL, CleanWave offers an efficient and effective solution for managing laundry operations.

The system's architecture ensures seamless navigation and quick access to essential features, allowing customers to easily explore services and place orders. Administrators benefit from a centralized dashboard that simplifies the management of services, orders, and customer interactions. Overall, CleanWave not only enhances operational efficiency but also fosters a sense of community through its commitment to social responsibility.

Major Modules of CleanWave

1. Service Management:

- This module allows administrators to manage various laundry services offered by CleanWave, including Wash + Fold, Wash + Iron, Dry Clean, Steam Iron, and Stain Clear. Administrators can add, update, or remove services, ensuring that the offerings remain relevant and up-to-date.

2. Customer Order Management:

- Customers can place orders through an intuitive interface, selecting their desired services and providing necessary details. This module captures customer information, service requests, and payment details, ensuring a smooth transaction process. Administrators can monitor, process, and update the status of orders through the admin dashboard.

3. Item Management:

- The item management module categorizes laundry services by demographic segments such as men, women, kids, and household. This organization allows customers to browse services more effectively and enables administrators to manage service offerings efficiently. Items can be added or modified easily, ensuring accurate pricing and descriptions.

4. Feedback Management:

- CleanWave includes a feedback management system that allows customers to submit reviews and ratings for services. This module helps administrators gather valuable insights into customer satisfaction, enabling them to make data-driven improvements to services.

5. Donation Module:

- The donation module is a unique feature of CleanWave, allowing users to donate their old clothes to those in need directly through the website. This module simplifies the donation process, encouraging community support and social responsibility. Users can easily submit their donations, and administrators can manage and track these contributions effectively.

6. User Authentication and Management:

- This module handles user registration, login, and account management. Customers can create accounts to access personalized features, view order history, and manage their profiles. Administrators can oversee user accounts, ensuring secure access to the system.

7. Reporting and Analytics:

- CleanWave provides reporting features that allow administrators to analyze various metrics related to service usage, customer satisfaction, and donations. This data-driven approach aids in strategic decision-making and helps improve overall service quality.

By integrating these modules, CleanWave effectively addresses the diverse needs of its users while promoting operational efficiency and community involvement. This project represents a significant step forward in modernizing laundry service management and encouraging charitable contributions through a seamless online platform.

2. SYSTEM ANALYSIS

A complete understanding of software requirement is essential to the success of software development effort. No matter how well design/well coded a poorly analyzed and specified program will disappoint the user and bring the grief to the user. The requirement analysis task is a process of discovery, refinement, modeling and specification. The software initially established by the system engineer and refined during software project, planning is refined in detail. Models of the required information and control flow, operational and data content are created. Alternative solutions are analyzed and allocation to various software elements. Requirements analysis, software-engineering task that bridge the gap between system level software allocation and software design .Requirements analysis enables to specify software function and performance indicates software's interface with other system elements and established design constraints that the software must meet. Requirement analysis allows the software engineers to refine. The main components of making software are:

- System and software requirements analysis
- Design and implementation of software
- Ensuring, verifying and maintaining software integrity

2.1 PROBLEM ANALYSIS

1. Problem analysis is the process of understanding real-world problems like the difficulty in finding the required product at their best price within less time. Getting to know whether the product we are looking to buy is available or not
2. The goal of problem analysis is to gain a better understanding, before development begins, of the problem being solved.
3. To identify the root cause, or the problem behind the problem, ask the people directly involved.
4. Identifying the actors on the system is a key step in problem analysis.
5. On analysis we need 2 actors for time as a start of computerization where admin can manage the details of shop and customers can look for the dress and order

2.1.1 EXISTINGSYSTEM

Before the development of CleanWave, traditional laundry services operated primarily through manual processes, often relying on in-person interactions and paper-based systems for order management. Customers typically visited physical locations to place orders, which limited convenience and accessibility. Service offerings were often not clearly communicated, leading to confusion regarding pricing and available options. Additionally, feedback mechanisms were lacking, making it challenging for service providers to gauge customer satisfaction and implement improvements.

The existing systems did not incorporate features that facilitated community engagement, such as clothing donations, which limited their social impact. Furthermore, the reliance on manual record-keeping posed risks of data loss and inaccuracies, affecting overall operational efficiency. The absence of a centralized digital platform hindered the ability to manage orders, track customer interactions, and analyze service performance effectively. Recognizing these limitations, CleanWave aims to revolutionize laundry management by introducing a comprehensive online solution that enhances customer convenience, streamlines operational processes, and promotes community involvement through its innovative features.

2.1.2 PROPOSED SYSTEM

The proposed system of CleanWave is a modern, web-based laundry management platform designed to streamline operations and enhance user experience. By integrating key features such as an intuitive service management interface, a seamless order processing system, and a robust feedback mechanism, CleanWave aims to provide customers with a convenient and efficient way to manage their laundry needs online. The inclusion of a unique donation module allows users to contribute to the community by donating old clothes effortlessly, fostering social responsibility. With a strong backend powered by MySQL, the system ensures reliable data management and security. Overall, CleanWave represents a significant upgrade over traditional laundry services, delivering a comprehensive solution that combines convenience, efficiency, and community engagement in a single platform.

2.1.3 FEASIBILITY STUDY

The feasibility study for CleanWave assesses the viability of developing the proposed laundry management system by evaluating technical, economic, operational, and legal aspects. Technical feasibility indicates that the project is achievable using existing web technologies such as PHP, CSS, JavaScript, and MySQL, ensuring robust performance and scalability. The team possesses the necessary skills and expertise to implement these technologies effectively. Economic feasibility analysis suggests that the project is financially viable, with potential cost savings from increased operational efficiency and enhanced customer satisfaction leading to higher revenue generation.

Operational feasibility demonstrates that CleanWave aligns with user needs, as it addresses the limitations of traditional laundry services by providing a more convenient and efficient solution. The user-friendly interface and integrated donation module cater to customer expectations and promote community engagement. Lastly, legal feasibility confirms that the system complies with data protection regulations and industry standards, ensuring user privacy and security. Overall, the feasibility study indicates that CleanWave is a practical and beneficial initiative that can significantly improve laundry management while fostering social responsibility within the community.

2.1.3.1 ECONOMICAL FEASIBILITY

The economic feasibility of CleanWave evaluates the financial viability of developing and implementing the laundry management system by analyzing potential costs and projected benefits. Initial investments will primarily focus on software development, infrastructure setup, and marketing efforts to attract users. However, the anticipated cost savings from automating processes such as order management and customer feedback collection are expected to offset these initial expenditures significantly. By streamlining operations, CleanWave can reduce labor costs and minimize errors associated with manual handling, leading to increased efficiency and productivity.

Furthermore, the introduction of an online platform will expand the customer base by offering services beyond geographical limitations, potentially increasing revenue through a higher volume of orders. The unique donation module not only enhances community engagement but can also attract partnerships with charitable organizations, further boosting the system's visibility and user participation. Overall, the projected financial benefits, including enhanced customer satisfaction, increased revenue generation, and long-term cost savings, suggest that CleanWave is a sound economic investment that can yield positive returns and contribute to sustainable growth.

2.1.3.2 TECHNICAL FEASIBILTY

The technical feasibility of **CleanWave** assesses the practicality of developing the proposed laundry management system using available technologies and resources. The project is designed to utilize widely adopted web development technologies, including PHP for server-side scripting, CSS for styling, and JavaScript for client-side interactivity, ensuring a robust and responsive user experience. The choice of MySQL as the database management system provides a reliable and efficient solution for storing and managing data related to services, orders, and user interactions.

The development team possesses the requisite skills and expertise in these technologies, which further supports the technical feasibility of the project. Additionally, the architecture of CleanWave is designed to be scalable, allowing for future enhancements and the incorporation of new features as user demands grow. The system will also leverage secure coding practices and comply with industry standards to ensure data protection and user privacy. Given these factors, the technical feasibility study indicates that CleanWave can be effectively developed and deployed, providing a solid foundation for the successful implementation of the laundry management system.

2.1.3.3 BEHAVIOURAL FEASIBILITY

The behavioral feasibility of CleanWave examines the willingness and ability of potential users—both customers and administrators—to adopt the proposed laundry management system. The project aims to address the common frustrations associated with traditional laundry services, such as lengthy wait times, lack of service transparency, and limited accessibility. By providing an intuitive, user-friendly interface and streamlined processes, CleanWave is designed to enhance user experience and promote acceptance among customers who seek convenience and efficiency in managing their laundry needs.

User engagement during the development phase, including feedback from focus groups and surveys, has indicated a strong interest in features like online order placement, real-time tracking, and the ability to donate clothes easily through the platform. Additionally, the inclusion of a feedback management system will empower users to share their experiences and suggestions, fostering a sense of community and involvement in shaping the service. Administrators are likely to embrace the system due to its ability to simplify operational tasks and improve overall service quality. Overall, the behavioral feasibility assessment suggests that CleanWave is well-positioned to gain user acceptance and encourage widespread adoption by meeting the evolving needs and preferences of its target audience.

2.2 REQUIREMENT SPECIFICATION

The requirement specification for CleanWave outlines the functional and non-functional requirements essential for the development of the laundry management system.Functional requirements include features such as user registration and authentication, which allow customers to create and manage their accounts securely. The system must support an intuitive interface for browsing laundry services, placing orders, and processing payments. Additionally, administrators require functionalities for managing service offerings, tracking orders, and collecting customer feedback, which are vital for maintaining high service quality.

The system must also incorporate a unique donation module, enabling users to easily donate old clothes to those in need. Non-functional requirements focus on performance, scalability, security, and usability. The system should be designed for quick response times to ensure a smooth user experience, even during peak usage periods. Security measures must be implemented to protect sensitive user information and comply with data privacy regulations. Furthermore, the interface should be user-friendly, accessible across various devices, and visually appealing to enhance user engagement. This comprehensive requirement specification serves as a foundation for the design and development of CleanWave, ensuring that the system meets the needs and expectations of its users while maintaining high standards of performance and security.

2.2.1 SOFTWARE REQUIREMENT AND SPECIFICATION

An SRS (Software Requirements Specification) for CleanWave serves as a comprehensive written understanding of the system requirements and dependencies from the perspective of both the customer and the development team at a specific point in time, typically prior to any actual design or development work. This document acts as a two-way insurance policy, ensuring that both the client and the organization have a mutual understanding of the requirements and expectations for the laundry management system.

The software requirements for CleanWave focus on recording essential details related to laundry services, user accounts, order management, and customer feedback. This includes managing service offerings, tracking orders received, and documenting user interactions to streamline operations. The system is designed to minimize the duplication of entries made by administrators, thereby enhancing efficiency and accuracy in data management. By clearly outlining these requirements, CleanWave aims to provide a structured approach to developing a reliable and user-friendly laundry management platform that meets the needs of both customers and administrators.

The SRS document itself states in precise and explicit language those functions and capabilities a software system must provide, as well as states any required constraints by which the system must abide. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as “parent” document because all subsequent project management documents, such as design specifications, statement of works, software architecture specifications, testing and validation plans, and documentation plans , are related to it.

It’s important to note that an SRS contains functional and non-functional requirements only; it doesn’t offer design suggestion, possible solutions to technology or business issues, or any other information other than what the development team understands the customer system requirement to be.

A well designed, well written SRS accomplishes four major goals:

1. It provides test details to the customer. An SRS is the customer’s assurance that the development organization understands the issues or problems to be solved and the software behavior necessary to address problems. Therefore, the SRS should be written in natural language in an unambiguous manner that may also include charts, tables, data flow diagrams, and decision tables and so on.
2. It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.

3. It serves as an input to the design specification. As mentioned previously, the SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirement so that a design solution can be devised.
4. It serves as a product validation check. The SRS also serve as the parent document for testing and validation strategies that will be applied to the requirements for verification.

2.3 HARDWARE AND SOFTWARE SELECTION AND JUSTIFICATION

HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and capacity requirements are also important. Below is some of the hardware that is required by the system:

Processor – i3 or above
Speed – 1.1 GHz
RAM – 4GB (min)
Input Devices – Mouse, Keyboard
Output Devices – Monitor, Printer

SOFTWARE SELECTION AND JUSTIFICATION

We require much different software to make the application which is in making to work efficiently. It is very important to select the appropriate software so that the software works properly.

Below are the software requirements.

Operating System : Windows 11 or more
Front End : PHP
Back End : MYSQL
Web Technologies: HTML,JavaScript,CSS
Web Browser :Mozilla Firefox/Google Chrome/Edge
Web Server : Apache.2

WINDOWS 11

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released on October 5, 2021. It succeeded Windows 10 (2015) and is available for free for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the canceled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiledsets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S suchas Auto HDR and Direct Storage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser, like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via the Microsoft Store and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).

MY SQL

MySQL was created by a Swedish company, MySQL AB, founded by David Axmark, Allan Larssonand Michael "Monty" Widenius. Original development of MySQL by Widenius and Axmark began in 1994. The first version of MySQL appeared on 23 May 1995. It was initially created for personal usage from MySQL based on the low-level language ISAM, which the creators considered too slow and inflexible. They created a new SQL interface, while keeping the same API as MySQL. By keeping the API consistent with the MySQL system, many developers were able to use MySQL instead of the (proprietarily licensed) MySQL antecedent.

MySQL can be built and installed manually from source code, but it is more commonly installed from a binary package unless special customization are required. On most Linux distributions, the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings. Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerfulhardware, such as a multi-processor server with gigabytes of memory.

There are, however, limits to how far performance can scale on a single server, so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can include a powerful master database which handles data write operations and is replicated to multiple slaves that handle all read operations.[82] The master server continually pushes binlog events to connected slaves so in the event of failure a slave can be promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using Memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

PHP

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. HP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, and modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity
- All of the PHP present in the Web page is processed and stripped from the page; the only thing returned to the client from the Web server is pure HTML output.
- All PHP code must be included inside one of the three special markup tags ATE are recognized by the PHP Parser.

In order to develop and run PHP Web pages three vital components need to be installed on your computer system.

- **Web Server** − PHP will work with virtually all Web Server software, including Microsoft's Internet Information Server (IIS) but then most often used is freely available Apache Server. Download Apache for free here − https://httpd.apache.org/download.cgi
- **Database** − PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database. Download MySQL for free here − https://www.mysql.com/downloads/
- **PHP Parser** − In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser. This tutorial will guide you how to install PHP parser on your computer.

2.4 USE CASE DIAGRAM

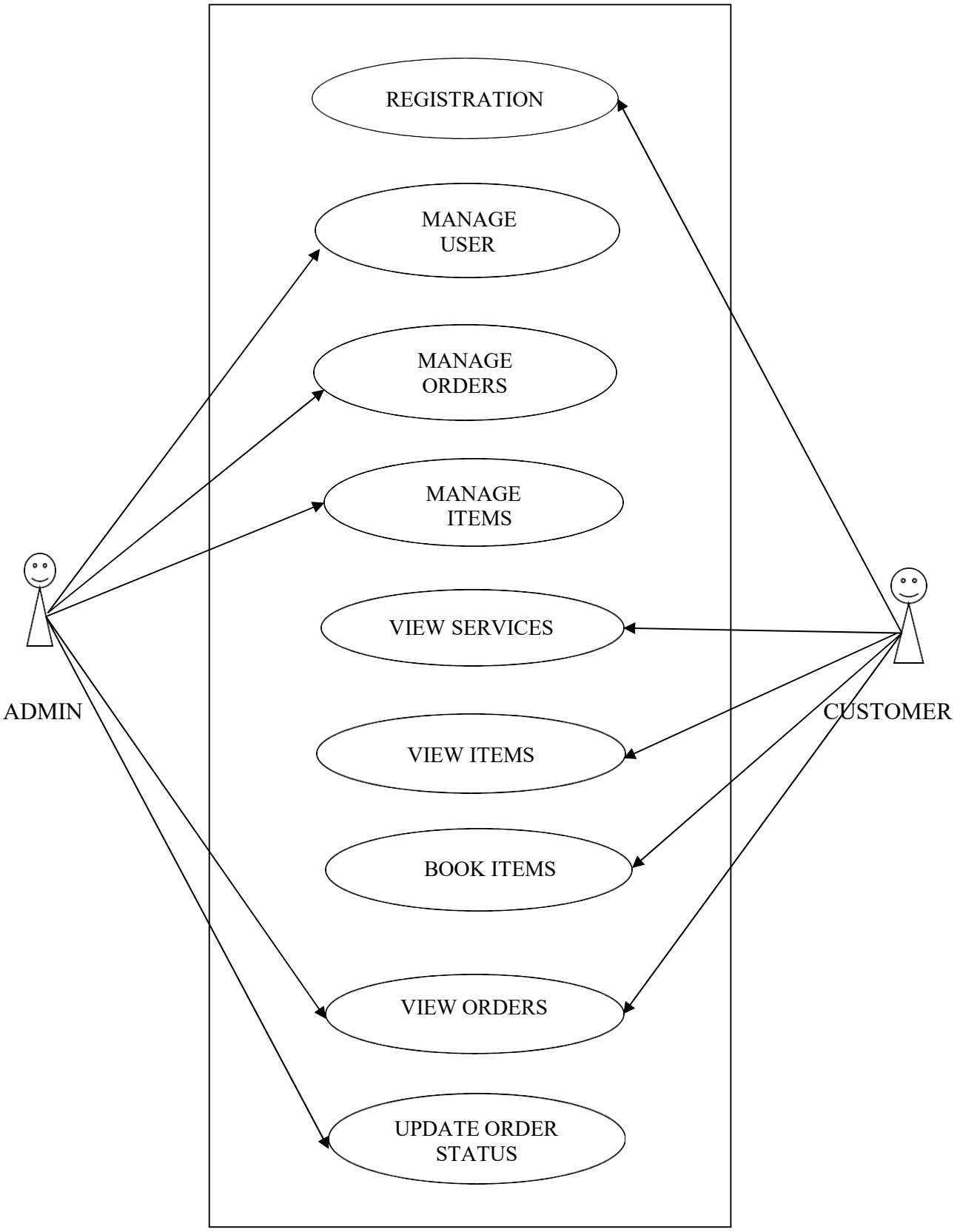


Fig 2.1: Use Case Diagram

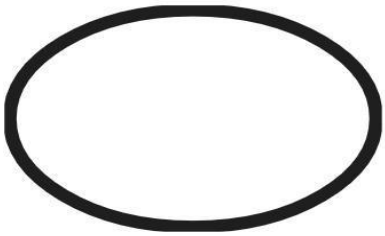
2.5 DATAFLOW DIAGRAM

A DFD also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations. It shows the flow of data through a system. It is a graphical tool because it presents a picture. The DFD may be partitioned into levels that represent increasing information flow and functional detail. Four simple notations are used to complete a DFD. These notations are given below:

DATA FLOW

A directed arc or an arrow is used as a Data Flow Symbol. This represents the data flow occurring between two processes or between an external entity and a process; in direction of the Data Flow Arrow. Data flow Symbols are annotated with corresponding data names.

PROCESS



A function is represented using a circle. This symbol is called a process or a bubble. Bubbles are annotated with the names of corresponding functions.

EXTERNAL ENTITY

An external entity such as a user, project manager etc. is represented by a rectangle. The external entities are essentially those physical entities external to the application system, which interact with the system by inputting data to the system or by consuming the data produced by the system. In addition to the human users the external entity symbols can be used to represent external hardware and software such as application software.

DATA STORE

A Data Store represents a logical file; it is represented using two parallel lines. A logical file can represent either Data Store Symbol, which can represent either data structure or a physical file on disk. Each data store is connected to a process by means of a Data Flow Symbol. The direction of the Data Flow Arrow shows whether data is being read from or written into a Data Store. An arrow flowing in or out of a data store implicitly represents the entire area of the Data Store and hence arrows connecting to a data store need not be annotated with the names of the corresponding data items.

2.5.1 CONTEXT LEVEL DIAGRAM

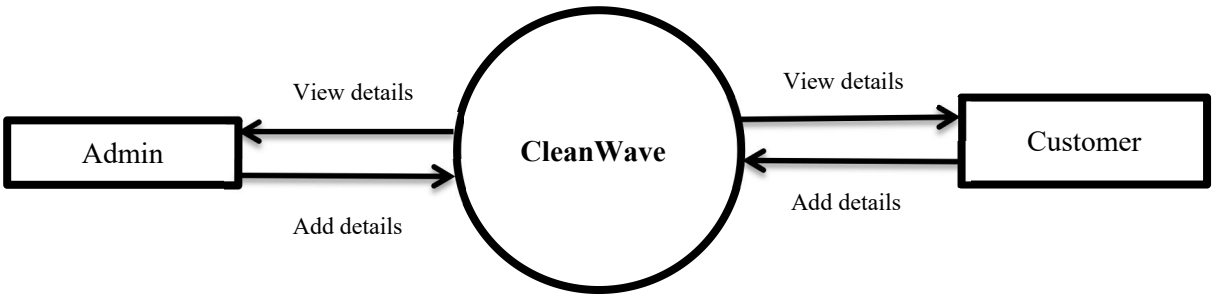


Fig 2.2: Zero Level Diagram

2.5.1 FIRST LEVEL DATA FLOW DIAGRAM

LEVEL 1 ADMIN

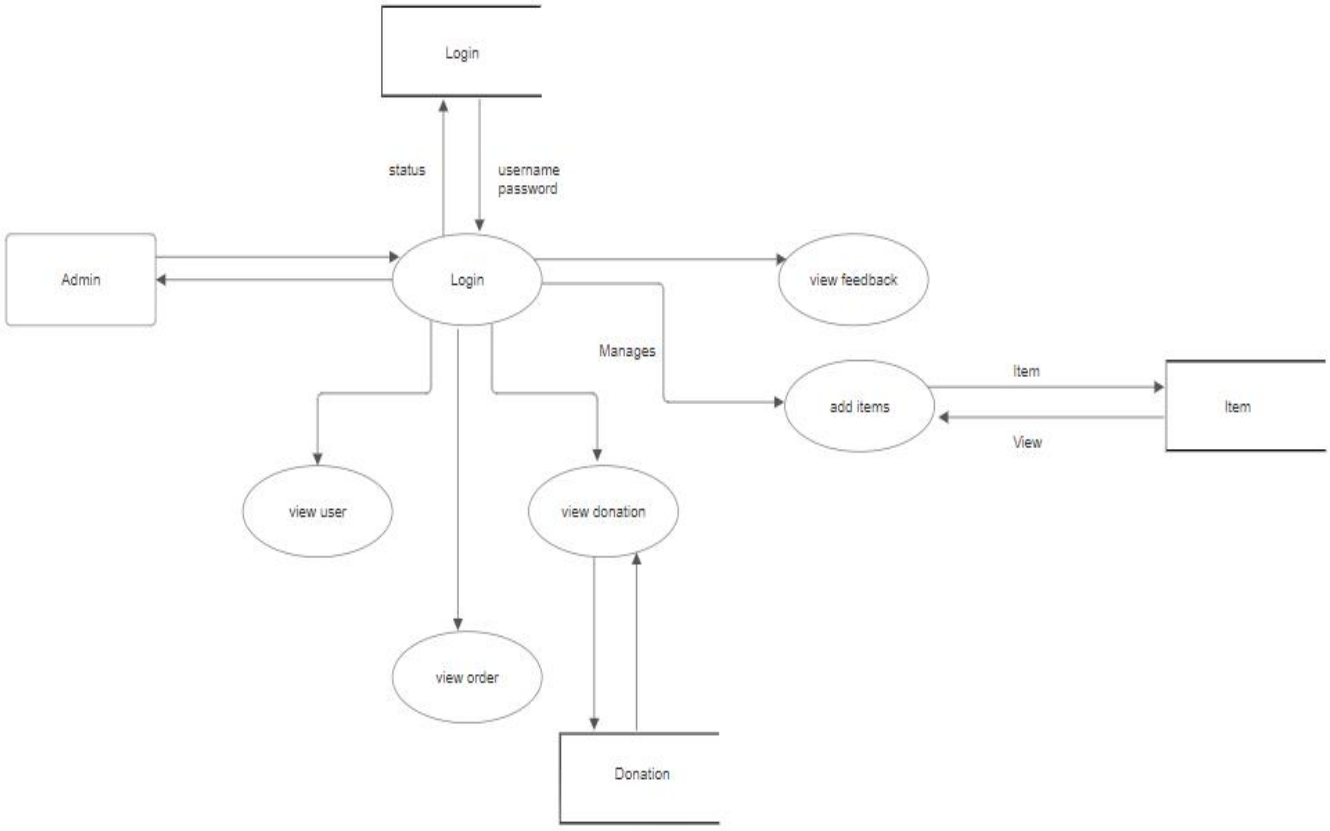


Fig 2.3: First Level DFD(Admin)

LEVEL 1 USER

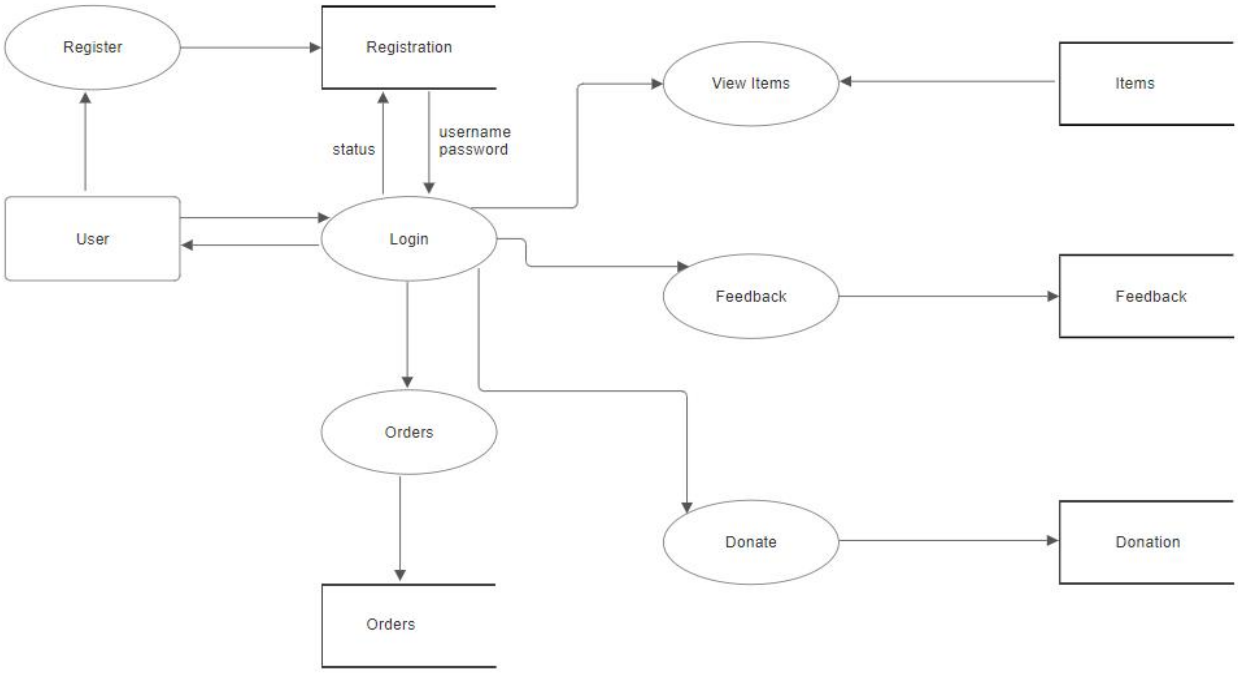
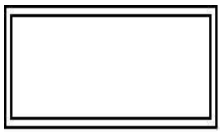


Fig 2.4: First Level DFD(User)

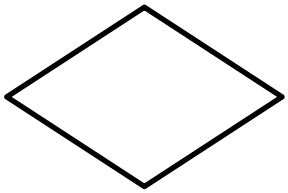
2.6 ENTITY RELATIONSHIP DIAGRAM

The ER model is a conceptual data model that views the real world as a construct of entities and associations or relationships between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects. The ER modelling technique is frequently used for the conceptual design of database applications and many database applications, and many database design tools employ its concepts.

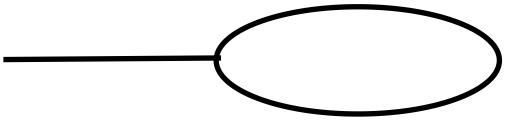
Entity Type



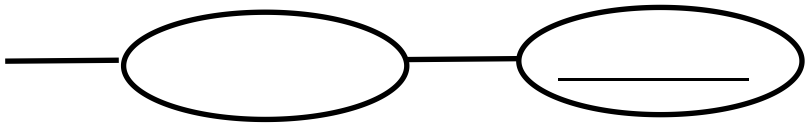
Weak Entity Type



Relationship Type



Attribute



Key Attribute

Multi-valued Attribute

2.6 ER DIAGRAM

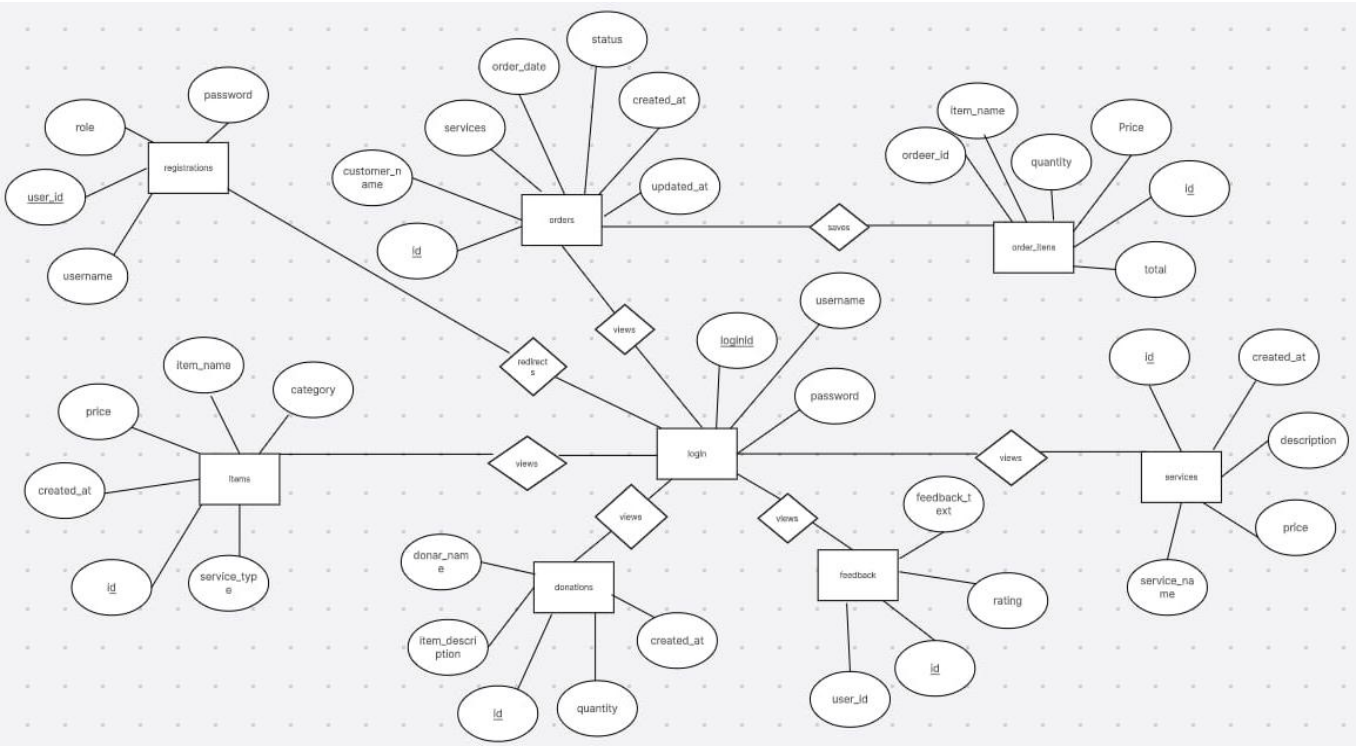


Fig 2.6: ER Diagram

3. SYSTEM DESIGN

System design focuses on the detailed implementation of the feasible system. Its emphasis on translating design specifications to performance specification. System design has two phases of development logical and physical design. During logical design phase the analyst describes inputs (sources), outputs(destinations), databases (data sores) and procedures (data flows) all in a format that meets the uses requirements. The analyst also specifies the user needs and at a level that virtually determines the information flow into and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which tell the programmers exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data through call and produce the required report on a hard copy or display it on the screen.

3.1 STRUCTED DESIGN METHODOLOGIES

Design methodology refers to the development of a system or method for a unique situation. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and requirements of the end user is the most critical concern. To employ design methodology various analyses and testing have been done so as to meet the desired user needs. Every input that the user input is being tested in this software. That means the validity of each data is being checked and if found invalid necessary warning and prompting messages are displayed. The output forms are also tested in detail to see whether the desired output is met or not. Also, the output forms are made clearer and more meaningful for the user to understand.

3.2 USER INTERFACE DESIGN

User interface design is one of the major functions in developing a system. It is a good understanding of the user needs very clearly. Because the user is the person who has to interact with system being developed. So that it should seek the needs of the user before developing it. The system is designed in a very user-friendly manner that makes the user with little knowledge of computer and of the organization can work very easily with the system.

The input forms that are used to enter the data are made clearer and easier to understand. Every time the user enters data the system is designed to check the validity of the data and if found as invalid

meaningful prompting and warning messages are displayed. This makes the user comfortable to interact with the system.

Also, when a user login to the system it checks the username and password entered to see whether it is valid user or not. It ensures security of the system and database. The data storage and data processing are made more efficient so that accurate results are being displayed on the output forms. And also, the retrieval of specific records as demanded by the user is made very faster that saves the user time.

Main Input Forms are as follows: -

1. **Login Form**: This form allows users to securely log into the CleanWave platform by entering their username and password.
2. **Service Details Form**: Admins can input and manage details of various laundry services offered, including descriptions, pricing, and any customization options available.
3. **Stock Management Form**: This form enables admins to record and track the inventory of laundry items and materials, ensuring efficient stock management and availability.
4. **Order Placement Form**: Customers use this form to place orders for laundry services, providing details such as service type, item quantity, and any specific instructions.
5. **Feedback Submission Form**: Customers can submit feedback or reviews regarding their experiences with the laundry services, helping to improve service quality.
6. **Donation Form**: This form allows users to donate old clothes easily, detailing the items they wish to contribute to those in need.

3.3 OUTPUT DESIGN

Output design is used to provide outputs to the users of the system. Computer output is the most important direct source of information to the user. Efficient intelligible output design improves the system relationships with the user and help in decision making major form of the output is the hardcopy from the printer and the screen reports. The output devices to consider depend on factors such as compatibility of the devices with the system, expected print quality and number of copies needed.

Here, in this case, I make use of forms which contains the tables to show the outputs of the processed data. The output design has been done so that the results of processing should be communicated to the user. Effective output design will improve the clarity and performance of outputs. Output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Output design phase of the system is concerned with the Convergence of information to the end user in a friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making.

The major output forms are as follows:

1. **View Products/materials:** Admin recorded categorized servicing items are displayed to users forinformation. Users can search the items they are looking for.
2. **Ordering:** The customer can order the item .
3. **Order Status:** Customer can view the status of the product ordered.

3.4 DATABASE DESIGN

A database is a collection of interrelated data stored within minimum redundancy to serve many users quickly and efficiently. It is a process of designing the database file, which is the key source of the information in the system. The objective of database is to design is to provide storage and it contributes to the overall efficiency of the system. The file should properly design and planned for collection, accumulation, editing and retrieving the required information.

3.4.1 DATA & INTEGRITY CONSTRAINTS

The primary objective of a database design are 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 languages. The theme behind a database is to handle information as an integrated whole thus the main objective is to make information as access easy, quick, inexpensive and flexible for the users. In this project, we mainly concentrated into relational databases.

Relational database stores data in tables, which is turn, are composed of rows also known as records, columns also known as fields. The fields in the relational model are:-

Primary Key

The key which is uniquely identify records. They also notify the not null constraints.

Foreign Key

The key which references the primary key, is the data inserted in the primary key column of the table.

Normalization

After the conceptual level, the next level of process of database design to organize the database structure into a good shape called Normalization. The normalization simplifies the entries, removing redundancies from the system data and finally builds a data structure, which is both flexible and adaptable to the system. The different normal forms obtained during the database design are given below:

In the database design, we create a database with different tables that is used to store the data. We normalize the data in the table. **Database normalization** is the process of organizing the fields and tables in a relational database to minimize redundancy and dependency. Normalization usually involves dividing large tables into smaller (and less redundant) tables and defining relationships between them. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

In the project I have made used of the 2nd normal form, Second Normal Form (2NF) is a property of database tables. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database Management System allows data to be protected and organized separately from other resources. Database is an integrated collection of data. This is the difference between logical and physical data.

In my project, I have made use of tables which are stored in the database named user. The tables are used to store the values that are generated by the application. The main field names and the key constraints of all the tables are shown above in detail.

3.5 TABLE DESIGN

1. Table Name: **orders**

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	Int	PRIMARY KEY	Order Id
customer_name	Varchar(255)	Not Null	Costumer name
services	Varchar(255)	Not Null	Service
order_date	date	Not Null	Date of order
status	Varchar(50)	Not Null	Status
created_at	timestamp	Null	
updated_at	timestamp	Null	

2. Table Name: order_items

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	Int	PRIMARY KEY	ID
order_id	Int	Not Null	User ID
item_name	Varchar(255)	Not Null	Item Name
Quantity	Int	Not Null	Quantity
price	Decimal(10,2)	Not Null	Price
total	Decimal(10,2)	Not Null	Total

3. Table Name: login

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
loginid	Int	PRIMARY KEY	UserId
username	Varchar(255)	Not Null	User name
password	Varchar(255)	Not Null	User Password

4.Table Name: items

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	int	PRIMARY KEY	id
item_name	Varchar(100)	Not Null	Item name
price	Decimal(10,2)	Not Null	Price
service_type	enum('wash+fold','wash+iron','dry clean','steam iron','stain clear')	Not Null	Service types
category	enum('men','women','kid','household')	Not Null	category
created_at	timestamp	Not Null	Created at

5.Table Name: services

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	Int	PRIMARY KEY	Id
service_name	Varchar(100)	Not Null	Service name
price	Decimal(10,2)	Not Null	Price
description	text	Not Null	description
created_at	timestamp	Not Null	Created at

6.Table Name: donations

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	Int	PRIMARY KEY	Id
donar_name	Varchar(100)	Not Null	Donar name
Item_description	Text	Not Null	Item description
quantity	Int	Not Null	Quantity
Created_at	timestamp	Not Null	Created at

7.Table Name: registeration

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
userid	Int	FOREIGN KEY	user Id
username	Varchar(255)	Not Null	User name
password	Varchar(255)	Not Null	Password
role	Varchar(100)	Not Null	Role

8.Table Name: feedback

FIELD NAME	DATATYPE	CONSTRAINTS	DESCRIPTION
id	Int	PRIMARY KEY	ID
user_id	Int	FOREIGN KEY	User ID
feedback_text	Text	Not Null	Feedback Text
rating	Int	Not Null	Rating
created_at	Timestamp	Not Null	Created at

4. CODING

4.1 PROGRAM CODE PREPARATION

When considered as a step-in software engineering, coding is viewed as a natural consequence of design. However, programming language characteristics and coding style can profoundly affect software quality and maintainability. The coding step translates a detail design representation into a programming language realization. The translation process continues when a compiler accepts source code as input and produces machine-independent object code as output. The initial translation step in detail design to programming language is a primary concern in the software engineering context. Improper interpretation of a detail design specification can lead to erroneous source code. Style is an important attribute of source code and can determine the intelligibility of a program. The elements of a style include internal documentation, methods for data declaration, procedures for statement construction, and I/O coding and declaration. In all cases, simplicity and clarity are key characteristics. An offshoot of coding style is the execution time and/or memory efficiency that is achieved. Coding is the phase in which we actually write programs using a programming language. In the coding phase, design must be translated into a machine in readable form. If design i s performed in a detailed manner, coding can be accomplished mechanistically. It was the only recognized development phase in early or unsystematic development processes, but it is just one of several phases in a waterfall process. The output of this phase is an implemented and tested collection of modules.

In my project I have made use of the PHP to code the whole project and have made use of the MySQL to act as a database to store the results of the processed data which is the output of the project.

5. IMPLEMENTATION OF SECURITY

The software quality assurance is comprised of a variety of tasks associated with seven major activities.

- Application of technical methods
- Conduct of formal technical reviews.
- Software testing.
- Enforcement of standards.
- Record keeping and reporting.

The quality begins with a set of technical methods and tools that help the analyst to achieve high quality specification and the designer to develop high quality design. The next activity involves assessment for quality for the design that is created which is the formal technical review. Software testing combines a multi-step strategy with a series of test case design methods that help ensure effective error detection.

5.1 DATA SECURITY

The software maintains a well-organized database for storing the details that are provided by the user. This helps us to eliminate the entering of invalid data. Data is not accessible to unauthorized users. The system analyst will provide the test data, specially designed to show that the system will operate successfully in all its aspects and produce expected results under expected conditions. Preparation of test data and the checking of results should be carried out in conjunction with the appropriate users and operational departments. Also the extent to which the system should be tested must be planned.

5.2 USER AND ACCESS RIGHTS

Admin:

Administrator or Admin is the super user and main controller of this system. Administrator controls all the activities of the Online Dress Store System.. He/ She can add the Category, brand and item. Admin can view the orders and can also update the status.

Customer:

Customer can perform various activities like registering into the account, can edit his/herdetails, viewing the items, and can order and make payment through online.

6. SYSTEM TESTING

Testing is a process of executing a program with intent of finding errors. In software development, errors can be injected at any stage of development. During testing, the program to be tested is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as expected.

There are mainly two approaches of testing namely, functional testing and structural testing. Functional testing is based on the functionality of the program and not the structure of the program. The test cases are solely on the basis of specification or requirements of the program. This type of testing is also called as black box testing.

Structural testing is also called as white box testing or glass box testing. Here the internal structure of the program is tested. Test cases are designed by examining the logic of the program.

"CleanWave" focuses on the functionality of the system and hence it mainly does the functional/black box testing. The test cases of this system are completely based on the specifications of the system.

The application was tested and found to be working as expected. There was no abnormal behavior reported during the testing of the program. Testing is a method by which we try reducing the testing efforts and bringing out the maximum output. Testing helps us in knowing whether the logical assumptions that we have taken for the system are correct, and if they are correct, we have obtained our goal. We test the system to know the errors, to check the validity of the information, to also group the modules with the aim that we meet the system requirements according to the system needs.

Testing is vital to the success of the system. System testing makes logical assumption that if all the parts of the system are correct, we have achieved the mission successfully. System testing is the stage of implementation that is aimed at assuring that the system works accurately and efficiently before the live operation commences.

Testing includes several levels of testing. They are:

1. Unit testing
2. Integration testing
3. System testing
4. Acceptance testing

6.1 UNIT TESTING

Unit testing is the first level of testing. In this process the code produced during the coding phase is verified. The goal is to test the internal logic of the modules. This is also known as "module testing". The modules are tested separately; this testing is carried out during programming stage itself.

In this testing step each module such as brand, items, customer, orders are found to be working satisfactorily as regard to the expected output from the module. Each and every form in the system is tested independently before integrating.

6.2 INTEGRATION TESTING

Integration testing is performed to check the correctness of the interface between the modules. The goal is to see if the modules can be integrated properly. In this project all the modules are combined and then entire program is tested as whole. The forms that need to connected like brand, category is connected to item form for categorization, similarly other forms are integrated to work as single unit. Thus, in the integration step, all the errors uncovered for the next testing steps.

6.3 USER ACCEPTANCE TESTING

Acceptance testing is sometimes performed with realistic data of client to demonstrate that the software is working satisfactorily.

The "Online Dress Store" was tested and found to be working as expected. There was no abnormal behaviour reported during the testing of the program. Testing is a method by which we try reducing the testing efforts and bringing out the maximum output. Testing helps us in knowing whether the logical assumptions that we have taken for the system are correct, and if they are correct, we have obtained our goal. We test the system to know the errors, to check the validity of the information, to also group the modules with the aim that we meet the system requirements according to the system needs.

6.4 TEST CASE DESIGN

Function Tested	Test condition	Expected result	Actual result	Status
Login	Invalid username and password	Not allowed	Not allowed	Pass
Email	Entered Invalid email	Not allowed	Not allowed	Pass
Phone No	Eneterd invalid phone number	Not Allowed	Not Allowed	Pass
Name	Entered invalid characters	Not Allowed	Not Allowed	Pass
Empty fields	Submitted the form without entering data	Not allowed	Fields values Required	Pass

6.5 TEST REPORT AND DEBUGGING

Testing means verifying correct behaviour. Testing can be done at all stages of module development: requirements analysis, interface design, algorithm design, implementation, and integration with other modules. In the following, attention will be directed at implementation testing. Implementation testing is not restricted to execution testing. An implementation can also be tested using correctness proofs, code tracing, and peer reviews, as described below.

Debugging is a cyclic activity involving execution testing and code correction. The testing that is done during debugging has a different aim than final module testing. Final module testing aims to demonstrate correctness, whereas testing during debugging is primarily aimed at locating errors. This difference has a significant effect on the choice of testing strategies.

- **Report error conditions immediately** - Much debugging time is spent zeroing in on the cause of errors. The earlier an error is detected, the easier it is to find the cause. If an incorrect module state is detected as soon as it arises then the cause can often be determined with minimal effort. If it is not detected until the symptoms appear in the client interface then may be difficult to narrow down the list of possible causes.

- **Maximize useful information and ease of interpretation** - It is obvious that maximizing useful information is desirable, and that it should be easy to interpret. Ease of interpretation is important in data structures. Some module errors cannot easily be detected by adding code checks because they depend on the entire structure. Thus it is important to be able to display the structure in a form that can be easily scanned for correctness.

- **Minimize useless and distracting information** - Too much information can be as much of a handicap as too little. If you have to work with a printout that shows entry and exit from every procedure in a module then you will find it very difficult to find the first place where something went wrong. Ideally, module execution state reports should be issued only when an error has occurred. As a general rule, debugging information that says "the problem is here" should be preferred in favour of reports that say "the problem is not here".

- **Avoid complex one-use testing code** - One reason why it is counterproductive to add module correctness checks for errors that involve the entire structure is that the code to do so can be quite complex. It is very discouraging to spend several hours debugging a problem, only to find that the error was in the debugging code, not the module under test. Complex testing code is only practical if the difficult parts of the code are reusable.

7. SYSTEM IMPLEMENTATION AND MAINTENANCE

Implementation is an activity that is contained throughout the development phase. It is the process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successful system should be delivered and users should have the confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation. Implementation is the stage of the system when the theoretical design is turned into working system. The implementation involves careful planning investigation of the current system and its constraints on implementing, design of methods to achieve the changeover, training of user over procedure and evaluation change over method. There are three types of implementations:

1. Implementation of a computer system to replace a manual system. The problems involved are converting files, training users, creating accurate files, and verifying printouts for integrity.
2. Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned, there can be many problems. Some larger systems have taken as long as a year to convert.
3. Implementation of a modified application to replace an existing one using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in files.

7.1 SYSTEM MAINTENANCE

Maintenance corresponds to restoring something to original conditions, covering a wide range of activities including correcting codes and design errors and updating user support. Maintenance is performed most often to improve the existing software rather than to a crisis or risk failure. The system would fail if not properly maintained. The software maintenance is an important one in the software development because we have to spend more efforts for maintenance. Software maintenance is to improve the software quality according to the requirements. After a system is successfully implemented, it should be maintained in a proper manner. The need for system maintenance is to make the system adaptable to the changes in the system environment. There may be social, economic or technical changes, which affect system being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to cope with these changes. We may define maintenance by describing four activities that are undertaken to after a program is released for use.

The first maintenance activity occurs because it is unreasonable to assume that software testing will

uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer. The process that includes the diagnosis and correction of one or more errors is called corrective maintenance.

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspects of computing. Therefore, adaptive maintenance – an activity that modifies software to properly interface with a changing environment is both necessary and common place.

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancements are received from users. To satisfy requests in this category, perfective maintenance is performed. This activity accounts for the majority of all efforts expended on software maintenance.

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

8. SCOPE OF THE PROJECT

CleanWave is a comprehensive web application designed to streamline the management of laundry services and promote sustainable practices through a dedicated donation module. The project aims to offer an intuitive and efficient platform for users to manage their laundry needs while encouraging community involvement through clothing donations.

The primary objective of CleanWave is to create a user-friendly web application that simplifies the laundry process for customers. Users will be able to register, place orders for various laundry services, track their orders in real-time, and manage their account information with ease. The application will support different types of laundry services, including dry cleaning, wash and fold, and ironing, and will provide users with a straightforward interface for scheduling pickups and deliveries.

A key feature of CleanWave is its donation module, which allows users to donate their old clothes to those in need. This module is designed to integrate seamlessly with local charities or NGOs, facilitating the donation process and scheduling pickups for donated items. Users who contribute to the donation program will receive discounts on future laundry services, promoting both community support and sustainability.

Administrative users will have access to a dedicated dashboard for managing users, orders, and donations. This dashboard will provide insights into application activity, order trends, and user engagement through reporting and analytics tools. Administrators will also manage discounts, promotional offers, and inventory, ensuring smooth operation of the platform.

In terms of technology, CleanWave will use modern web technologies for both the frontend and backend. The frontend will be developed using HTML, CSS, and JavaScript, with frameworks like React or Angular to build a responsive and dynamic user interface.

The scope of the project excludes the development of native mobile apps in the initial phase and does not include complex AI-based features or integration with unrelated services. Future enhancements may include mobile app development, advanced AI features, and expanded integration with local charities.

This scope outlines the core functionalities and goals of CleanWave, providing a clear understanding of the project's objectives and deliverables.

9. FUTURE ENHANCEMENT

1. Mobile Application Development: Creating a mobile app for CleanWave would enhance accessibility, allowing customers to place orders, track their laundry, and manage their accounts conveniently from their smartphones.
2. Real-time Order Tracking: Implementing a real-time order tracking system would enable customers to receive live updates on their laundry status, including pickup and delivery times.
3. Customer Loyalty Program: Introducing a loyalty program could incentivize repeat business by offering discounts, rewards, or exclusive promotions for frequent customers.\
4. Advanced Search and Filter Options: Enhancing the search functionality to include advanced filtering options (e.g., by service type, price range, and user ratings) would improve the customer experience by making it easier to find specific services.
5. Integration of AI Chatbot: Incorporating an AI-driven chatbot on the platform could assist customers with inquiries, provide instant support, and guide them through the ordering process, enhancing user engagement.
6. Payment Gateway Integration: Adding multiple payment options, including digital wallets and cryptocurrencies, would provide customers with greater flexibility and convenience in completing transactions.
7. Reporting and Analytics Dashboard: Developing a comprehensive analytics dashboard for admins to visualize key metrics and trends in service usage, customer behavior, and financial performance could facilitate better decision-making.

8. Sustainability Features: Introducing features that promote eco-friendly practices, such as options for customers to choose sustainable laundry processes, could align with growing consumer preferences for environmentally responsible services.
9. Multilingual Support: Adding multilingual support to the platform would cater to a broader audience, making the service more accessible to non-English speaking customers.

These future enhancements aim to improve the overall functionality and user experience of CleanWave, ensuring that the platform remains competitive and aligned with customer needs in the evolving laundry service industry.

10. CONCLUSION

In conclusion, the CleanWave project represents a significant advancement in the management of laundry services and community support through its innovative web application. By combining efficient laundry management with a sustainable donation module, CleanWave addresses key needs of both users and service providers, fostering a more connected and responsible community.

Advantages of CleanWave:

1. **Enhanced User Experience:**Provides a streamlined and intuitive platform for managing laundry services, making it easier for users to place orders, track progress, and manage their accounts.
2. **Community Impact:**The integrated donation module promotes sustainability by encouraging users to donate old clothes, benefiting those in need and supporting local charities.
3. **Comprehensive Admin Tools:**Equips administrators with a robust dashboard for managing users, orders, and donations, along with reporting and analytics tools to gain valuable insights.
4. **Promotional Incentives:**Offers discounts and rewards for donations, incentivizing users to participate in the donation program and fostering customer loyalty.
5. **Future-Proof Design:**The platform is designed with scalability in mind, allowing for future enhancements such as mobile apps and advanced features to further enhance user engagement and service offerings.

CleanWave stands out as a forward-thinking solution that not only meets the practical needs of laundry management but also encourages positive social impact through its donation features. By leveraging modern technologies and focusing on user-centric design, CleanWave is well-positioned to deliver significant value to its users and the community alike.

11. BIBLIOGRAPHY

Elias. M. Awad, "System *Analysis and* Design", Galgoeia publications – Second Edition2005.

Henry Korth," Data *Base Design Concept*", Mc grew hill – Fifth Edition 2001.

Roger.S.Pressman," Software *Engineering* ",Mc grew hill Fifth Edition 2006.

www.w3schools.com

www.htmlcolorcodes.com

Appendix 1 CODING

Index.php

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>CleanWave|Assna VS</title>
  <!-- CSS Link -->
  <link rel="stylesheet" href="style.css" />
  <!-- Box Icon Link for Icons -->
  <link
   href="https://unpkg.com/boxicons@2.1.4/css/boxicons.min.css"
   rel="stylesheet"
  />
 </head>
 <body>
  <!-- Header & Navbar Section -->
  <header>
   <nav>
    <div class="nav_logo">
     <a href="#">
      <img src="images/CleanWave_logo.png" alt="Coffee Logo" />
      <h2>CleanWave</h2>
     </a>
    </div>
    <input type="checkbox" id="click" />
    <label for="click">
     <i class="menu_btn bx bx-menu"></i>
     <i class="close_btn bx bx-x"></i>
    </label>
    <ul>
     <li><a href="#">Home</a></li>
     <li><a href="#about">About</a></li>
     <li><a href="#service">Services</a></li>
     <li><a href="#why">Why Us</a></li>
     <li><a href="#gallery">Gallery</a></li>
     <li><a href="#contact">Contact</a></li>
     <li><a href="#" class="login-btn" onclick="showLoginForm()">Login</a></li>
     <li><a href="#" class="signup-btn" onclick="showSignupForm()">Sign Up</a></li>
    </ul>
   </nav>
  </header>
```

```
    <!-- Hero Section -->
    <!-- Login Form -->
<div class="login-form" id="login-form" style="display: none;">
  <form action="login.php" method="POST">
    <h1>LogIn</h1>
    <br>
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" placeholder=" email"  required><br><br>
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required><br><br>
    <input type="submit" value="login" id="login" name="login">
  </form>

</div>

  <!-- Signup Form -->
<div class="signup-form" id="signup-form" style="display: none;">
  <form action="signup.php" method="POST">
    <h1>SignUp</h1>
    <label for="signup-username">Username:</label>
    <input type="email" id="signup-username" name="username" placeholder="enter email" required>
    <label for="signup-password">Password:</label>
    <input type="password" id="signup-password" name="password" required>
    <label for="signup-cpassword">Confirm Password:</label>
    <input type="password" id="signup-cpassword" name="cpassword" required>
    <input type="checkbox" id="signup-checkbox">
    <label for="signup-checkbox">By continuing, I agree to the Terms and Conditions and Privacy
Policy of CleanWave.</label>
    <input type="submit" value="signup" id="signup" name="signup">
  </form>
</div>

<script>
  function showLoginForm() {
    document.getElementById("login-form").style.display = "block";
    document.getElementById("signup-form").style.display = "none";
  }

  function showSignupForm() {
    document.getElementById("signup-form").style.display = "block";
    document.getElementById("login-form").style.display = "none";
  }
</script>

<script>
   // Function to validate an email address
  function validateEmail(email) {
    const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;
    return emailRegex.test(email);
  }
```

```
  // Get the signup form element
  const signupForm = document.getElementById('signup-form');

  // Add an event listener to the form's submit event
  signupForm.addEventListener('submit', (e) => {
 // Get the input values
    const username = document.getElementById('signup-username').value;
    const password = document.getElementById('signup-password').value;
    const cpassword = document.getElementById('signup-cpassword').value;
    const checkbox = document.getElementById('signup-checkbox').checked;

    // Validate the input values
    if (username === '' || password === '' || cpassword === '') {
      alert('Please fill in all fields');
      e.preventDefault(); // Prevent form submission if validation fails
      return;
    }

    // Check if the username is a valid email address
    if (!validateEmail(username)) {
      alert('Invalid email address');
      e.preventDefault(); // Prevent form submission if validation fails
      return;
    }

    // Check if the password and confirm password match
    if (password !== cpassword) {
      alert('Passwords do not match');
      e.preventDefault(); // Prevent form submission if validation fails
      return;
    }

    // Check if the terms and conditions checkbox is checked
    if (!checkbox) {
      alert('You must agree to the Terms and Conditions');
      e.preventDefault(); // Prevent form submission if validation fails
      return;
    }

    // If all validations pass, do not prevent form submission
    // The form will submit normally and redirect to the specified URL
  });
  // Function to show login form
  function showLoginForm() {
    document.getElementById("login-form").style.display = "block";
    document.getElementById("signup-form").style.display = "none";
  }

  // Function to show signup form
  function showSignupForm() {
    document.getElementById("signup-form").style.display = "block";
    document.getElementById("login-form").style.display = "none";
  }
</script>
```

```
<section class="hero_section">
  <div class="section_container">
    <div class="hero_container">
      <div class="text_section">
        <h2>Welcome to CleanWave</h2>
        <h3>- Your Trusted Laundry Partner </h3>
        <p>
         Experience the Best in Laundry Care and Community Support
        </p>
        <div class="hero_section_button">

  <a href="service.php" class="button">Book Slot</a>
         <a href="admin/add_feedback.php" class="button">Feedback</a>
        </div>
         </div>

      <div class="image_section">
        <img src="images/CleanWave_logo.png" alt="Coffee" />
      </div>
    </div>
  </div>
</section>
<!-- About Us Section -->
<section class="about_us" id="about">
  <div class="section_container">
    <div class="about_container">
      <div class="text_section">
        <h2 class="section_title">About Us</h2>
        <p>
          Welcome to CleanWave, your trusted partner in laundry and clothes donation services. At CleanWave, we are dedicated to providing a seamless and efficient laundry experience, ensuring your clothes are treated with the utmost care. Our user-friendly platform allows you to easily book slots for laundry cleaning and donate used or old clothes to those in need. We believe in making a positive impact on our community by promoting cleanliness and sustainability. Join us in our mission to create a cleaner, more compassionate world, one load of laundry at a time.
        </p>
      </div>
      <div class="image_section">
        <img src="images/w1.jpg" alt="coffee" />
      </div>
    </div>
  </div>
</section>
<!-- Services Section -->
<section class="services" id="service">
  <h2 class="section_title">Our Services</h2>
  <div class="section_container">
    <div class="service_container">
      <div class="services_items">
        <img src="images/c1.png" alt="Hot Beverages" />
        <div class="services_text">
          <h3>Wash + Fold</h3>
```

```
          <p>
            Enjoy freshly washed and neatly folded clothes, ready to wear or store.
          </p>
        </div>
      </div>
      <div class="services_items">
        <img src="images/c2.png" alt="Cold Beverages" />
        <div class="services_text">
          <h3>Wash + Iron</h3>
          <p>
            Get your clothes washed and perfectly ironed for a crisp, professional look.
          </p>
        </div>
      </div>
      <div class="services_items">
        <img src="images/c3.png" alt="Refreshment" />
        <div class="services_text">
          <h3>Dry Clean</h3>
          <p>Specialized cleaning for delicate fabrics and garments, ensuring they look their best.</p>
        </div>
      </div>
      <div class="services_items">
        <img src="images/c4.png" alt="Special Combos" />
        <div class="services_text">
          <h3>Steam Iron</h3>
          <p>
            Professional steam ironing to remove wrinkles and refresh your clothes.
          </p>
        </div>
      </div>
      <div class="services_items">
        <img
          src="images/c5.png"
          alt="Burger & French Fries"
        />
        <div class="services_text">
          <h3>Stain Clear</h3>
          <p>Effective treatment to remove tough stains and restore your clothes to their original
condition.</p>
        </div>
      </div>
      <div class="services_items">
        <img src="images/c6.png" alt="Desserts" />
        <div class="services_text">
          <h3>DONATION!</h3>
          <p>
            Donate your used or old clothes to those in need, making a positive impact on the community.
          </p>
        </div>
      </div>
    </div>
  </div>
</section>
```

```
<!-- Why Us Section -->
<section class="why_us" id="why">
  <h2 class="section_title">Why Us?</h2>
  <div class="section_container">
    <div class="why_container">
      <div class="why_items">
        <img src="images/s1.png" alt="Delicious" />
        <div class="why_us_text">
          <h3>Quality Service</h3>
          <p>
            We ensure your clothes are treated with the utmost care, using high-quality detergents and
advanced cleaning techniques.
          </p>
        </div>
      </div>
      <div class="why_items">
        <img src="images/s2.png" alt="Pleasant" />
        <div class="why_us_text">
          <h3>Affordable Prices</h3>
          <p>
            Enjoy top-notch laundry services at competitive prices.
          </p>
        </div>
      </div>
      <div class="why_items">
        <img src="images/s3.png" alt="Hospitality" />
        <div class="why_us_text">
          <h3>Customer Satisfaction</h3>
          <p>
            We prioritize your satisfaction and strive to exceed your expectations with every service.
          </p>
        </div>
      </div>
    </div>
  </div>
</section>
<!-- Gallery Section -->
<section class="gallery" id="gallery">
  <h2 class="section_title">Gallery</h2>
  <div class="section_container">
    <div class="gallery_container">
      <div class="gallery_items">
        <img src="images/g1.jpg" alt="Gallery Image" />
      </div>
      <div class="gallery_items">
        <img src="images/g2.jpg" alt="Gallery Image" />
      </div>
      <div class="gallery_items">
        <img src="images/g3.jpg" alt="Gallery Image" />
      </div>
      <div class="gallery_items">
        <img src="images/g4.jpg" alt="Gallery Image" />
      </div>
```

```
      <div class="gallery_items">
        <img src="images/g5.jpg" alt="Gallery Image" />
      </div>
      <div class="gallery_items">
        <img src="images/g6.jpg" alt="Gallery Image" />
      </div>
    </div>
  </div>
</section>
<!-- Contact Section -->
<section class="contact" id="contact">
  <h2 class="section_title">Contact Us</h2>
  <div class="section_container">
    <div class="contact_container">
      <div class="contact_form">
        <form action="#">
          <div class="field">
            <label for="Name">Full Name</label>
            <input type="text" id="Name" placeholder="Your Name" required />
          </div>
          <div class="field">
            <label for="email">Your Email</label>
            <input
              type="text"
              id="email"
              placeholder="Your Email"
              required />
          </div>
          <div class="field">
            <label for="number">Your Number</label>
            <input
              type="number"
              id="number"
              placeholder="Your Contact Number"
              required
            />
          </div>
          <div class="field">
            <label for="textarea">Message</label>
            <textarea
              name="textarea"
              id="textarea"
              placeholder="Your Message"
            ></textarea>
          </div>
          <button class="button" id="submit">Submit</button>
        </form>
      </div>
      <div class="contact_text">
        <div class="contact_items">
          <i class="bx bx-current-location"></i>
          <div class="contact_details">
            <h3>Our Location</h3>
```

```
          <p>Kerela, India</p>
          </div>
        </div>
        <div class="contact_items">
          <i class="bx bxs-envelope"></i>
          <div class="contact_details">
            <h3>General Enquries</h3>
            <p>cleanwave@xyz.com</p>
          </div>
        </div>
        <div class="contact_items">
          <i class="bx bxs-phone-call"></i>
          <div class="contact_details">
            <h3>Call Us</h3>
            <p>+91 9076459817</p>
          </div>
        </div>
        <div class="contact_items">
          <i class="bx bxs-time-five"></i>
          <div class="contact_details">
            <h3>Our Timing</h3>
            <p>Mon-Sat : 10:00 AM - 7:00 PM</p>
          </div>
        </div>
      </div>
    </div>
  </div>
 </section>
 <!-- Footer Section -->
 <footer>
  <div class="section_container">
   <div class="footer_section">
    <div class="footer_logo">
     <a href="index.html">
      <img src="images/CleanWave_logo.png" alt="Coffee Logo" />
<h2>CleanWave</h2>
     </a>
    </div>
    <div class="useful_links">
     <h3>Useful Links</h3>
     <ul>
      <li><a href="#about">About</a></li>
      <li><a href="#service">Services</a></li>
      <li><a href="#why">Why Us</a></li>
      <li><a href="#gallery">Gallery</a></li>
      <li><a href="#contact">Contact</a></li>
     </ul>
    </div>
    <div class="contact_us">
     <h3>Contact</h3>
     <ul>
      <li>
```

```
            <i class="bx bx-current-location"></i>
            <span>Kerela, India</span>
          </li>
          <li>
            <i class="bx bxs-phone-call"></i> <span>+91 9076459817</span>
          </li>
          <li>
            <i class="bx bxs-time-five"></i>
            <span>Mon-Sat : 10:00 AM - 7:00 PM</span>
          </li>
        </ul>
      </div>
      <div class="follow_us">
        <h3>Follow</h3>
        <i class="bx bxl-facebook-circle"></i>
        <i class="bx bxl-twitter"></i>
        <i class="bx bxl-instagram-alt"></i>
      </div>
    </div>
  </div>
 </footer>
</body>
</html>
```

Admin.php

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
   <meta charset="utf-8">
   <title>Admin Dashboard | CleanWave</title>
   <style>
      /* CSS code goes here */
      @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=swap
');
      *{
         margin: 0;
         padding: 0;
         box-sizing: border-box;
         font-family: 'Poppins', sans-serif;
      }

      /* ... rest of the CSS code ... */
      .wrapper{
           height: 100%;
           width: 300px;
           position: relative;
           background: #81007f; /* updated */

         }
```

```
.wrapper .menu-btn{
  position: absolute;
  left: 20px;
  top: 10px;
  background: #660066;
  color: #ffffff;
  height: 45px;
  width: 45px;
  z-index: 9999;
  border: 1px solid #333;
  border-radius: 5px;
  cursor: pointer;
  display: flex;
  align-items: center;
  justify-content: center;
  transition: all 0.3s ease;
}
#btn:checked ~ .menu-btn{
  left: 247px;
}
.wrapper .menu-btn i{
  position: absolute;
  transform: ;
  font-size: 23px;
  transition: all 0.3s ease;
}
.wrapper .menu-btn i.fa-times{
  opacity: 0;
}
#btn:checked ~ .menu-btn i.fa-times{
  opacity: 1;
  transform: rotate(-180deg);
}
#btn:checked ~ .menu-btn i.fa-bars{
  opacity: 0;
  transform: rotate(180deg);
}
#sidebar{
  position: fixed;
  background: #990099;
  height: 100%;
  width: 270px;
  overflow: hidden;
  left: -270px;
  transition: all 0.3s ease;
}
#btn:checked ~ #sidebar{
  left: 0;
}
```

```
#sidebar .title{
  line-height: 65px;
  text-align: center;
  background: #660066;
  font-size: 25px;
  font-weight: 600;
  color: #ffffff;
  border-bottom: 1px solid #333;
}
#sidebar .list-items{
  position: relative;
  background: #660066;
  width: 100%;
  height: 100%;
  list-style: none;
}
#sidebar .list-items li{
  padding-left: 40px;
  background: #81007f;
  line-height: 50px;
  border-top: 1px solid rgba(255,255,255,0.1);
  border-bottom: 1px solid #333;
  transition: all 0.3s ease;
}
#sidebar .list-items li:hover{
  border-top: 1px solid transparent;
  border-bottom: 1px solid transparent;
  box-shadow: 0 0px 10px 3px #222;
}
#sidebar .list-items li:first-child{
  border-top: none;
  background: #81007f;

}

#sidebar .list-items li a{
  color: #ffffff;
  background: #81007f;
  text-decoration: none;
  font-size: 18px;
  font-weight: 500;
  height: 100%;
  width: 100%;
  display: block;
}
#sidebar .list-items li a i{
  margin-right: 20px;
  background:#81007f;
}
```

```
        #sidebar .list-items .icons{
          width: 100%;
          height: 40px;
          text-align: center;
          position: absolute;
          bottom: 100px;
          line-height: 40px;
          display: flex;
          align-items: center;
          justify-content: center;
        }
        #sidebar .list-items .icons a{
          height: 100%;
          width: 40px;
          display: block;
          margin: 0 5px;
          font-size: 18px;
          color: #ffffff;
          background: #660066;
          border-radius: 5px;
          border: 1px solid #383838;
          transition: all 0.3s ease;
        }
        #sidebar .list-items .icons a:hover{
          background: #404040;
        }
        .list-items .icons a:first-child{
          margin-left: 0px;
        }
        .content{
          position: absolute;
          top: 50%;
          left: 50%;
          transform: translate(-50%,-50%);
          color: #202020;
          z-index: -1;
          width: 100%;
          text-align: center;
        }
        .content .header{
          font-size: 45px;
          font-weight: 700;
          color: #81007f;
        }
        .content p{
          font-size: 40px;
          font-weight: 700;
          color: #81007f;
        }
    </style>
</head>
```

```
<body>
  <!-- HTML code goes here -->
  <div class="wrapper">
    <!-- ... rest of the HTML code ... -->
    <input type="checkbox" id="btn" hidden>
   <label for="btn" class="menu-btn">
    <i class="fas fa-bars"></i>
    <i class="fas fa-times"></i>
   </label>
   <nav id="sidebar">
    <div class="title">Side Menu</div>
    <ul class="list-items">
     <li><a href="admin.php"><i class="fas fa-home"></i>Dashboard</a></li>
     <li><a href="manage_users"><i class="fas fa-sliders-h"></i>Manage User</a></li>
     <li><a href="manage_orders"><i class="fas fa-address-book"></i>Manage order</a></li>
     <li><a href="manage_services"><i class="fas fa-cog"></i>Manage Services</a></li>
     <li><a href="manage_items"><i class="fas fa-globe-asia"></i>Manage items</a></li>
     <li><a href="manage_donations"><i class="fas fa-stream"></i>Manage Donations</a></li>
     <li><a href="manage_feedback"><i class="fas fa-user"></i>Manage feedback</a></li>
     <!--<li><a href="#"><i class="fas fa-globe-asia"></i>Languages</a></li>
     <li><a href="#"><i class="fas fa-envelope"></i>Contact us</a></li>-->
     <!--<div class="icons">
      <a href="#"><i class="fab fa-facebook-f"></i></a>
      <a href="#"><i class="fab fa-twitter"></i></a>
      <a href="#"><i class="fab fa-github"></i></a>
      <a href="#"><i class="fab fa-youtube"></i></a>
     </div>-->
    </ul>
   </nav>
  </div>
  <div class="content">
   <div class="header">CleanWave </div>
   <p>admin dashboard</p>
  </div>
</body>
</html>
```

Service.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CleanWave - Services</title>
  <link rel="stylesheet" href="style.css" />
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f9f9f9;
    }
```

```
        .container {
            max-width: 1200px;
            margin: 0 auto;
            padding: 20px;
            display: flex;
            flex-wrap: wrap;
            justify-content: space-between;
        }
        .service-card {
            background-color: #fff;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            margin: 10px;
            padding: 20px;
            width: 30%;
            text-align: center;
        }
        .service-card img {
            width: 100%;
            height: auto;
            border-radius: 10px;
        }
        .service-title {
            font-size: 20px;
            margin: 10px 0;
        }
        .service-button {
            display: inline-block;
            padding: 10px 20px;
            background-color: #333;
            color: #fff;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 16px;
            transition: background-color 0.3s;
        }
        .service-button:hover {
            background-color: #FFC107;
        }
        /* Add a style for the heading */
        .services-heading {
            text-align: center;
            font-size: 32px;
            font-weight: bold;
            margin: 40px 0 20px;
            color: #333;
        }
    </style>
</head>
```

```
<body>
  <header>
    <nav>
      <div class="nav_logo">
        <a href="#">
          <img src="images/CleanWave_logo.png" alt="CleanWave Logo" />
          <h2>CleanWave</h2>
        </a>
      </div>
      <ul>
        <li><a href="index.php">Home</a></li>
        <!--<li><a href="index.php">About</a></li>
        <li><a href="index.php">Services</a></li>
        <li><a href="index.php">Contact</a></li>-->
      </ul>
    </nav>
  </header>
  <!-- Services Heading -->
  <h2 class="services-heading">SERVICES</h2>
  <div class="container">
    <div class="service-card">
      <img src="images/c1.png" alt="Wash + Fold">
      <h3 class="service-title">Wash + Fold</h3>
      <button class="service-button" onclick="window.location.href='wash_fold.php'">More
Details</button>
    </div>
    <div class="service-card">
      <img src="images/c2.png" alt="Wash + Iron">
      <h3 class="service-title">Wash + Iron</h3>
      <button class="service-button" onclick="window.location.href='wash_iron.php'">More
Details</button>
    </div>
    <div class="service-card">
      <img src="images/c3.png" alt="Dry Clean">
      <h3 class="service-title">Dry Clean</h3>
      <button class="service-button" onclick="window.location.href='dry_clean.php'">More
Details</button>
    </div>
    <div class="service-card">
      <img src="images/c4.png" alt="Steam Iron">
      <h3 class="service-title">Steam Iron</h3>
      <button class="service-button" onclick="window.location.href='steam_iron.php'">More
Details</button>
    </div>
    <div class="service-card">
      <img src="images/c5.png" alt="Stain Clear">
      <h3 class="service-title">Stain Clear</h3>
      <button class="service-button" onclick="window.location.href='stain_clear.php'">More
Details</button>
    </div>
    <div class="service-card">
      <img src="images/c6.png" alt="Donation">
      <h3 class="service-title">Donation</h3>
      <button class="service-button" onclick="window.location.href='donation.php'">More
Details</button>
```

```
    </div>
  </div>
  <script src="https://kit.fontawesome.com/your-font-awesome-kit-id.js"
crossorigin="anonymous"></script>
</body>
</html>
```

Place_order.php

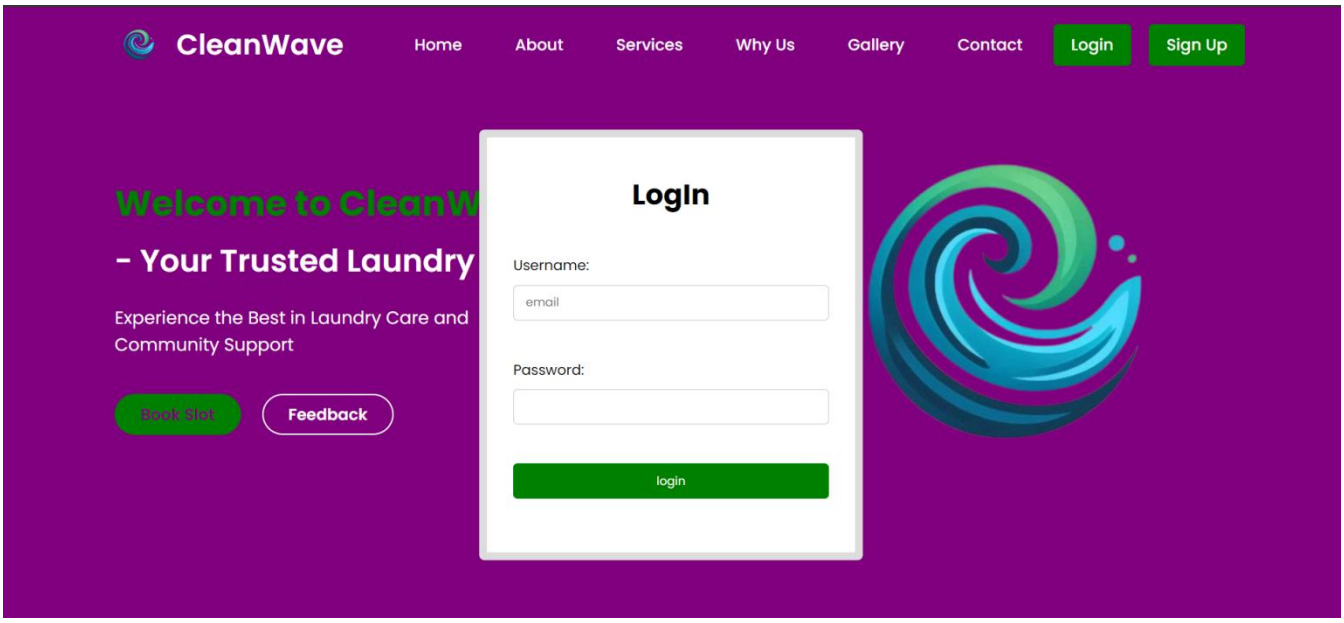
```
<?php
// Include the database connection file
include('includes/db_connect.php');
session_start();
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  // Capture form data
  $customer_name = $_POST['customer_name'];
  $items = $_POST['items'] ?? []; // Selected items
  $quantity = $_POST['quantity'] ?? []; // Corresponding quantities
  // Capture the service name from the form
  if (isset($_POST['service']) && !empty($_POST['service'])) {
    $service_name = $_POST['service']; // Use the service name passed in the form
  } else {
    // If service name is not provided, return an error message (optional handling)
    die("Error: No service selected.");
  }
  // Insert data into the orders table
  try {
    $pdo->beginTransaction();
    // Insert into orders table
    $query = "INSERT INTO orders (customer_name, service, order_date, status)
          VALUES (:customer_name, :service, CURDATE(), 'booked')";
    $stmt = $pdo->prepare($query);
    $stmt->execute([
      ':customer_name' => $customer_name,
      ':service' => $service_name   ]);
    // Get the last inserted order ID
    $order_id = $pdo->lastInsertId();
    // Insert each item into the order_items table
    foreach ($items as $item_name) {
      if (!empty($quantity[$item_name])) {
        $item_quantity = $quantity[$item_name];
        $query = "INSERT INTO order_items (order_id, item_name, quantity)
VALUES(:order_id, :item_name, :quantity)";
        $stmt = $pdo->prepare($query);
        $stmt->execute([':order_id' => $order_id':item_name' => $item_name,':quantity' =>
$item_quantity
        ]);  }      }
    // Commit the transaction
    $pdo->commit();
    echo "Order placed successfully!";
  } catch (Exception $e) {
    // Rollback the transaction if an error occurs
    $pdo->rollBack();
    echo "Failed to place order: " . $e->getMessage();
  }}?>
```

Place_order.php

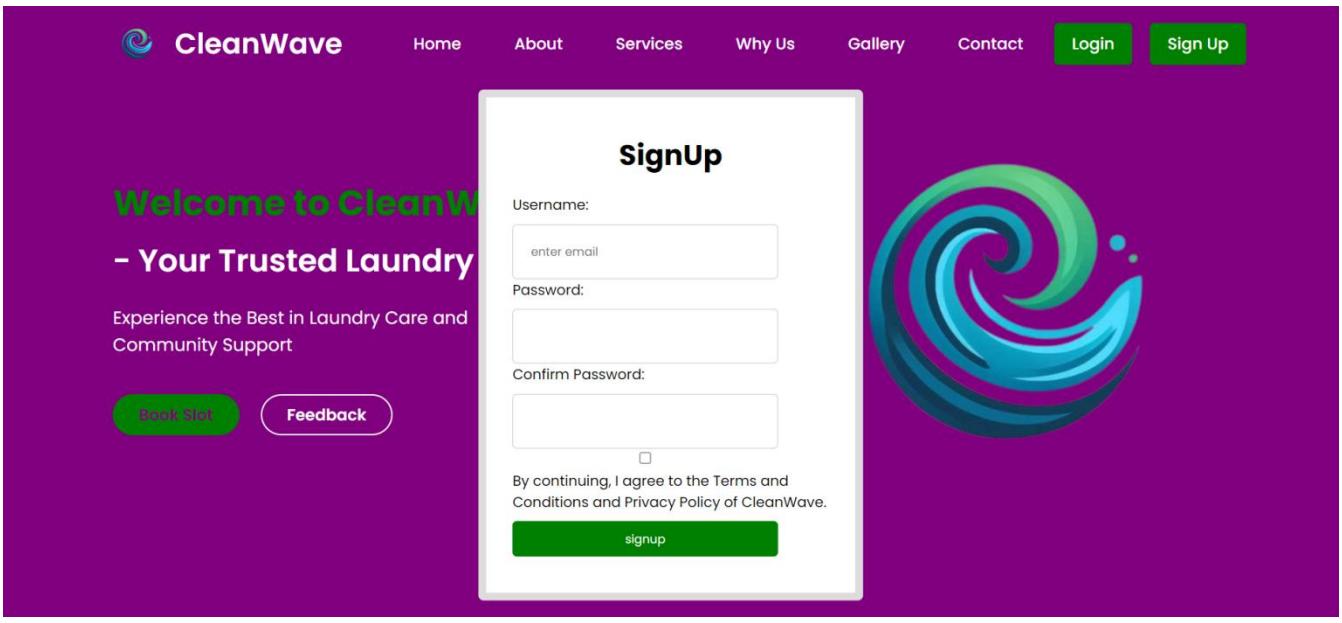
```
<?php
// Include the database connection file
include('includes/db_connect.php');
session_start();
// Check if the form was submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
   // Capture form data
   $customer_name = $_POST['customer_name'];
   $items = $_POST['items'] ?? []; // Selected items
   $quantity = $_POST['quantity'] ?? []; // Corresponding quantities
   // Capture the service name from the form
   if (isset($_POST['service']) && !empty($_POST['service'])) {
      $service_name = $_POST['service']; // Use the service name passed in the form
   } else {
      // If service name is not provided, return an error message (optional handling)
      die("Error: No service selected.");
   }
   // Insert data into the orders table
   try {
      $pdo->beginTransaction();

      // Insert into orders table
      $query = "INSERT INTO orders (customer_name, service, order_date, status)
            VALUES (:customer_name, :service, CURDATE(), 'booked')";
      $stmt = $pdo->prepare($query);
      $stmt->execute([
         ':customer_name' => $customer_name,
         ':service' => $service_name
      ]);
      // Get the last inserted order ID
      $order_id = $pdo->lastInsertId();

      // Insert each item into the order_items table
      foreach ($items as $item_name) {
         if (!empty($quantity[$item_name])) {
            $item_quantity = $quantity[$item_name];
            $query = "INSERT INTO order_items (order_id, item_name, quantity)
                  VALUES (:order_id, :item_name, :quantity)";
            $stmt = $pdo->prepare($query);
            $stmt->execute([
               ':order_id' => $order_id,
               ':item_name' => $item_name,
               ':quantity' => $item_quantity
            ]);}}
      // Commit the transaction
      $pdo->commit();
      echo "Order placed successfully!";
   } catch (Exception $e) {
      // Rollback the transaction if an error occurs
      $pdo->rollBack();
      echo "Failed to place order: " . $e->getMessage();
   }
}
?>
```

APPENDIX- II SCREENSHOTS

Login form

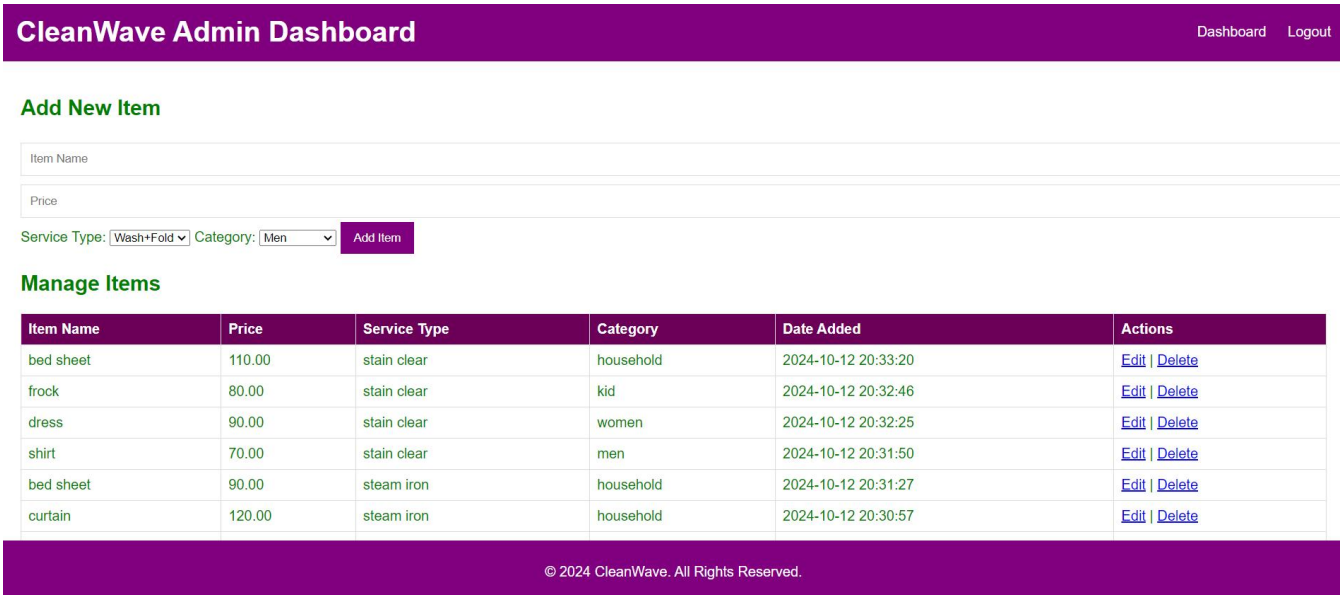


SignUp form

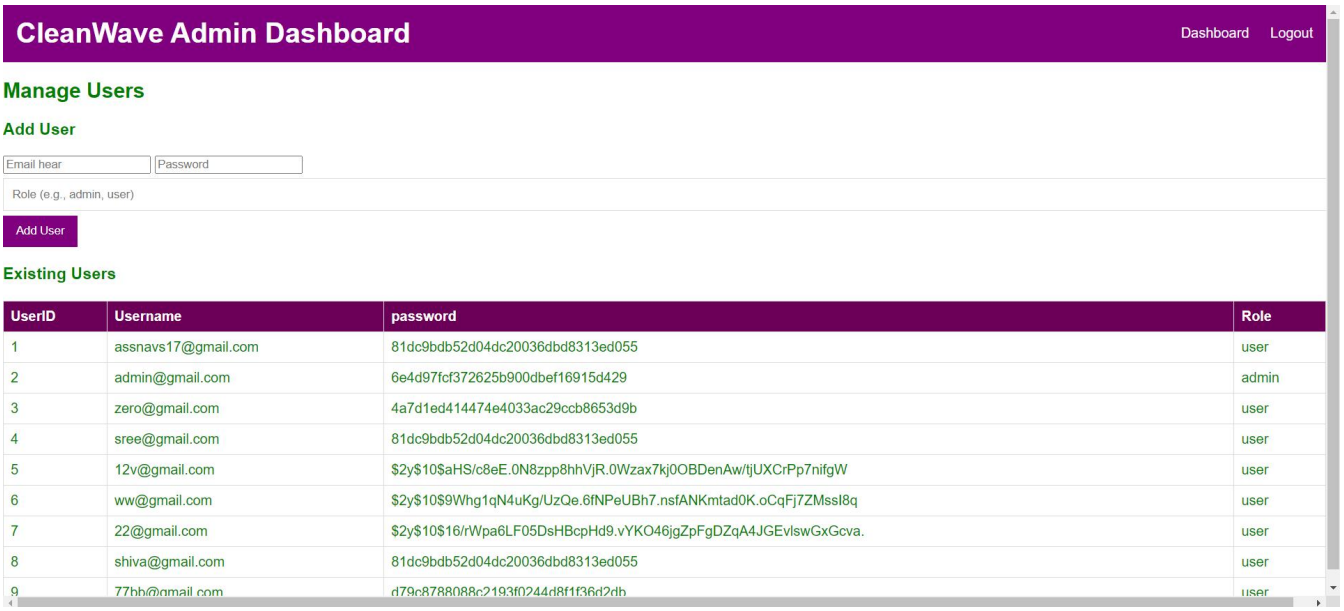


Admin Dashboard

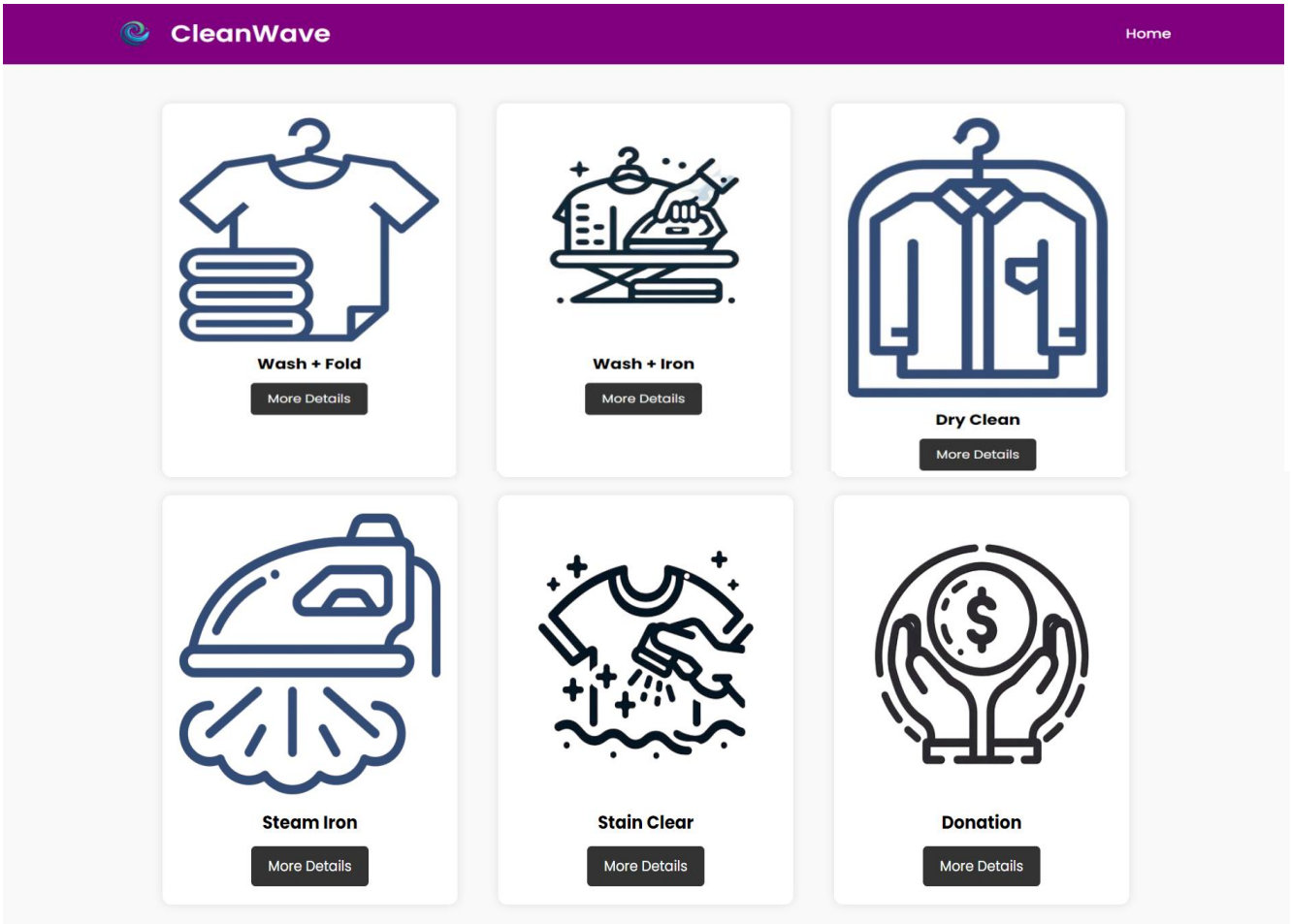
Manage Items



Manage User



Service.php



Booking slot

Index page