INTEGRATION OF B2C, C2B, B2B MODEL IN E-COMMERCE WEBSITE

**A PROJECT REPORT**

***Submitted by***

**CHANDRU M (110521205007)**

**DEVIPRIYA J (110521205008)**

**LOKESH D (110521205304)**

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

**BACHELOR OF TECHNOLOGY**

**IN INFORMATION TECHNOLOGY**

****

**GOJAN SCHOOL OF BUSINESS AND TECHNOLOGY**

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2025**

ANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

Certified that this project report “**INTEGRATION OF B2B, C2B, B2B MODEL IN E-COMMERCE WEBSITE”** is the Bonafide work of **CHANDRU M (110521205007), DEVIPRIYA J (110521205008), LOKESH D (11051205304),** who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  **Mrs. REKHA S., M.E**  **SUPERVISOR**  **Assistant Professor**  Department of Information Technology, Gojan School of Business and Technology, 80 Feet Road,  Edapalayam, Redhills ,  Chennai-52 | **SIGNATURE**  **Mrs. REKHA S., M.E.**  **HEAD OF THE DEPARTMENT**  **Assistant Professor**  Department of Information Technology, Gojan School of Business and Technology, 80 Feet Road,  Edapalayam, Redhills ,  Chennai-52 |

**CERTIFICATE OF EVALUATION**

|  |  |  |
| --- | --- | --- |
| **COLLEGE NAME** | **:** | Gojan School of Business &Technology |
| **DEGREE/BRANCH** | **:** | B. TECH / Information Technology |
| **SEMESTER** | **:** | VIII |
| **NAME OF THE STUDENTS** | **:** | **CHANDRU M (110521205007)**  **DEVIPRIYA J (110521205008)**  **LOKESH D (110521205304)** |
| **TITLE OF THE PROJECT** | **:** | **INTEGRATION OF B2B, C2B, B2B MODEL IN E-COMMERCE WEBSITE** |
| **SUPERVISOR** | **:** | **Mrs. REKHA S., M.E** |

The report of this project has been submitted by the above student in partial fulfillment for the award of the degree **Bachelor of technology in Information Technology** from **Anna University** Chennai. The report has been evaluated and the authenticity of the work has been confirmed.

**Submitted for the University Examination held on**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGMENT**

We express our deepest gratitude to our **Chairman Dr. G. Natarajan, Ph.D.,** and **Chairperson Mrs. Brindha Natarajan, B.Com,** for their valuable guidance and blessings.

We are deeply indebted to our beloved **Principal Dr. C. Selvakumar, Ph.D.,** Gojan School of Business and Technology, for providing us an excellent environment to carry out our course successfully.

We also related to express our thanks to our **Head of the Department Mrs.Rekha S ,M.E Assistant Professor**, who has been a constant source of inspiration and guidance in the course of the project.

We record our sincere thanks to our **Supervisor Mrs.Pavithra R., M.E, Assistant Professor**, for being instrumental in the completion of our project with his exemplary guidance.

We thank all the **Staff members** of our department for their valuable support and assistance at various stages of our project development.

Finally, we take this opportunity to extend our deep sense of gratitude and appreciation to our family and friends for all that they meant to us during crucial the times of the completion of our project.

**ABSTRACT**

This project focuses on the design and implementation of a feature-rich e-commerce platform dedicated exclusively to sustainable products, incorporating the integration of B2C (Business to Consumer), C2B (Consumer to Business), and B2B (Business to Business) models within a unified web application. The system is developed using HTML and CSS for the frontend, Flask for the backend logic, and MySQL for robust data storage and management. The platform supports various user roles—admin, sellers, and customers—with unique functionalities tailored to each role.The B2C model enables customers to browse and purchase eco-friendly products directly from sellers. The C2B model allows consumers to propose resale or return of products, enabling circular commerce. The B2B component facilitates bulk orders and quotations, streamlining wholesale transactions between businesses. Key features include secure user authentication, product categorization by sustainability attributes (such as materials and certifications), order management, bulk discounting, and dynamic notifications.All products listed on the site adhere to sustainable criteria, and the platform ensures traceability and transparency through certification tagging and sustainability scoring. This project not only aims to foster responsible consumption but also demonstrates how digital solutions can be leveraged to promote environmental awareness and support circular economy principles through integrated commerce models.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO** | **TITLE** | **PAGE**  **NO** |
|  | **ABSTRACT**  **LIST OF FIGURES LIST OF SYMBOLS**  **LIST OF ABBREVIATIONS** | **V VIII IX**  **XII** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 GENERAL | **1** |
|  | 1.2 EXISTING SYSTEM | **1** |
|  | 1.2.1 DISADVANTAGES | **1** |
|  | 1.3 PROPOSED SYSTEM | **2** |
|  | 1.3.1 ADVANTAGES | **4** |
| **2** | **LITERATURE SURVEY** | **5** |
| **3** | **SYSTEM REQUIREMENT** | **10** |
|  | 3.1 GENERAL | **10** |
|  | 3.2 HARDWARE REQUIREMENTS | **11** |
|  | 3.3 SOFTWARE REQUIREMENTS | **11** |
| **4** | **SYSTEM DESIGN** | **12** |
|  | 4.1 SYSTEM ARCHITECTURE | **12** |
|  | 4.2 WORK FLOW DIAGRAM | **13** |
|  | 4.3 USE CASE DIAGRAM | **14** |
|  | 4.4 DATA FLOW DIAGRAM | **15** |
|  | 4.5 MODULE DESIGN | **16** |
|  | 4.6 DATABASE DESIGN | **17** |
|  | 4.7 SECURITY DESIGN | **18** |
|  | 4.8 SYSTEM OVERFLOW | **19** |

|  |  |  |
| --- | --- | --- |
| **5** | **MODULE DESCRIPTION** | **21** |
|  | 5.1 USER AUTHENTICATION | **21** |
|  | 5.2 PRODUCT MANAGEMENT | **23** |
|  | 5.3 CART AND ORDER | **24** |
|  | 5.4 RETURN AND REQUIREMENT | **26** |
|  | 5.5 BULK REQUEST | **27** |
|  | 5.6 SECURITY | **30** |
|  | 5.6.1 NETWORK SECURITY | **30** |
|  | 5.6.2 ADVANCE ENCRYPTION STANDARD | **31** |
|  | 5.7 ADMIN CONTROL | **32** |
| **6** | **CONCLUSION AND FUTURE ENHANCEMENT** | **40** |
| **7** | **SAMPLE CODING** | **41** |
| **8** | **SCREENSHOTS** | **61** |
|  | **REFERENCES** | **64** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE.NO** | **NAME OF FIGURE** | **PAGE.NO** |
| 4.1 | HOME PAGE OF SUSTAINABLE  E-COMMERCE WEBSITE | 12 |
| 4.2 | USER REGISTRATION AND LOGIN INTERFACE | 13 |
| 4.3 | ADMIN PANEL FOR USER AND PRODUCT OVERSIGHT | 14 |
| 4.4 | PRODUCT LISTING PAGE WITH CATEGORIES | 15 |
| 4.5 | ORDER HISTORY AND RETURN REQUEST | 16 |
| 4.6 | BULK ORDER REQUEST FORM | 17 |
| 4.7 | DATABASE SCHEMA DIAGRAM | 18 |

**LIST OF SYMBOLS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **DESCRIPTION** |
| 1. | Assignment operator (==) | Assigns the value on the right to the variable on the left. |
| 2. | Equality Operator (=) | Compares two values for equality. |
| 3. | Not Equal Operator (=!) | Returns True if two values are not equal. |
| 4. | Greater Than (>) | Checks if the value on the left is greater than the value on the right. |
| 5. | Less Than (<) | Checks if the value on the left is less than the value on the right. |

|  |  |  |
| --- | --- | --- |
| 6. | Addition / Concatenation (+) | Adds numbers or joins strings. |
| 7. | Indexing/Access Operator (:) | Accesses elements in a list, dictionary, or array. |
| 8. | Function Call/Grouping (()) | Used for function calls or grouping expressions. |
| 9. | Dictionary/Block Scope ({}) | Defines dictionaries or blocks in JavaScript. |
| 10. | Dictionary/Block Scope(::) | Defines dictionaries or blocks in JavaScript. |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **B2B** | Business to Business |
| **B2C** | Business to Customer |
| **C2B** | Customer to Business |
| **UI** | User Interface |
| **UX** | User Experience |
| **CSS** | Cascading Style Sheet |
| **HTML** | Hyper Text Markup Language |
| **JS** | Java Script |
|  |  |
| **DB** | Database |
| **SQL** | Structured Query Language |
| **API** | Application Programming Interface |

**CHAPTER 1 INTRODUCTION**

* 1. **GENERAL**

With the rising demand for environmentally responsible products, sustainable e-commerce has emerged as a powerful avenue to promote eco-conscious consumption. This project focuses on building a full-stack e-commerce platform that deals exclusively with sustainable products. The application integrates three major business models—Business to Consumer (B2C), Consumer to Business (C2B), and Business to Business (B2B)—into a single ecosystem. Developed using HTML, CSS, Flask, and MySQL, this system supports multiple user roles, including admin, sellers, and consumers, and is equipped with functionalities like product management, bulk order handling, requirement proposals, and return features. The aim is to bridge the gap between sustainability-conscious consumers and environmentally responsible sellers.

* 1. **EXISTING SYSTEM**

Most existing e-commerce platforms primarily focus on traditional consumer goods and are not tailored to handle sustainable product lifecycles or the reverse flow of goods. The existing systems often support only one business model, typically B2C, and lack the capability to manage complex interactions between consumers and sellers such as proposals, returns, and bulk transactions.

* + 1. **DISADVANTAGES**
       - Lack of dedicated support for sustainable or eco-friendly products
       - Single-model operation (only B2C or B2B, not integrated)
       - No support for consumer-driven selling (C2B)
       - Limited tracking of sustainability attributes (e.g., materials, certifications)
       - Inadequate return or reuse mechanism
       - Lack of transparency in sourcing and product lifecycle
  1. **PROPOSED SYSTEM:**

The proposed system is a full-stack web-based e-commerce platform exclusively focused on promoting sustainable products and responsible commerce. Unlike traditional platforms, it integrates three core business models—B2C (Business to Consumer), C2B (Consumer to Business), and B2B (Business to Business)—into a single cohesive system. Built using HTML, CSS, Flask, and MySQL, the platform provides a seamless and dynamic user experience tailored for different roles including administrators, sellers, and customers.

In the B2C model, consumers can browse and purchase eco-friendly products directly from certified sellers. In the C2B model, customers can submit requests to return or resell used sustainable products, encouraging reuse and circular economy practices. The B2B model facilitates bulk transactions and quotations between businesses, supporting wholesale and supply chain needs.

a commitment to confidentiality throughout the platform. This approach enhances the overall privacy and security aspects of the user experience. Through the rigorous verification process, the application builds a foundation of trust and confidence among users. Customers can confidently request rides, knowing that only verified drivers are part of the service. This sense of security contributes to a positive user experience and fosters long-term relationships between customers, drivers, and service providers. The system includes modules for user authentication, product and inventory management, bulk order handling, quotation proposals, review and rating mechanisms, and return management. It also emphasizes product sustainability by capturing detailed information such as sustainability scores, eco-certifications, and material origins. Admins have control over user roles, content moderation, and system configuration. This integrated platform not only encourages eco-conscious decision-making but also creates a scalable, modular framework capable of growing with future sustainability demands and technological upgrades. Key features include **role-based dashboards**, **product lifecycle tracking**, **notification systems**, **secure login and sessions**, **inventory control**, and **return handling**. The admin can oversee user activities, approve listings, and moderate platform interactions to maintain integrity. This system promotes eco-conscious commerce by combining sustainability with digital convenience. It not only addresses current market gaps but also sets a strong foundation for future enhancements like AI-based recommendations, carbon footprint tracking, and blockchain for supply chain transparency.

* + 1. **ADVANTAGES**
* Unified platform supporting B2C, C2B, and B2B business models
* Promotes eco-friendly products and responsible consumption
* Features like return management and proposal submission encourage product reuse
* Sustainability information (materials, certifications) enhances transparency
* Admin dashboard to manage users, orders, and products effectively
* Scalability and modular design to incorporate additional features easily

**CHAPTER 2 LITERATURE SURVEY**

* 1. **TITLE:** Challenges in Building Sustainable E-Commerce Platforms

**AUTHOR:** Amit Kumar and Meena Gupta

**YEAR:** 2022

Since 2011, The evolution of e-commerce has driven convenience and accessibility for global consumers, but sustainability has often been overlooked in traditional platforms. Companies like **Amazon** and **Flipkart** have transformed digital retail through the B2C (Business to Consumer) model, yet lack infrastructure for circular product flows or transparency in sustainable practices.In their 2022 publication *“Challenges in Building Sustainable E-Commerce Platforms,”* Amit Kumar and Meena Gupta explore the growing need for incorporating sustainability into online commerce systems. With the rapid expansion of e-commerce globally, the environmental impacts of production, packaging, logistics, and product disposal have become significant concerns. The authors emphasize that most mainstream platforms focus on operational efficiency and user convenience while largely ignoring sustainability goals such as circular product flow, material traceability, and environmental certifications.The paper identifies several key challenges in designing sustainable e-commerce systems. One major challenge is the **lack of infrastructure** to support reverse logistics, which is essential for product reuse, returns, and recycling. Another is the **absence of consumer engagement features** that promote eco-friendly behavior, such as trade-in programs, sustainability scoring, or detailed product origin data. Additionally, most platforms lack **integration with regulatory compliance systems**, making it difficult to verify environmental claims through recognized certifications like FSC, GOTS, or RoHS.

* 1. **TITLE:**  Business Model Innovation for Sustainability in E-Commerce

**AUTHOR:** Shweta Mehta and Ramesh Joshi

**YEAR:** 2022

Their findings advocate for systems that not only sell but also buy back or redistribute used goods, and cater to bulk requests for sustainable procurement in the B2B sector. In the 2022 study titled *“Business Model Innovation for Sustainability in E-Commerce,”* authors Shweta Mehta and Ramesh Joshi present a comprehensive examination of how business model innovation can enhance sustainability in the e-commerce domain. The paper emphasizes that while traditional e-commerce focuses on sales efficiency and logistics, integrating sustainability requires a fundamental rethinking of the business architecture, value creation, and customer engagement processes.The authors argue that sustainable e-commerce should not be restricted to the conventional Business to Consumer (B2C) model. Instead, platforms must embrace hybrid business models that incorporate Consumer to Business (C2B) and Business to Business (B2B) operations. Such integration allows for features like product resale, reverse supply chains, bulk eco-procurement, and long-term partnerships focused on environmentally conscious practices.

* 1. **TITLE :** Flask Web Development: Developing Web Applications with Python

**AUTHOR:** Miguel Grinberg

**YEAR :**2018

In this paper, advocates for Flask due to its simplicity, scalability, and ease of integration with SQL databases like MySQL, making it ideal for educational and commercial full-stack applications **Oracle’s MySQL Documentation (2023)** further supports its use as a reliable relational database system for structured e-commerce data storage and complex queries.In his book *“Flask Web Development: Developing Web Applications with Python,”* Miguel Grinberg (2018) provides an in-depth guide to building scalable and maintainable web applications using the Flask microframework. The book has become a widely referenced resource for developers and students due to its practical approach to full-stack development and its flexibility in integrating various components of a modern web system. Grinberg explains that Flask is a lightweight, modular framework that allows developers to build web applications quickly without the complexity of larger frameworks like Django. The book emphasizes the use of **route-based architecture**, **template rendering**, **request handling**, and **database integration** using SQLAlchemy, making it highly suitable for rapid prototyping and scalable application development.A notable aspect of the book is its emphasis on **blueprint-based modular design**, which promotes clean separation of concerns—essential for larger applications involving multiple roles such as admins, sellers, and customers, as seen in sustainable e-commerce systems. The book also covers **user authentication**, **form handling**, and **RESTful API integration**, all of which are crucial for the backend functionality of dynamic platforms.

* 1. **TITLE:** ThredUp Resale Report"

**AUTHOR: Patagonia’s Worn Wear** and **ThredUp**

**YEAR :**2021

The *“ThredUp Resale Report”* (2021), collaboratively referenced by Patagonia’s Worn Wear and ThredUp, offers valuable insights into the environmental and economic potential of the second-hand market in the digital commerce space. The report highlights the rise of the resale economy and its crucial role in reducing environmental waste, conserving resources, and promoting sustainable consumer behavior. According to the report, the fashion industry is one of the largest polluters globally, responsible for massive carbon emissions, water consumption, and textile waste. In response, platforms like **ThredUp** and **Patagonia’s Worn Wear** have pioneered the integration of **Consumer to Business (C2B)** models within e-commerce, where consumers can resell used products, particularly clothing, to promote reuse and extend product lifecycles. The report reveals that resale is projected to double in value over the coming five years, surpassing the growth rate of traditional retail.The document emphasizes how digital platforms can effectively enable circular economy principles by supporting reverse logistics, smart sorting systems, condition-based product evaluations, and eco-conscious branding. ThredUp’s model allows consumers to submit garments, which are then cleaned, verified, and resold—creating a sustainable loop of consumption. For the current project, this report provides foundational relevance. The proposed sustainable e-commerce platform supports C2B functionality, allowing users to submit previously purchased items for resale or return. Additionally, it promotes transparency by tagging products with **sustainability metrics**, such as certifications and material sources, encouraging informed consumer decisions.

* 1. **TITLE:** Integrating Circular Economy Principles in E-Commerce Platforms

**AUTHOR:**  L. Zhang and K. Tan

**YEAR:** 2021

The growing environmental challenges, such as climate change, resource depletion, and waste management, have prompted industries across the world to adopt sustainability practices. One of the most prominent strategies in recent years is the Circular Economy (CE) model, which emphasizes rethinking traditional linear models of consumption by reducing waste, reusing resources, and recycling materials. The authors, L. Zhang and K. Tan, in their 2021 paper, explore how circular economy principles can be effectively integrated into e-commerce platforms to foster sustainability while meeting consumer demands. Circular economy is a model designed to minimize waste and make the most of available resources. Unlike the traditional "take, make, dispose" model of linear economies, a circular economy promotes a more regenerative system, where products and materials are reused, refurbished, remanufactured, and recycled. In a circular economy, products are designed with their entire lifecycle in mind, ensuring they can be recycled or reused after their initial use phase. This model aims to close the loop of product lifecycles through greater resource efficiency, which contrasts with the conventional approach of extracting raw materials, using them to produce goods, and discarding them after use. The rise of e-commerce has contributed to increased resource consumption and waste generation. With the exponential growth of online shopping platforms, more products are being sold, often leading to higher packaging waste, fast-fashion consumption, and an increase in the number of discarded goods. However, e-commerce platforms are uniquely positioned to integrate circular economy principles due to their global reach and ability to facilitate direct interactions between producers and consumers. Through innovative technologies and business practices, e-commerce can transform its role from contributing to environmental degradation to promoting sustainability.

A formal representation of Digital Identity (D-ID) would provide support for the general structure of any D-ID, its information value, its security implication, its complexity assessment, and the optimal/minimum structure to create a complete mapping to build any D-ID without unnecessary redundant information and complexity. Our approach is to provide the ‘’minimum’’ and axiomatic frame for any D-ID. We present a liminary formal description of D-ID, its formal construction, and the basis relationship to create any D-ID through its ownership relationships in a formal frame. A major shift towards circularity starts at the design phase. E-commerce platforms should encourage manufacturers and brands to design products with the intention of extending their lifecycle. This includes using recyclable materials, designing products that can be easily disassembled for recycling, and promoting durability over disposability. Platforms can facilitate this shift by working with brands that prioritize sustainable product design and by highlighting these products to environmentally-conscious consumers. Zhang and Tan emphasize the potential of e-commerce platforms to host second-hand markets or product take-back programs. Online platforms can create sections dedicated to reselling pre-owned goods, whether they are clothes, electronics, or furniture. By offering incentives like trade-ins or buy-back programs, e-commerce companies can extend the lifespan of products, allowing them to be reused or refurbished instead of discarded. The paper highlights the importance of supply chain transparency in achieving circular economy goals. E-commerce platforms can help consumers make more sustainable choices by providing clear information on the environmental impact of products, including details on sourcing, production methods, and end-of-life options. Additionally, reverse logistics systems can be implemented to facilitate the return of products to manufacturers for reuse, repair, or recycling.

**CHAPTER 3 SYSTEM REQUIREMENTS**

* 1. **GENERAL:**

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. These are the requirements for doing the project. Without using these tools and software’s we can’t do the project. So we have two requirements to do the project. They are

The following are the requirements that are to be discussed.

* 1. Environment requirements

3.2.1 Hardware requirements

3.2.2 Software requirements

**ENVIRONMENT REQUIREMENTS**

The proposed sustainable e-commerce platform is a web-based system designed to operate efficiently in a standard development and deployment environment. It is intended to be hosted on a local server during development and on a cloud-based server or virtual machine for production deployment. The platform does not require any special environmental conditions but should ideally operate in a stable environment with:

* Reliable internet connectivity
* Secure power supply
* Proper air circulation for server hardware (if self-hosted)
* Minimal latency for frontend/backend communication during deployment
* Environments adhering to data privacy regulations (for cloud hosting)

This system can be deployed on any operating system that supports Python, MySQL, and Flask, including Windows, Linux (Ubuntu preferred), and macOS.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

PROCESSOR : Intel Core i3 / AMD Ryzen 3 or higher RAM :4 GB minimum (8 GB recommended)

MONITOR : 13" screen, 1366x768 resolution minimum

STORAGE : 20 GB HDD/SSD

**3.3 SOFTWARE REQUIREMENTS**

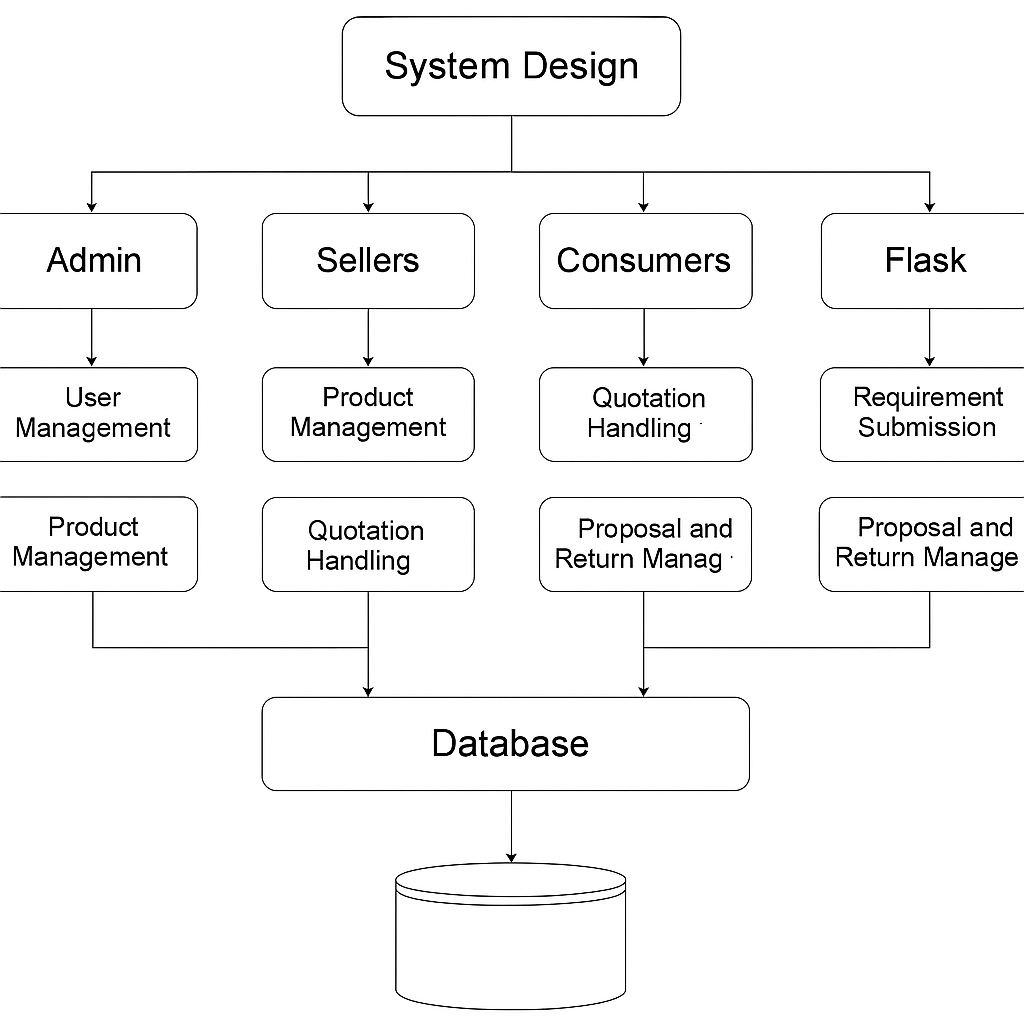
The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

Front End : HTML/CSS

Back End : MY SQL 8.0 External Backend : Flask 3.0.0

IDE : VS Code / PyCharm

**CHAPTER 4 SYSTEM DESIGN**

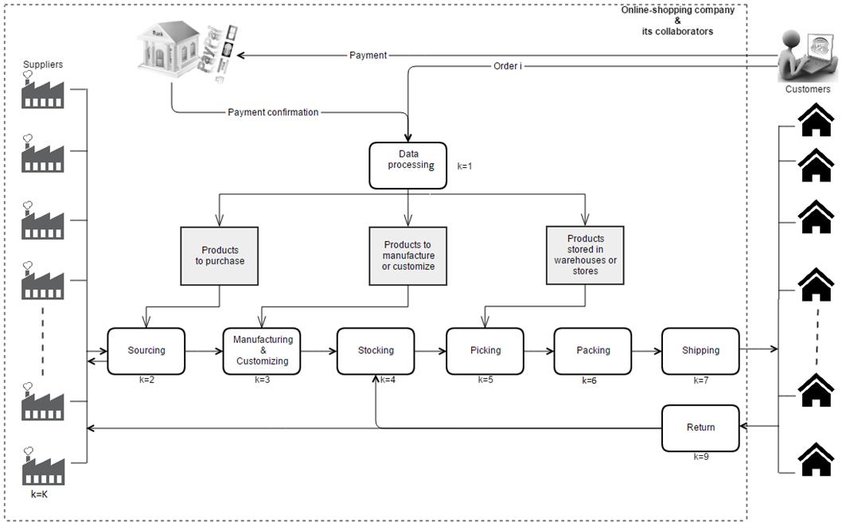
****

* 1. **SYSTEM ARCHITECTURE**

The system architecture of the proposed sustainable e-commerce platform is based on a modular and layered design, enabling scalability, maintainability, and efficient management of multiple user roles. It follows a three-tier structure consisting of the presentation layer, application layer, and data layer. Each layer has distinct responsibilities that work together to deliver a seamless and secure user experience. The presentation layer, also known as the frontend, is built using HTML and CSS. It serves as the user interface for different types of users including Admin, Seller, and Customer.

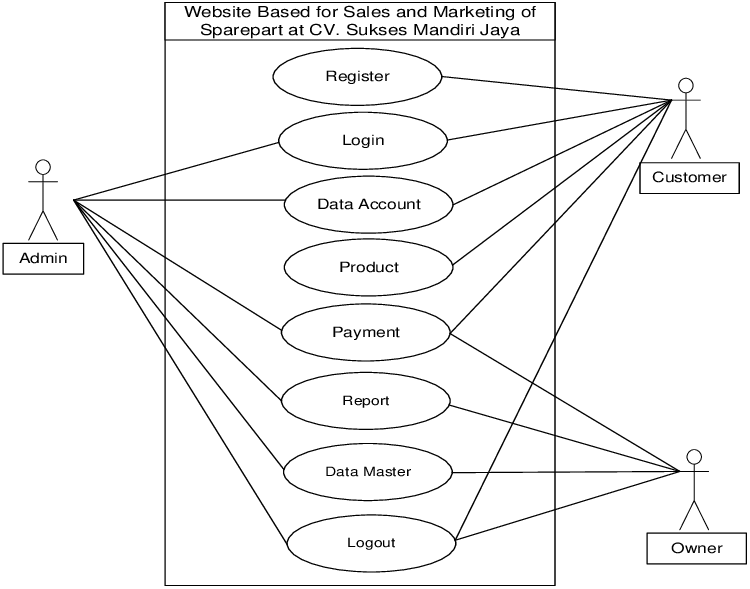
The application layer represents the backend and is developed using the Flask framework in Python. This layer processes business logic, handles form submissions, manages session control, and routes user requests to the appropriate components. It ensures security by enforcing role-based access control and validating input data. Flask Blueprints are used to organize various modules such as authentication, product management, order processing, and notifications, promoting code modularity and reusability.

**WORK FLOW DIAGRAM**

****

Sellers can upload products with sustainability details, manage stock, and respond to customer requirements. The admin oversees user management, product approvals, and platform monitoring.

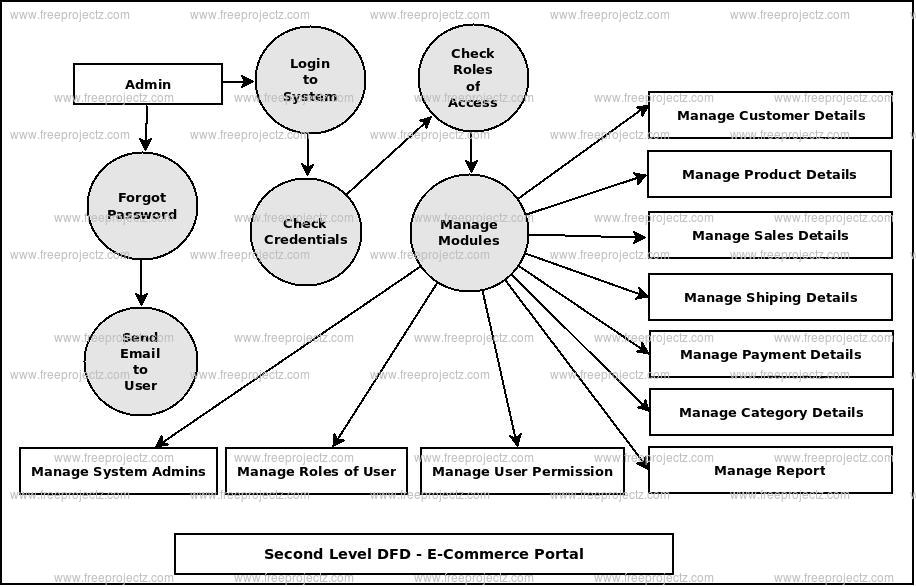
* 1. **USE CASE DIAGRAM**

****

The use case diagram is the main building block of [object oriented](http://en.wikipedia.org/wiki/Object_oriented) modeling. It is used both for general [conceptual modeling](http://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modeling translating the models into [programming code](http://en.wikipedia.org/wiki/Programming_code). For this in our component diagram first propose a data. In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data. A ride-hailing service like Uber or Lyft facilitates seamless transportation by connecting users with nearby drivers through a mobile application. The system revolves around two primary actors: the user and the driver. Users initiate interactions by registering or logging into the platform and then requesting a ride by specifying their pickup location and destination. Upon receiving a ride request, drivers have the option to either accept or reject it. Once a driver accepts, the system calculates the fare based on factors like distance, time, and any surge pricing in effect. Upon

confirmation, the ride is booked, and both user and driver are notified. During the ride, users can track the driver's location if necessary.

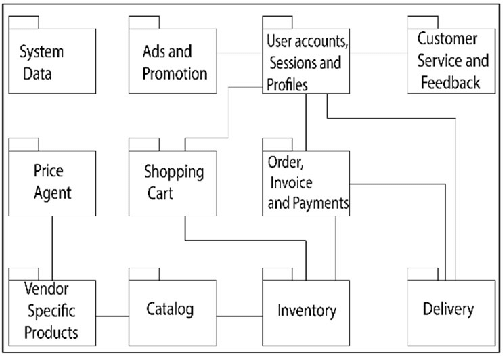
* 1. **DATA FLOW DIAGRAM**

****

The Data Flow Diagram (DFD) illustrates how data moves through the sustainable e-commerce system. It shows interactions between users (Admin, Seller, Customer) and core modules such as authentication, product management, cart, orders, and returns. Customers send requests through the interface, which are processed by the backend and reflected in the database. Sellers manage listings and respond to proposals, while admins monitor activities. Each action triggers data updates, ensuring smooth transitions between user inputs, backend logic, and storage.

Customers initiate actions like registration, login, product browsing, and placing orders, which are processed by the backend and stored in the database. Sellers add and manage sustainable products, respond to requirements, and process returns or bulk requests. Admins monitor platform activity, verify users, and manage product approvals. Data flows between the user interface, backend logic (Flask), and the MySQL database, ensuring accurate data handling and updates. Each user action triggers backend processing, which modifies or fetches data as required and responds with appropriate views or email notifications. The DFD ensures that each module operates with defined data responsibilities, promoting system clarity and efficiency.

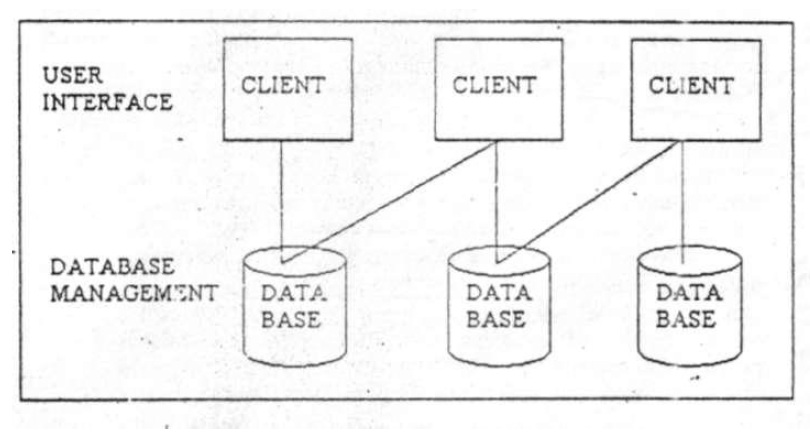
* 1. **MODULE DESIGN**

****

The module design of the sustainable e-commerce platform follows a modular architecture using Flask Blueprints to separate functionalities by user roles and features.

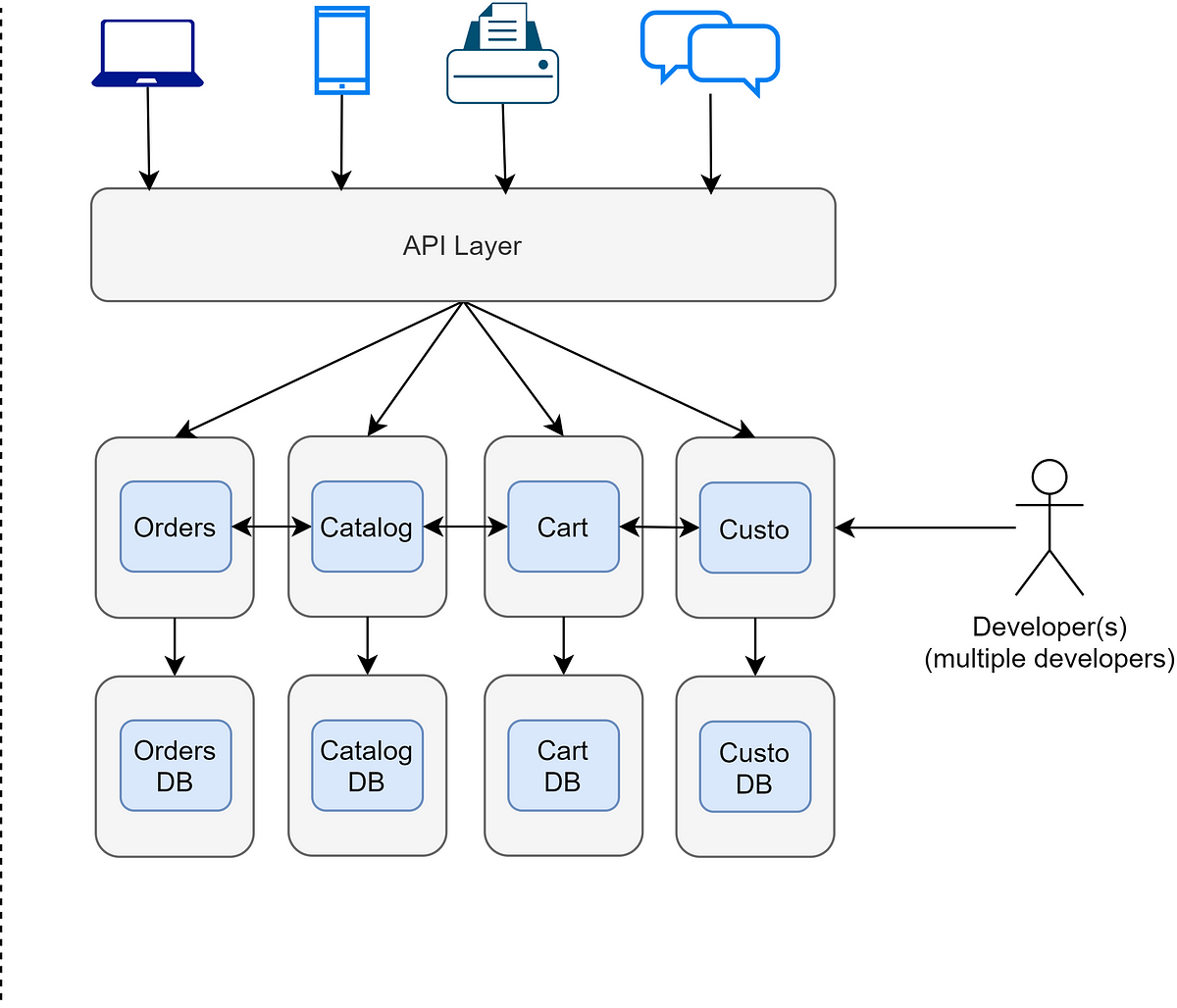
Key modules include Authentication, Product Management, Cart and Orders, Admin Dashboard, Notifications, Requirements and Proposals, and Bulk Quote Handling. Each module operates independently, handling specific tasks such as user registration, product listing, order processing, and communication between sellers and customers. The modular structure enhances scalability, maintainability, and ease of debugging. It also ensures role-based access control, where each user interacts only with the relevant modules based on their permissions and responsibilities within the system.

* 1. **DATABASE DESIGN**

****

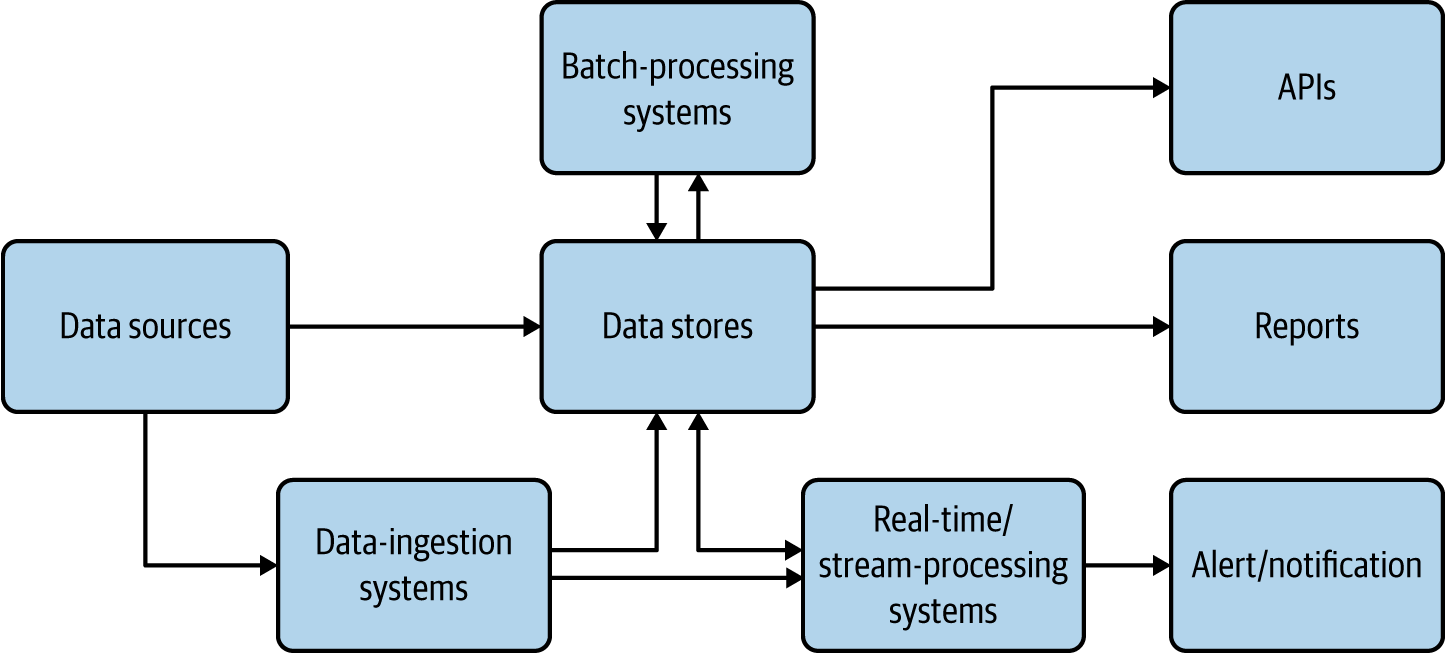
The database design of the sustainable e-commerce platform is structured using a **relational model** implemented in **MySQL**, ensuring data consistency, integrity, and scalability. The schema follows a normalized structure to reduce redundancy and optimize performance. Key entities include **Users**, **Products**, **Orders**, **Cart**, **Proposals**, **Quotes**, **Requirements**, **Payments**, and **Notifications**. Each table is connected through primary and foreign key relationships to support various interactions across the platform.

* 1. **SECURITY DESIGN**

****

Security is a critical aspect of the sustainable e-commerce platform, as it involves sensitive user data, transactional information, and role-based operations. The security design of this system ensures data protection, secure access, and prevention against common web vulnerabilities. The application follows best practices in web security using a combination of **Flask's built-in security features**, custom validations, and database-level safeguards. **User authentication** is handled through the Flask-Login extension, enabling secure login sessions and password hashing using industry-standard algorithms like **PBKDF2 or SHA-256**. Passwords are never stored in plain text, ensuring safety even if database leaks occur.

* 1. **SYSTEM OVERFLOW**

****

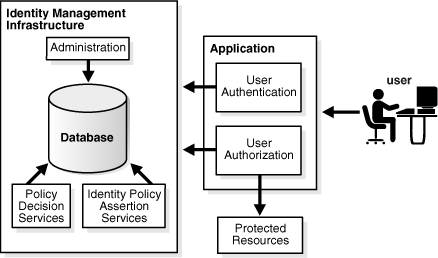
The system workflow of the sustainable e-commerce platform describes how various user types interact with the system to perform key operations. The platform supports three roles—**Admin**, **Seller**, and **Customer**—each with specific workflows designed to maintain operational clarity and efficiency across B2C, C2B, and B2B models.

The workflow begins with **user authentication**, where a new user registers by selecting their role and providing basic information. After registration, users must log in using valid credentials. The system verifies login details using Flask-Login and redirects users to their respective dashboards based on their roles.For **customers**, the workflow involves browsing sustainable products, adding items to the cart, placing orders, and making payments. They can also submit requirements for returns or proposals for bulk purchase (C2B and B2B flows). Once an order is placed, a confirmation notification is sent, and order data is stored securely in the database. **Sellers** follow a workflow that includes logging in, uploading product listings with sustainability details (materials, certifications, etc.), managing inventory, viewing customer requirements, and sending proposals or bulk quotes. Sellers can accept return requests and process reverse logistics, enabling circular commerce. **Admins** are responsible for platform oversight. Their workflow includes managing users, verifying product listings, handling system notifications, and monitoring transactions across all roles. Admins can activate or deactivate user accounts and ensure that all listed products meet sustainability criteria. All workflows interact with a common backend via secure Flask routes. The system ensures that every action—from registration to order confirmation—flows logically through request handling, backend processing, and database updates. Notifications are sent at key stages (order placed, return approved, proposal received) to keep users informed.

20

**CHAPTER 5 MODULE DESCRIPTION**

* 1. **USER AUTHENTICATION**

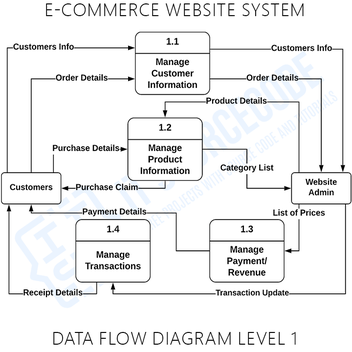
****

The **User Authentication Module** is one of the most critical components of the sustainable e-commerce platform. It is responsible for securely managing user registration, login, logout, session handling, and role-based access control. This module ensures that users can interact with the system based on their roles—**Admin**, **Seller**, or **Customer**—while keeping their data and sessions protected from unauthorized access.The **login process** involves validating user credentials against the stored hashed passwords using **Flask-Login**, a robust session management extension. Once authenticated, users are redirected to their respective dashboards based on their roles. Session cookies are securely managed using HTTP-only and same-site flags to prevent session hijacking. The module supports **role-based redirection**. For example, customers are directed to the shopping dashboard, sellers to the product management panel, and admins to the system control panel.

It also includes features like **logout** and **session expiration**, enhancing security by invalidating inactive sessions. Form data submitted during registration or login is validated using **Flask-WTF** to prevent injection attacks and ensure input correctness.Additionally, this module includes functionality for **email verification** and **password reset**, improving user trust and account recovery processes. SMTP integration allows the platform to send verification links and reset instructions securely. Overall, the User Authentication Module provides a strong foundation for platform security, user access control, and a smooth login experience, ensuring that only verified users can access and operate within the system as intended. During **registration**, users provide details such as username, email, password, and select their role. Passwords are hashed using secure algorithms like **SHA-256** before being stored in the database, ensuring they are never saved in plain text. This enhances protection against potential data breaches.The **login process** involves validating user credentials against the stored hashed passwords using **Flask-Login**, a robust session management extension. Once authenticated, users are redirected to their respective dashboards based on their roles. Session cookies are securely managed using HTTP-only and same-site flags to prevent session hijacking.The module supports **role-based redirection**. For example, customers are directed to the shopping dashboard, sellers to the product management panel, and admins to the system control panel. This ensures that users only have access to features relevant to their roles.

Managing customer support and resolving disputes or complaints promptly is essential for maintaining a positive reputation. Collaborating with local governments and stakeholders to address regulatory issues and promote sustainable mobility solutions is another important aspect.

* 1. **PRODUCT MANAGEMENT**

****

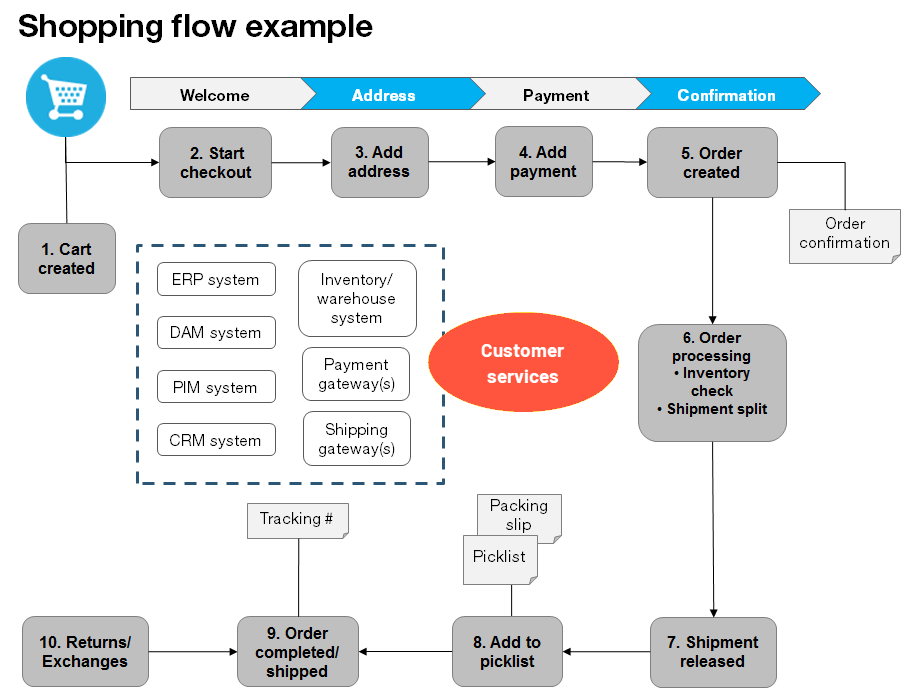
The **Product Management Module** is a core component of the sustainable e-commerce platform, enabling sellers and admins to manage the lifecycle of sustainable products on the platform. This module is designed to ensure that only eco-friendly products, which meet sustainability standards, are listed for sale to promote conscious consumption. Sellers can **add new products** through a dedicated form that captures essential details such as product name, description, category, subcategory, price, available stock, brand, and images. In addition to basic product information, the module includes specific fields to highlight **sustainability attributes**, including **materials used**, **eco-certifications** (e.g., GOTS, FSC, RoHS), and a **sustainability score**. This scoring system helps consumers evaluate how environmentally friendly a product is before purchasing. Once a product is submitted, the data is stored in the MySQL database via SQLAlchemy ORM

The module supports **CRUD operations** (Create, Read, Update, Delete). Sellers can view and manage their product listings through a dashboard, where they can edit product details, update stock availability, change prices, or remove discontinued items. These changes are reflected in real-time and synced with the backend.Admins have advanced access to **view, edit, or remove any product**, regardless of the seller, ensuring that the marketplace remains compliant with environmental standards and terms of service. Admins can also monitor seller activity to detect violations or non-compliance.On the frontend, products are displayed in a user-friendly interface with filtering options by category, price, and sustainability score, allowing customers to make informed decisions. Each product detail page shows clear descriptions, images, certifications, and stock availability.

* 1. **CART AND ORDER**

The **Cart and Order Module** is a fundamental part of the sustainable e-commerce platform, responsible for managing the purchasing process from product selection to final order placement. It is primarily used by customers to interact with the products listed by sellers and plays a critical role in supporting the B2C (Business to Consumer) model. This module ensures a smooth, secure, and efficient transaction experience for both buyers and sellers, while also maintaining accurate inventory and order tracking. The cart system allows customers to add products they intend to purchase. Each item added to the cart includes the product name, price, quantity, stock availability, seller information, and a calculated subtotal. Users can update the quantity or remove items at any point before proceeding to checkout. Real-time validation ensures that customers cannot add more items than what is available in stock.

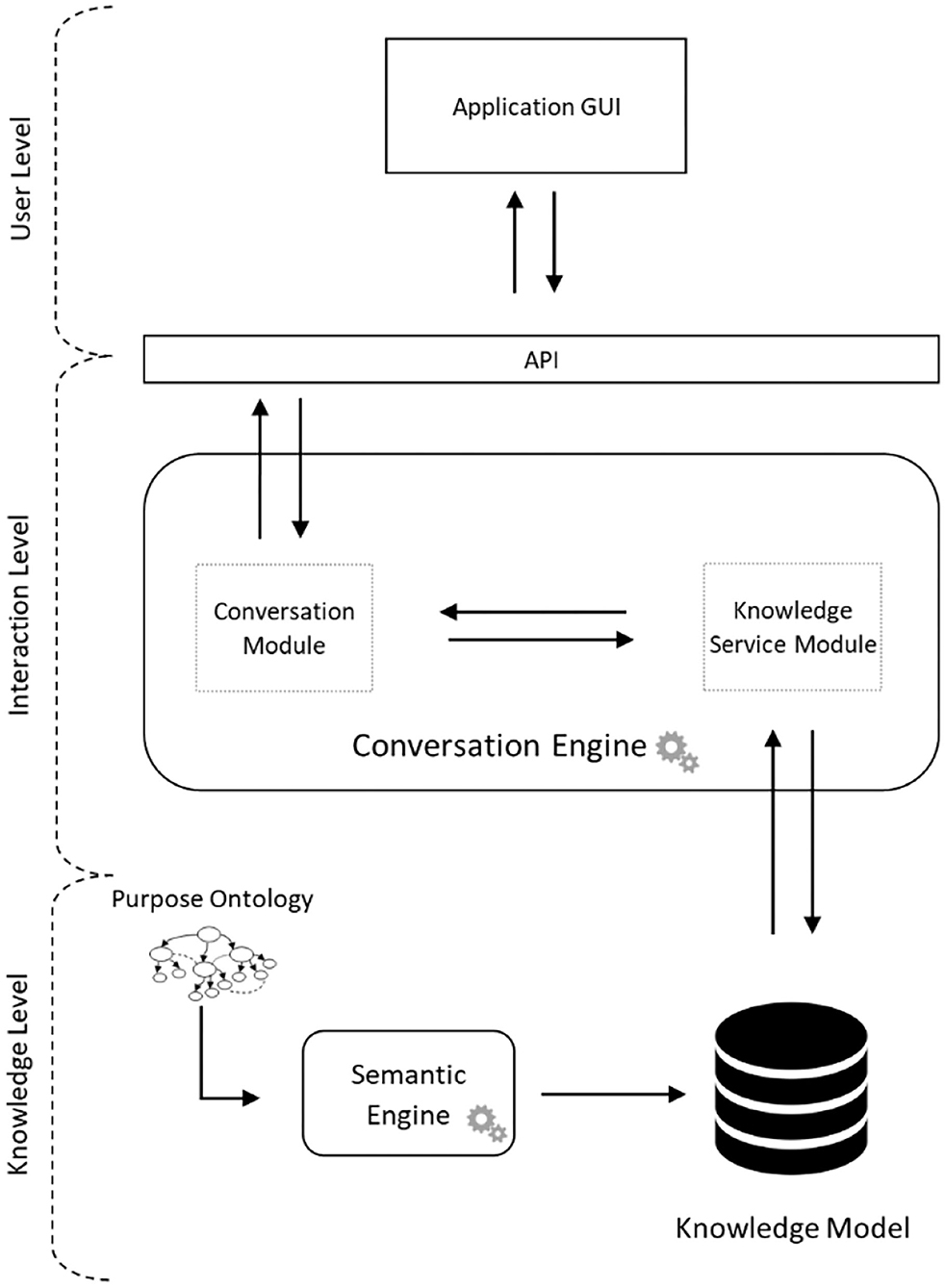
When a product is added to the cart, it is stored temporarily in the session for guest users or persistently in the database for logged-in users



Once the user proceeds to checkout, the system initiates the order creation process. Customers must confirm their shipping address and contact details. The backend verifies stock availability and calculates the total amount including applicable discounts or taxes. The payment step may be integrated with a payment gateway for secure online transactions. Upon successful order placement, an entry is created in the **Orders** table, which includes details like order ID, user ID, seller ID, product ID(s), order date, status (e.g., pending, shipped, delivered), and total amount. An email confirmation and an in-app notification are sent to the customer and the seller. Sellers can view all incoming orders on their dashboard. They can update order statuses (e.g., processing, shipped, completed), add shipping details, and mark orders as fulfilled. Admins have oversight of all orders across the platform and can intervene in case of disputes, errors, or system monitoring.

The Cart and Order Module is tightly integrated with the Product Management Module. Upon order confirmation, product stock is automatically reduced. If stock levels fall below a certain threshold, the seller receives a notification prompting restocking. This real-time inventory control prevents overselling and keeps stock information accurate.

**5.4 RETURN AND REQUIREMENT**

****

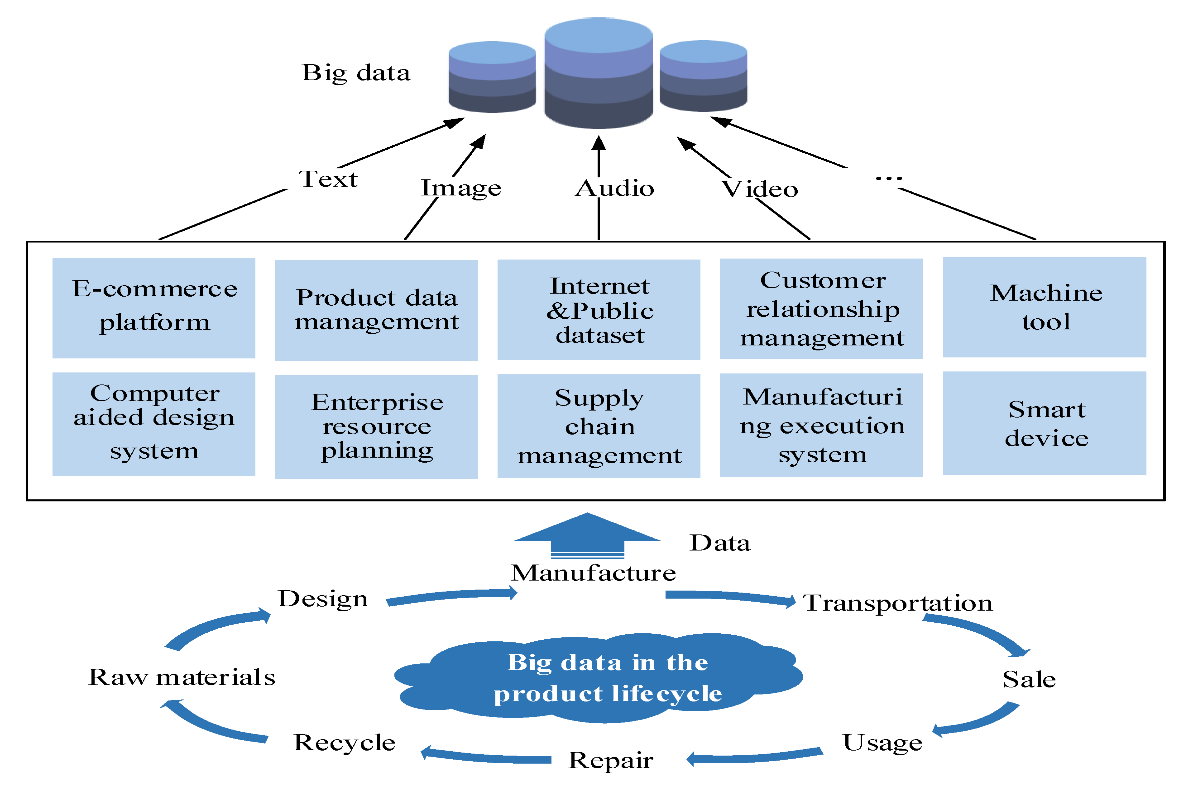
The **Return and Requirement Module** plays a vital role in supporting the **Consumer to Business (C2B)** functionality of the sustainable e-commerce platform. This module encourages **circular commerce** by allowing customers to return products they no longer need or submit specific product requirements, which can be fulfilled by sellers. It enhances the sustainability focus of the platform by promoting reuse, resale, and responsible consumption. Customers can initiate a **return request** directly from their order history dashboard. The return form captures essential details such as order ID, reason for return, product condition, and whether the return is for exchange, refund, or resale. Users can also upload images of the product to provide sellers with visual confirmation of its condition

Once submitted, the return request is stored in the **Returns** table in the MySQL database. Sellers are notified instantly and can view pending returns from their dashboard. Based on the product’s condition and the seller’s return policy, the seller can either **approve**, **reject**, or **request additional information** from the customer. If approved, logistics details (e.g., pickup address, courier partner) are shared with the customer. The module supports partial and full returns and adjusts stock and order status accordingly. For refunds, integration with payment systems ensures seamless reversal. This process is monitored by the admin to ensure fair handling of return disputes and overall compliance. Apart from returns, the module enables customers to submit **custom product requirements**. For example, a customer can request a sustainable product not currently listed—such as bulk biodegradable packaging or personalized organic cotton bags. The requirement form captures details like product type, specifications, quantity, purpose, and delivery timeline. These requirements are saved in the **Requirements** table and displayed to sellers on their dashboard. Sellers can **respond to requirements** by submitting **proposals** that include product details, pricing, delivery estimates, and certifications. This C2B model opens up the platform for reverse interactions, allowing customers to express demand and enabling sellers to fulfill it dynamically.

**5.5 BULK REQUEST**

The **Bulk Request and Quotation Module** is an essential component of the sustainable e-commerce platform that supports **Business-to-Business (B2B)** operations. This module allows business buyers to place large-volume orders for sustainable products and enables sellers to respond with custom quotes.

Registered users categorized under business or wholesale roles, or customers with large-scale requirements, can submit **bulk purchase requests** directly from the product page or through a dedicated request form. The form captures important details such as product name, quantity, preferred pricing range, delivery timeline, location, and any custom specifications like branding, packaging, or material preferences. Once submitted, these requests are saved in the **BulkRequest** table in the database. Sellers are immediately notified of new requests via email and dashboard alerts. This functionality allows businesses to communicate their exact needs to sellers instead of purchasing pre-listed quantities.



**ALGORITHM EXPLANATION**

AES is a symmetric encryption algorithm that ensures data confidentiality by converting plaintext data into ciphertext, which can only be decrypted with the correct key. This helps to prevent unauthorized access to user information

and transaction details. Additionally, other network security measures like TLS (Transport Layer Security) or SSL (Secure Sockets Layer) are used to establish secure connections between clients and servers. These protocols encrypt the data transmitted over the network, protecting it from eavesdropping and man-in-the-middle attacks.

User Authentication and Authorization: AES encryption can be used to securely store and transmit user credentials (such as usernames and passwords) between the user's device and the ride-hailing service's servers. This helps ensure that only authorized users can access the service and their account information remains protected.

Payment Security: AES encryption can be applied to encrypt payment details, such as credit card numbers or digital wallet information, during transactions between the user's device and the service's servers. This helps prevent unauthorized access to sensitive financial data and ensures the security of payment transactions.

Location Privacy: Ride-hailing services rely heavily on location data for matching users with drivers and providing navigation instructions. AES encryption can be used to encrypt location data transmitted between the user's device and the service's servers, protecting the privacy of users' movements and preventing location tracking by unauthorized parties.

Data Storage: AES encryption can also be employed to encrypt data stored on servers or in databases used by ride-hailing services. This adds an extra layer of security to protect user data in case of a security breach or unauthorized access to the server infrastructure. API Security, Many ride-hailing services offer APIs (Application Programming Interfaces) for third-party developers to integrate with their platforms. AES encryption can be used to secure communications between the service's servers and external applications using these APIs, ensuring that data exchanged remains confidential and secure. End-to-End Encryption, Some ride-hailing services may implement end-to-

end encryption using AES or similar encryption algorithms. This means that data is encrypted on the user's device and remains encrypted until it reaches its intended recipient, providing maximum security and privacy for user communications and transactions. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

**5.6 SECURITY**

**5.6.1NETWORK SECURITY TECHNIQUE :**

Network security stands as a linchpin in the operation of ride-hailing services, serving multifaceted roles in protecting sensitive user data, ensuring the integrity of communications, and upholding the trust of its clientele. Through robust encryption mechanisms like AES and secure communication protocols such as HTTPS, ride-hailing platforms safeguard critical information like user authentication details, payment information, and location data from interception and unauthorized access. Moreover, proactive measures like firewalls, intrusion detection systems, and DDoS mitigation strategies fortify the network infrastructure against malicious attacks, ensuring service availability and reliability . As cyber threats continue to evolve, ride-hailing companies prioritize ongoing employee training and awareness initiatives, cultivating a security-conscious culture to mitigate risks posed by human error and insider threats. In this interconnected digital landscape, network security emerges not only as a technological imperative but also as a cornerstone of user trust and business continuity within the ride-hailing industry.

Data Encryption: As discussed earlier, encryption algorithms like AES are used to encrypt sensitive data transmitted over networks. This includes user authentication details, payment information, and location data. Encrypting this data ensures that even if intercepted, it remains unreadable and secure.

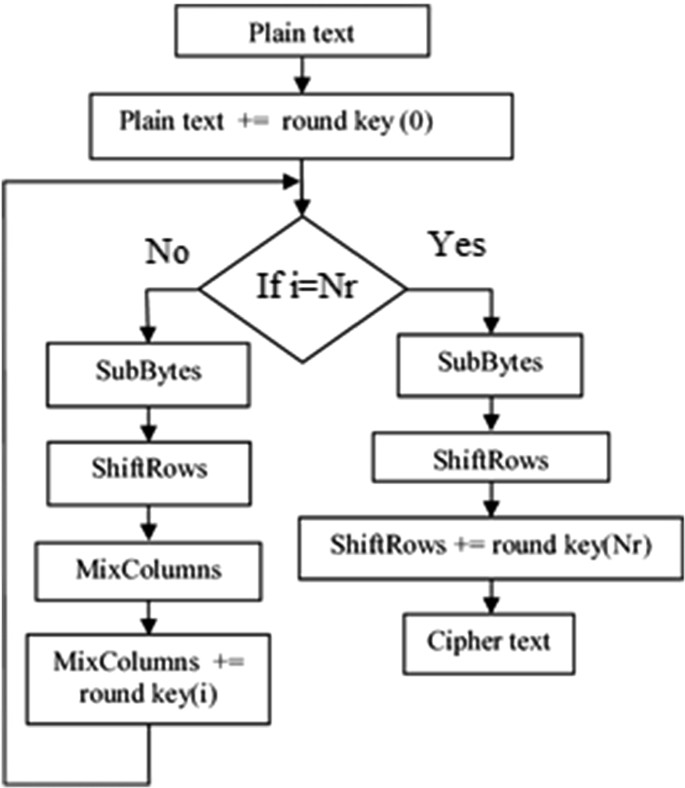
Secure Communication Protocols: Secure communication protocols like HTTPS (HTTP Secure) are employed to encrypt data in transit between the

user's device and the ride-hailing service's servers. These protocols protect against eavesdropping and interception of sensitive information.

**5.6.2 ADVANCE ENCRYPTION ALGORITHM :**

AES is a symmetric encryption algorithm that ensures data confidentiality by converting plaintext data into ciphertext, which can only be decrypted with the correct key. This helps to prevent unauthorized access to user information and transaction details. Additionally, other network security measures like TLS (Transport Layer Security) or SSL (Secure Sockets Layer) are used to establish secure connections between clients and servers. These protocols encrypt the data transmitted over the network, protecting it from eavesdropping and man-in-the-middle attacks.

Data Confidentiality, AES encrypts the data, making it unreadable to anyone without the decryption key. This ensures that sensitive information remains confidential during transit. Data Integrity, AES can also be used in conjunction with integrity checks like HMAC (Hash-based Message Authentication Code) to ensure that the data has not been tampered with during transmission. Secure Communication.



This encryption helps protect against unauthorized access and ensures the confidentiality and integrity of the data as it travels across the network. Additionally, ride-hailing services often employ other security measures, such as SSL/TLS for secure communication, authentication mechanisms, and access controls, to further enhance network security and protect both users and the platform from cyber threats.

* 1. **ADMIN CONTROL:**

# Deploying the admin module

The **Admin Control Module** is the central authority module of the sustainable e-commerce platform. It enables the administrator to oversee, manage, and control every component of the system to ensure a secure, functional, and policy-compliant environment. Admins are given elevated permissions to supervise user activities, validate data integrity, maintain platform standards, and enforce sustainability regulations.

* + 1. **ADMIN**

Upon logging in, admins are directed to a powerful dashboard that provides a comprehensive overview of platform metrics and activities. This includes:

* Total number of users categorized by roles (Admins, Sellers, Customers)
* Number of active products and pending product approvals
* Number of orders placed, completed, returned, or cancelled
* Bulk requests and quotes summary
* System logs of user activity and login attempts
* Notifications and messages requiring attention

The dashboard is dynamic, pulling live data from the database and presenting it in a structured and interactive format using tables, cards, and visual summaries.

.

* + 1. **USER MANAGEMENT**

Admins have complete control over all user accounts. They can:

* View, search, and filter user profiles
* Add new users manually if needed
* Update user details (email, status, role)
* Activate, deactivate, or permanently delete accounts
* Monitor login activity and suspicious behaviors
* Promote trusted users (e.g., approving seller registration requests)

Admins are responsible for verifying new seller accounts before they can list products, ensuring all sellers comply with platform guidelines and sustainability goals. Admins can **view detailed user profiles**, including email, registration date, role type, activity status, and login history. A powerful search and filter tool allows the admin to locate specific users quickly by name, role, or account status. New user registrations, especially **seller accounts**, are subject to manual review and verification by the admin to prevent unauthorized or fraudulent listings. In cases of suspicious behavior, inactive accounts, or policy violations, the admin can **temporarily deactivate** or **permanently delete** user accounts. Admins can also **reset passwords**, **update email addresses**, or **promote/demote roles** (e.g., convert a seller to a customer or vice versa). Furthermore, the admin receives real-time notifications about login failures, multiple IP address logins, or flagged users. This enables quick response to potential security issues or rule violations. The User Management system ensures that the platform remains trustworthy, secure, and compliant with sustainability and user conduct policies.

The **User Management** feature under the Admin Control Module is responsible for overseeing the lifecycle and activities of all users registered on the platform. This includes **Admins**, **Sellers**, and **Customers**, each with distinct roles and permissions. The admin has full access to view, monitor, modify, and control these user accounts to ensure smooth and secure operations..

# PRODUCT MODERATION AND APPROVAL

Every product uploaded by a seller may require **admin approval** before going live, depending on platform settings. The admin can:

* View product details including images, materials, certifications, sustainability score
* Approve, reject, or request modifications on product listings
* Flag or remove products that violate terms (e.g., non-sustainable materials, misleading descriptions)
* Monitor the most sold, most returned, or most viewed products

This ensures that only environmentally friendly and accurately described products are available to customers.In terms of **product management**, the admin has authority to approve, reject, or remove product listings. This ensures that only verified, sustainable products are displayed to customers. Admins can also validate product details such as sustainability score, materials used, certifications, and images.The admin can monitor all **orders and returns** across the platform, including disputes or issues raised by users. This helps maintain customer satisfaction and operational fairness. For B2B operations, the admin can oversee **bulk requests and quotations**, ensuring pricing consistency and ethical business practices.

A critical responsibility of the admin is to enforce **sustainability compliance**. They ensure that sellers adhere to eco-certification requirements and that listed products meet environmental standards. Admins can also audit sellers and products periodically.Additionally, the module supports **notifications and communication tools**, allowing admins to send platform-wide announcements, respond to queries, and manage alerts. Through this comprehensive set of functionalities, the Admin Control Module ensures the platform remains secure, transparent, and aligned with its mission of promoting sustainable and responsible commerce. **Management** is another vital function. Admins review every new product listing submitted by sellers before it goes live. This ensures that products meet the platform’s sustainability guidelines, including eco-certifications (e.g., GOTS, FSC, RoHS), material details, sustainability score, and transparency in product descriptions. Admins can approve listings, reject them with feedback, or request corrections. They can also remove any live products that are flagged for violating sustainability standards or misleading consumers. The module also provides powerful tools for **Order and Return Oversight**. Admins can access the entire order database, including detailed records of purchases, shipments, cancellations, and refunds. In cases of disputes between customers and sellers regarding order quality, delivery delays, or return rejections, the admin acts as the final arbitrator. Admins can reverse transactions, manually update order statuses, and initiate corrective actions to ensure customer satisfaction.

The platform’s **Bulk Request and Quotation System** is also managed in part by admins. While sellers typically respond to bulk purchase requests, admins can monitor these high-value interactions, ensure quote legitimacy, and step in to mediate negotiations if needed. They verify pricing consistency, ensure quoted products meet sustainability standards, and check that terms are fair and clear to both parties. A significant responsibility of the admin is ensuring **Sustainability Compliance**. The admin performs audits on sellers and products, checking for valid certification documents, accuracy of sustainability scores, and product traceability. Sellers found to be engaging in “greenwashing” or misleading eco-claims may be warned, suspended, or banned. This oversight preserves the integrity of the platform and builds consumer trust. The Admin Control Module also includes a **Notifications and Communication System**. Admins can send platform-wide announcements—such as feature updates, policy changes, or scheduled maintenance alerts—directly to all users. They can also respond to support tickets and system-generated alerts for flagged activity, including failed login attempts, suspicious transactions, or reported listings. From a technical standpoint, admins have limited backend access for **System Maintenance**, including database health monitoring, schema integrity checks, activity logs, and real-time error reporting. While deep system-level access is restricted to developers or super-admins, standard admins can trigger non-invasive maintenance tasks like clearing outdated logs or initiating platform restarts during critical failures.

# Benefits of Admin module :

One of the main reasons why the Java Server Pages technology has evolved into what it is today and it is still evolving is the overwhelming technical need to simplify application design by separating dynamic content from static

template display data. Another benefit of utilizing JSP is that it allows to more cleanly separate the roles of web application/HTML designer from a software developer. The JSP technology is blessed with a number of exciting benefits, which are chronicled as follows: They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server side dynamic content solutions.

1. The JSP technology is platform independent, in its dynamic web pages, its web servers, and its underlying server components. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web- enabled application server. The JSP pages can be accessed from any web server.
2. The JSP technology emphasizes the use of reusable components. These components can be combined or manipulated towards developing more purposeful components and page design. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA Web Logic and IBM Web Sphere., each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as static documents. A web page can merely display static content and it also lets the user navigate through the content, but a web application provides a more interactive experience. Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server.

They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server-side dynamic content solutions, so before going forward it is very important to know the difference between CGI and the Servlets.Java Servlet is a generic server extension that means a java class can be loaded dynamically to expand the functionality of a server. Servlets are used with web servers and run inside a Java Virtual Machine (JVM) on the server so these are safe and portable. Unlike applets they do not require support for java in the web browser.

Unlike CGI, servlets don't use multiple processes to handle separate request. Servlets can be handled by separate threads within the same process. Servlets are also portable and platform independent. A web server is the combination of computer and the program installed on it. Web server interacts with the client through a web browser. It delivers the [web pages](http://www.roseindia.net/servlets/IntroductionToWebServer.shtml) to the client and to an application by using the web browser and he HTTP protocols respectively. The define the web server as the package of large number of programs installed on a computer connected to Internet or intranet for downloading the requested files using [File Transfer](http://www.roseindia.net/servlets/IntroductionToWebServer.shtml) Protocol, serving e-mail and building and publishing web pages. A web server works on a client server model.

They are mostly used to extend web servers, and are efficient replacement for CGI scripts. CGI was one of the earliest and most prominent server side dynamic content solutions. This definitely reduces development time apart from the At development time, JSPs are very different from Servlets, however, they are precompiled into Servlets at run time and executed by a JSP engine which is installed on a Web-enabled application server such as BEA Web Logic and IBM Web Sphere., each application had its own client program and it worked as a user interface and need to be installed on each user's personal computer. Most web applications use HTML/XHTML that are mostly supported by all the browsers and web pages are displayed to the client as

static documents. A web page can merely displays static content and it also lets the user navigate through the content, but a web application provides a more interactive experience. Any computer running Servlets or JSP needs to have a container. A container is nothing but a piece of software responsible for loading, executing and unloading the Servlets and JSP. While servlets can be used to extend the functionality of any Java- enabled server. That is, JSP pages perform perfectly without any hassle on any platform, run on any web server, and web-enabled application server. The JSP pages can be accessed from any web server.

JSP and Servlets are gaining rapid acceptance as means to provide dynamic content on the Internet. With full access to the Java platform, running from the server in a secure manner, the application possibilities are almost limitless. When JSPs are used with Enterprise JavaBeans technology, e-commerce and database resources can be further enhanced to meet an enterprise's needs for web applications providing secure transactions in an open platform. J2EE technology as a whole makes it easy to develop, deploy and use web server applications instead of mingling with other technologies such as CGI and ASP. Java Server Pages (JSP) technology is the Java platform technology for delivering dynamic content to web clients in a portable, secure and well- defined way. The Java Server Pages specification extends the Java Servlet API to provide web application developers with a robust framework for creating dynamic web content on the server using HTML, and XML templates, and Java code, which is secure, fast, and independent of server platforms.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT CONCLUSION:**

This project presents the successful design and implementation of a full-stack, role-based e-commerce platform exclusively focused on sustainable products. The system integrates three major business models—**Business to Consumer (B2C)**, **Consumer to Business (C2B)**, and **Business to Business (B2B)**—within a unified architecture. Built using **HTML, CSS, Flask (Python), and MySQL**, the platform ensures secure, efficient, and eco-conscious interactions between admins, sellers, and customers.

**FUTURE ENHANCEMENT**

* Implementing a real time payment gateway
* Improving the Mobile application development.
* Implement Real time AI chat bot.

**CHAPTER 7 SAMPLE CODING**

**HTML CODE:**

1. **HOME PAGE:**

{% extends "base.html" %}

{% block title %}Home{% endblock %}

{% block content %}

<div class="container">

    <div class="jumbotron text-center">

        <h1 class="display-4">Welcome to Our E-commerce Store</h1>

        <p class="lead">Discover amazing products at great prices</p>

        <hr class="my-4">

        <p>Browse our collection of products or create an account to start shopping!</p>

        <div class="mt-4">

            <a href="{{ url\_for('products.list\_products') }}" class="btn btn-primary btn-lg">Browse Products</a>

            {% if not current\_user.is\_authenticated %}

                <a href="{{ url\_for('auth.register') }}" class="btn btn-success btn-lg ms-2">Register</a>

                <a href="{{ url\_for('auth.login') }}" class="btn btn-outline-primary btn-lg ms-2">Login</a>

            {% endif %}

        </div>

    </div>

    <div class="row mt-5">

        <div class="col-md-4">

            <div class="card">

                <div class="card-body text-center">

                    <i class="fas fa-shopping-cart fa-3x mb-3"></i>

                    <h3>Easy Shopping</h3>

                    <p>Browse and purchase products with ease</p>

                </div>

            </div>

<div class="col-md-4">

            <div class="card">

                <div class="card-body text-center">

                    <i class="fas fa-truck fa-3x mb-3"></i>

                    <h3>Fast Delivery</h3>

                    <p>Quick and reliable shipping options</p>

                </div>

            </div>

        </div>

        <div class="col-md-4">

            <div class="card">

                <div class="card-body text-center">

                    <i class="fas fa-shield-alt fa-3x mb-3"></i>

                    <h3>Secure Payments</h3>

                    <p>Safe and secure payment processing</p>

                </div>

<div class="hero-section">

    <div class="container">

        <div class="row align-items-center min-vh-100">

            <div class="col-lg-6">

                <div class="hero-content">

                    <h1 class="display-4 mb-4">Welcome to EcoShop</h1>

                    <p class="lead mb-4">Experience sustainability in a whole new dimension</p>

                    <div class="hero-buttons">

                        <a href="{{ url\_for('products.list\_products') }}" class="btn btn-primary btn-lg me-3">

                            <i class="fas fa-store"></i> Shop Now

                        </a>

                        {% if not current\_user.is\_authenticated %}

                            <a href="{{ url\_for('auth.register') }}" class="btn btn-outline-primary btn-lg">

                                <i class="fas fa-user-plus"></i> Join Us

                            </a>

                        {% endif %}

                    </div>

                </div>

            </div>

            <div class="col-lg-6">

                <div id="product-showcase" class="product-showcase"></div>

            </div>

        </div>

    </div>

</div>

<div class="features-section py-5">

    <div class="container">

        <div class="row g-4">

            <div class="col-md-4">

                <div class="feature-card">

                    <div class="feature-icon">

                        <i class="fas fa-leaf"></i>

                    </div>

                    <h3>Sustainable</h3>

                    <p>Eco-friendly products that help protect our planet</p>

                </div>

            </div>

            <div class="col-md-4">

                <div class="feature>

*from* app *import* create\_app

*from* extensions *import* db

*from* models.user *import* User

def **activate\_admin**():

    app = create\_app()

*with* app.app\_context():

*# Find admin user*

        admin = User.query.filter\_by(*email*='admin@example.com').first()

*if* admin:

            print(f"Current admin user:")

            print(f"Username: {admin.username}")

            print(f"Email: {admin.email}")

            print(f"Role: {admin.role}")

            print(f"Is Active: {admin.is\_active}")

*# Activate admin*

            print("\nActivating admin user...")

            admin.is\_active = True

            admin.role = 'admin'  *# Ensure role is set correctly*

            db.session.commit()

            print("Admin user activated!")

            print("\nUpdated admin user:")

            print(f"Username: {admin.username}")

            print(f"Email: {admin.email}")

            print(f"Role: {admin.role}")

            print(f"Is Active: {admin.is\_active}")

*else*:

            print("Admin user not found!")

*if* \_\_name\_\_ == '\_\_main\_\_':

    activate\_admin()

**APP.PY**

*from* flask *import* Flask

*from* config *import* Config

*from* extensions *import* db, login\_manager, mail, migrate

*from* dotenv *import* load\_dotenv

*import* os

*from* flask\_login *import* LoginManager

*from* models.user *import* User

*from* models.product *import* Product

*from* models.cart *import* Cart, CartItem

*from* models.order *import* Order, OrderItem

*from* models.notification *import* Notification

*from* models.requirement *import* Requirement

*from* models.proposal *import* Proposal

*from* models.requirement\_message *import* RequirementMessage

*from* models.quote *import* Quote

*from* models.review *import* Review

*from* models.bulk\_request *import* BulkRequest

*import* sys

*# Load environment variables*

load\_dotenv()

sys.path.append(os.path.abspath(os.path.join(os.path.dirname(\_\_file\_\_), '..')))

def **create\_app**():

*# Initialize Flask app*

    app = Flask(\_\_name\_\_)

*# Load configuration*

    app.config.from\_object('config.Config')

*# Initialize extensions*

    db.init\_app(app)

    migrate.init\_app(app, db)

    mail.init\_app(app)

*# Initialize login manager*

    login\_manager = LoginManager()

    login\_manager.init\_app(app)

    login\_manager.login\_view = 'auth.login'

    login\_manager.login\_message\_category = 'info'

*with* app.app\_context():

*try*:

*# Import base model first*

*from* models.base *import* BaseModel

*# Import models in dependency order*

*from* models.user *import* User

*from* models.product *import* Product

*from* models.cart *import* Cart, CartItem

*from* models.order *import* Order, OrderItem

*from* models.payment *import* Payment

*from* models.requirement *import* Requirement

*from* models.requirement\_message *import* RequirementMessage

*from* models.quote *import* Quote

*from* models.notification *import* Notification

*from* models.review *import* Review

*from* models.bulk\_request *import* BulkRequest

*# Verify database connection*

            db.engine.connect()

            print("✅ Database connection successful")

*# Check existing data*

            user\_count = User.query.count()

            product\_count = Product.query.count()

            order\_count = Order.query.count()

            print(f"\nCurrent Database Status:")

            print(f"Users: {user\_count}")

            print(f"Products: {product\_count}")

            print(f"Orders: {order\_count}")

*except* Exception *as* e:

            print(f"❌ Database connection error: {str(e)}")

*raise* e

*# Import blueprints here to avoid circular imports*

*from* routes.main *import* main\_bp

*from* routes.auth *import* auth\_bp

*from* routes.products *import* products\_bp

*from* routes.cart *import* cart\_bp

*from* routes.order *import* order\_bp

*from* routes.payment *import* payment\_bp

*from* routes.admin *import* admin\_bp

*from* routes.requirement *import* requirement\_bp

*# Register blueprints*

        app.register\_blueprint(main\_bp)

        app.register\_blueprint(auth\_bp)

        app.register\_blueprint(products\_bp)

        app.register\_blueprint(cart\_bp)

        app.register\_blueprint(order\_bp)

        app.register\_blueprint(payment\_bp)

        app.register\_blueprint(admin\_bp)

        app.register\_blueprint(requirement\_bp)

*# Set up user loader*

        @login\_manager.user\_loader

        def **load\_user**(*user\_id*):

*return* User.query.get(int(*user\_id*))

*# Create database tables if they don't exist*

        db.create\_all()

        print("✅ Database tables verified")

*# Create admin user if none exists*

*if* not User.query.filter\_by(*role*='admin').first():

            admin = User.create\_admin(

*username*='admin',

*email*='admin@example.com',

*password*='admin123'

            )

*if* admin:

                print("✅ Admin user created")

*# Create sample products only if no products exist*

*if* Product.query.count() == 0:

            admin = User.query.filter\_by(*role*='admin').first()

*if* admin:

                sample\_products = [

                    {

                        'name': 'Eco-friendly Laptop',

                        'description': 'High-performance laptop with recycled materials and energy-efficient components',

                        'price': 999.99,

                        'category': 'Recycled Electronics',

                        'stock': 10,

                        'admin\_id': admin.id,

                        'sustainability\_score': 4,

                        'materials': 'Recycled aluminum, recycled plastic',

                        'certifications': 'EPEAT Gold',

                        'bulk\_discount': '15% off for orders over 5 units',

                        'condition': 'new'

                    },

                    {

                        'name': 'Bamboo Cutlery Set',

                        'description': 'Reusable bamboo cutlery for eco-conscious dining, includes fork, knife, spoon, and chopsticks',

                        'price': 19.99,

                        'category': 'Reusable Kitchenware',

                        'stock': 50,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Bamboo, stainless steel',

                        'certifications': 'FSC certified',

                        'bulk\_discount': '20% off for orders over 20 sets',

                        'condition': 'new'

                    },

                    {

                        'name': 'Organic Cotton T-Shirt',

                        'description': '100% organic cotton t-shirt, fair trade certified, made with natural dyes',

                        'price': 29.99,

                        'category': 'Reusable Fashion',

                        'stock': 100,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Organic cotton',

                        'certifications': 'Fair Trade Certified, GOTS certified',

                        'bulk\_discount': '25% off for orders over 50 pieces',

                        'condition': 'new'

                    },

                    {

                        'name': 'Solar-Powered LED Lamp',

                        'description': 'Portable solar-powered lamp with 12-hour battery life, perfect for outdoor activities',

                        'price': 39.99,

                        'category': 'Recycled Electronics',

                        'stock': 25,

                        'admin\_id': admin.id,

                        'sustainability\_score': 4,

                        'materials': 'Recycled plastic, solar panels',

                        'certifications': 'RoHS compliant',

                        'bulk\_discount': '15% off for orders over 10 units',

                        'condition': 'new'

                    },

                    {

                        'name': 'Bamboo Toothbrush Set',

                        'description': 'Pack of 4 biodegradable bamboo toothbrushes with BPA-free bristles',

                        'price': 12.99,

                        'category': 'Refillable Personal Care',

                        'stock': 75,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Bamboo, BPA-free bristles',

                        'certifications': 'FSC certified',

                        'bulk\_discount': '30% off for orders over 100 sets',

                        'condition': 'new'

                    },

                    {

                        'name': 'Recycled Glass Water Bottle',

                        'description': 'Elegant water bottle made from recycled glass with protective silicone sleeve',

                        'price': 24.99,

                        'category': 'Reusable Kitchenware',

                        'stock': 40,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Recycled glass, silicone sleeve',

                        'certifications': 'FDA approved',

                        'bulk\_discount': '20% off for orders over 25 units',

                        'condition': 'new'

                    },

                    {

                        'name': 'Hemp Shopping Bag',

                        'description': 'Durable shopping bag made from sustainable hemp with organic cotton lining',

                        'price': 15.99,

                        'category': 'Reusable Fashion',

                        'stock': 60,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Hemp, organic cotton lining',

                        'certifications': 'GOTS certified',

                        'bulk\_discount': '25% off for orders over 30 bags',

                        'condition': 'new'

                    },

                    {

                        'name': 'Beeswax Food Wraps',

                        'description': 'Set of 3 reusable beeswax food wraps, perfect alternative to plastic wrap',

                        'price': 18.99,

                        'category': 'Reusable Kitchenware',

                        'stock': 45,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Organic cotton, beeswax, tree resin',

                        'certifications': 'Organic certified',

                        'bulk\_discount': '20% off for orders over 20 sets',

                        'condition': 'new'

                    },

                    {

                        'name': 'Solar Phone Charger',

                        'description': 'Portable solar charger for smartphones and tablets, with built-in power bank',

                        'price': 49.99,

                        'category': 'Recycled Electronics',

                        'stock': 30,

                        'admin\_id': admin.id,

                        'sustainability\_score': 4,

                        'materials': 'Recycled plastic, solar panels',

                        'certifications': 'RoHS compliant',

                        'bulk\_discount': '15% off for orders over 10 units',

                        'condition': 'new'

                    },

                    {

                        'name': 'Organic Cotton Face Masks',

                        'description': 'Pack of 5 reusable face masks made from organic cotton with filter pocket',

                        'price': 14.99,

                        'category': 'Reusable Fashion',

                        'stock': 200,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Organic cotton',

                        'certifications': 'GOTS certified',

                        'bulk\_discount': '30% off for orders over 50 packs',

                        'condition': 'new'

                    },

                    {

                        'name': 'Bamboo Coffee Cup',

                        'description': 'Insulated bamboo coffee cup with silicone lid, perfect for hot and cold drinks',

                        'price': 16.99,

                        'category': 'Reusable Kitchenware',

                        'stock': 55,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Bamboo, food-grade silicone',

                        'certifications': 'FDA approved',

                        'bulk\_discount': '25% off for orders over 20 units',

                        'condition': 'new'

                    },

                    {

                        'name': 'Recycled Paper Notebook',

                        'description': 'A5 size notebook made from 100% recycled paper with eco-friendly binding',

                        'price': 8.99,

                        'category': 'Office Supplies',

                        'stock': 100,

                        'admin\_id': admin.id,

                        'sustainability\_score': 5,

                        'materials': 'Recycled paper, vegetable-based ink',

                        'certifications': 'FSC certified',

                        'bulk\_discount': '20% off for orders over 50 notebooks',

                        'condition': 'new'

                    }

                ]

*for* product\_data *in* sample\_products:

*try*:

*# Check if product already exists*

                        existing\_product = Product.query.filter\_by(*name*=product\_data['name']).first()

*if* not existing\_product:

                            product = Product(\*\*product\_data)

                            db.session.add(product)

                            db.session.commit()

                            print(f"Created product: {product.name}")

*except* Exception *as* e:

                        db.session.rollback()

                        print(f"Error creating product {product\_data['name']}: {e}")

        print("\nApplication startup completed successfully!")

*return* app

*if* \_\_name\_\_ == '\_\_main\_\_':

    app = create\_app()

    app.run(*debug*=True)

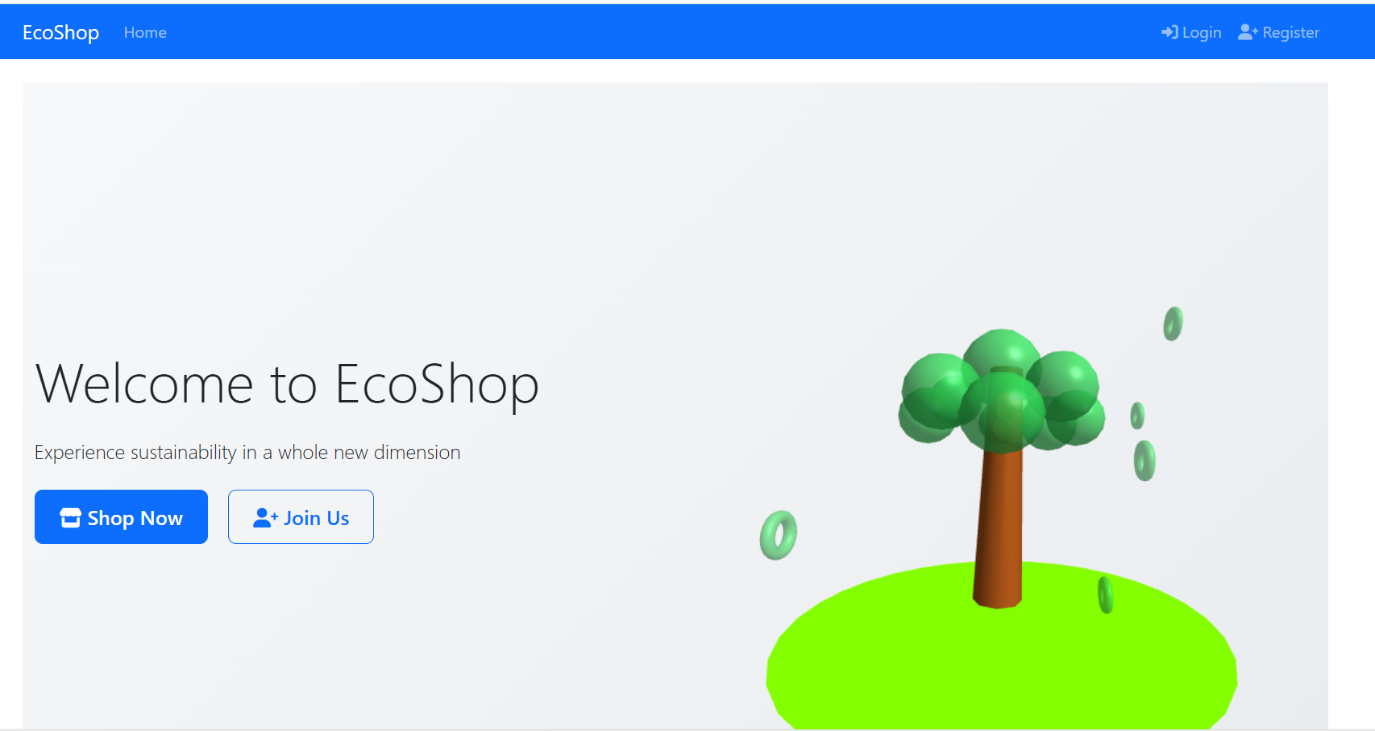
*# Export app for wsgi*

app = create\_app()

**MODULE 1:**

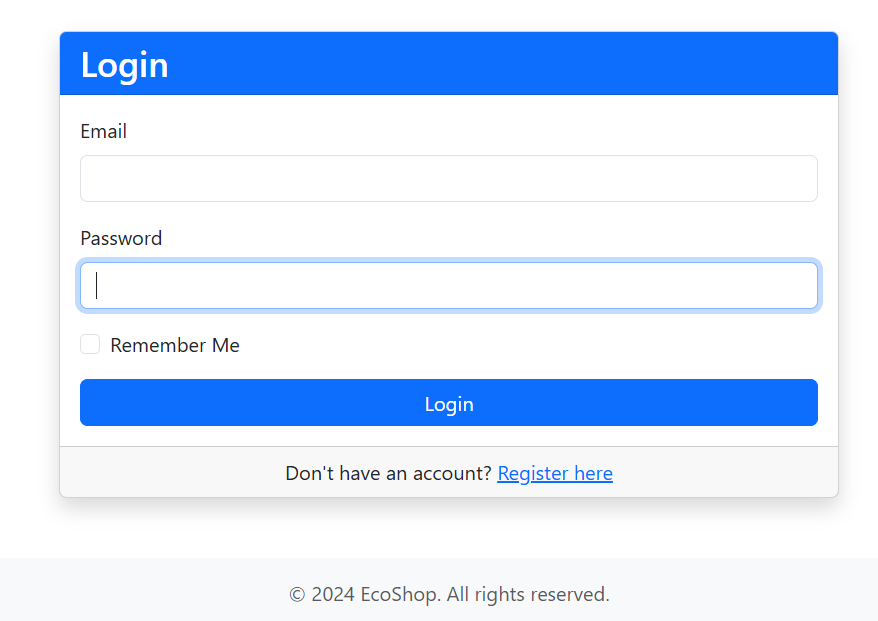
**CHAPTER 8 SCREENSHOTS**

**HOMEPAGE:**

****

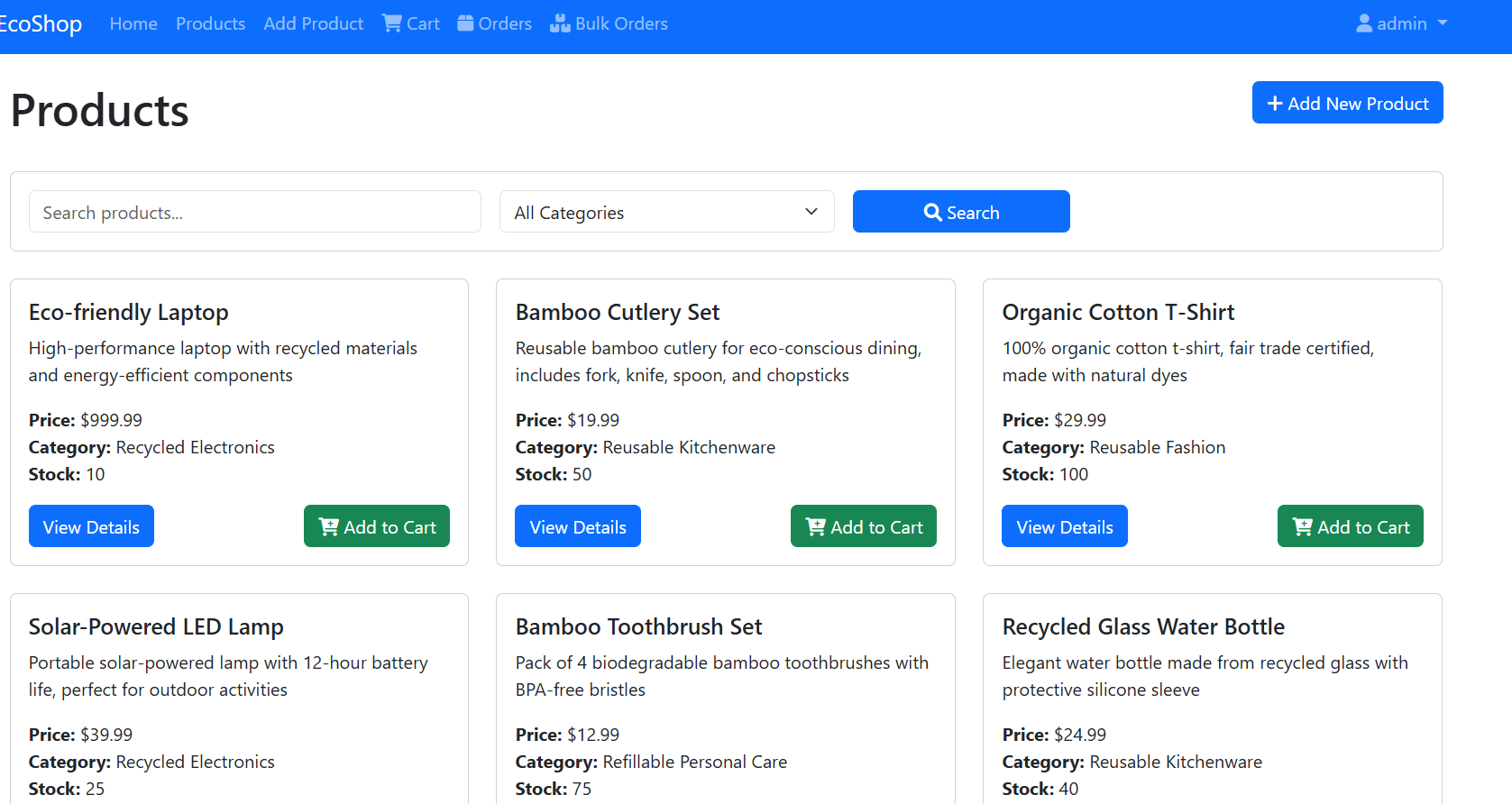
**MODULE 2:**

**LOGIN/SIGNUP PAGE:**

****

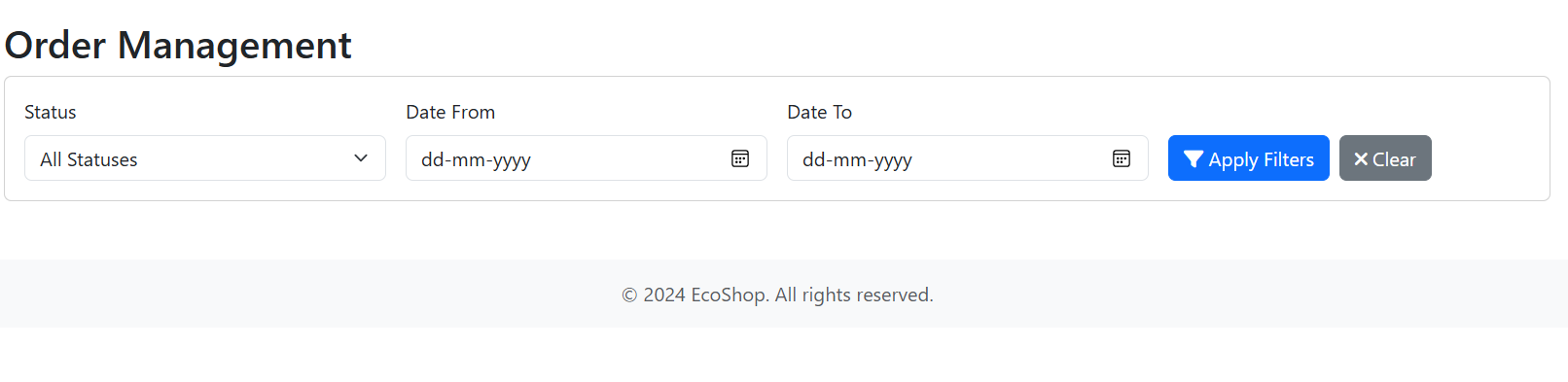
**MODULE 3:**

**PRODUCTS LIST PAGE:**

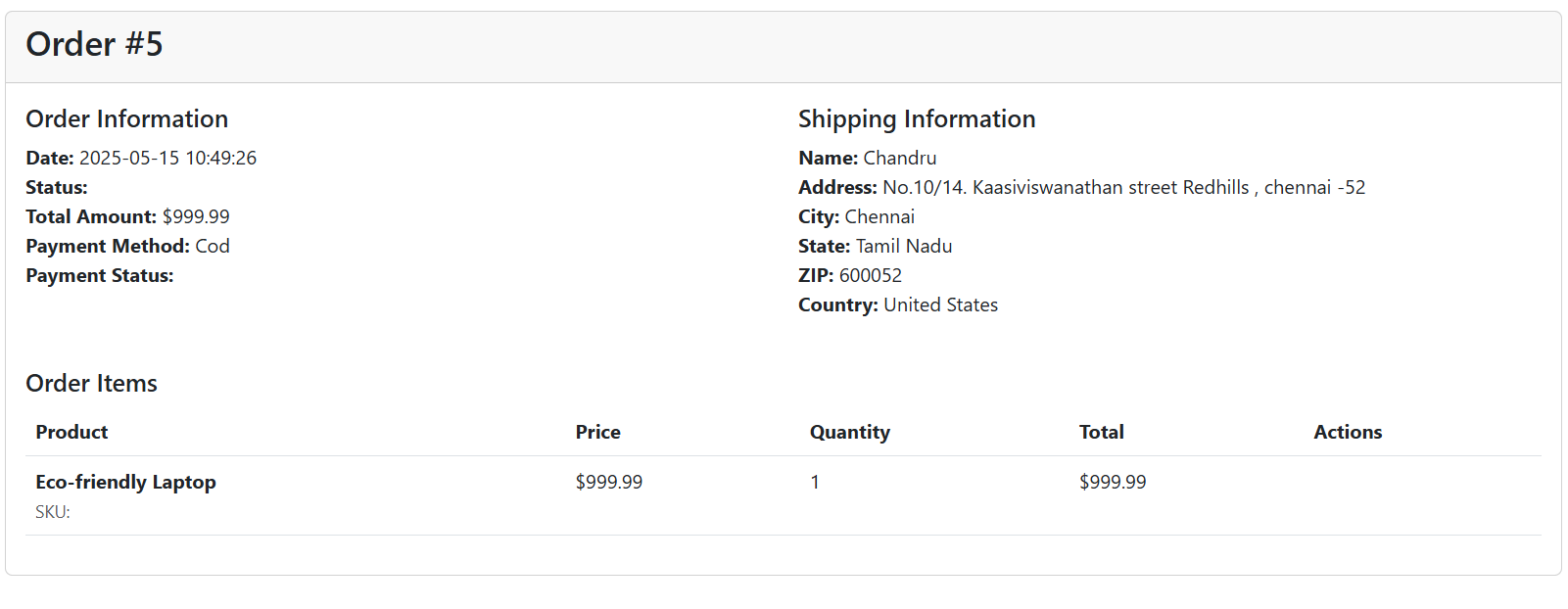
****

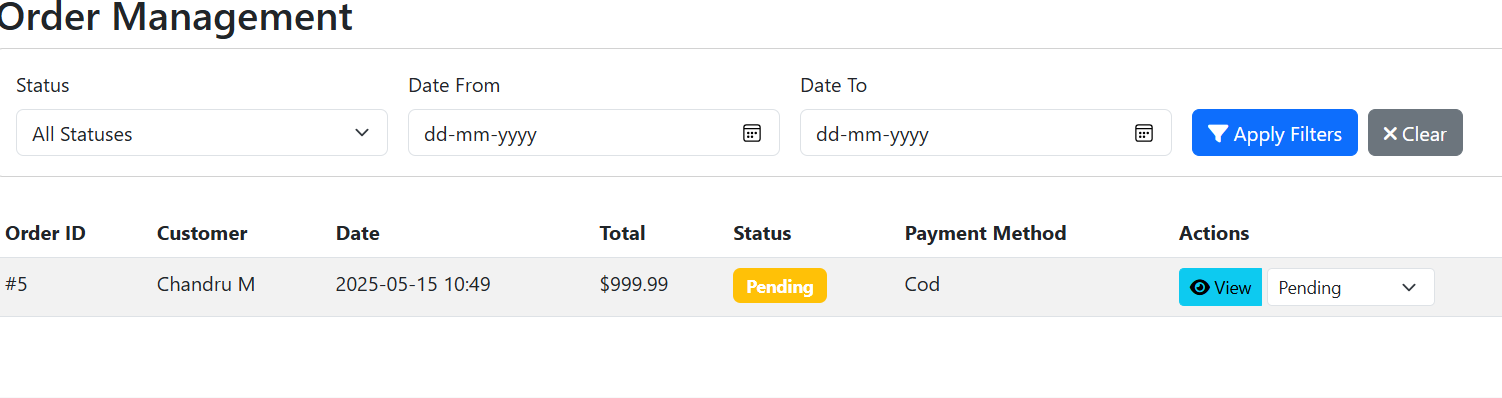
**MODULE 4:**

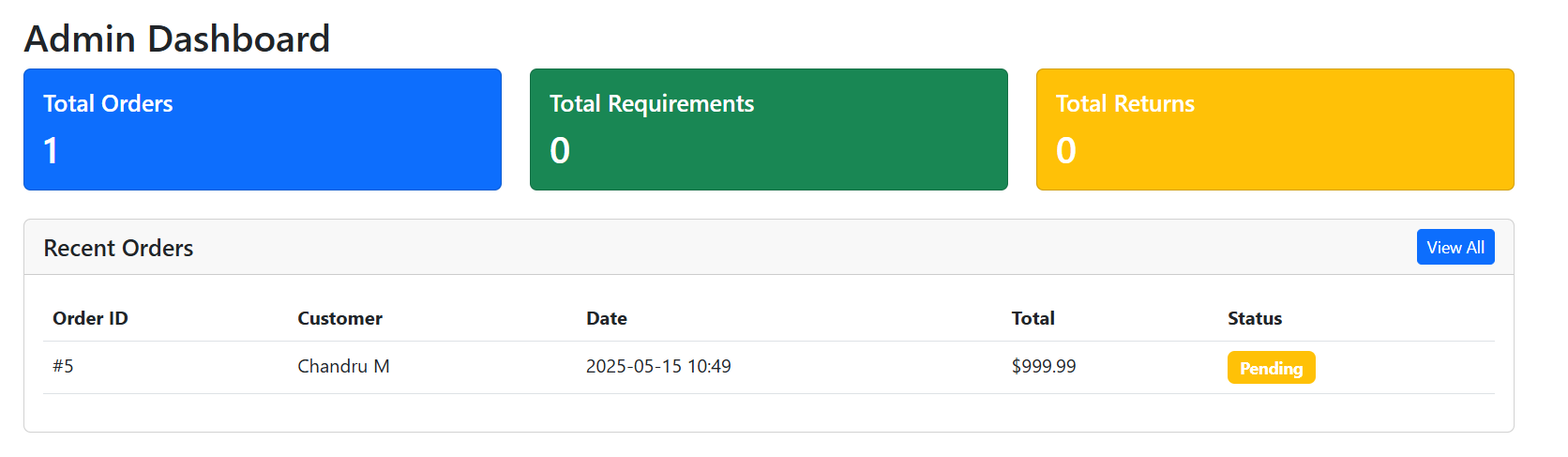
**ORDER MANAGEMENT PAGE:**

****

**OUTPUT(ORDER CONFIRMED PAGE):**

****

****

****

**REFERENCES**

[1]. Amit Kumar and Meena Gupta, *“Challenges in Building Sustainable E-Commerce Platforms”*, International Journal of E-Business Research, Vol. 18, No. 3, pp. 45–57, 2022.

[2]. Shweta Mehta and Ramesh Joshi, *“Business Model Innovation for Sustainability in E-Commerce”*, IEEE Access, Vol. 10, pp. 10254–10266, 2022.

[3]. Miguel Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 2nd Edition, 2018.

[4]. ThredUp and Patagonia, *“ThredUp Resale Report”*, 2021. [Online].

[5].L. Zhang and K. Tan, *“Integrating Circular Economy Principles in E-Commerce Platforms”*, Journal of Cleaner Production, Vol. 290, Article ID 125612, 2021.

[6].Oracle Corporation, *“MySQL 8.0 Reference Manual”*, 2023. [Online].

[7]. Global Organic Textile Standard (GOTS), *“GOTS Certification Overview”*, 2023. [Online].

[8]. Forest Stewardship Council (FSC), *“FSC Certification for Sustainable Forest Products”*, 2022. [Online].

[9]. European Commission, *“Restriction of Hazardous Substances (RoHS) Directive”*, 2021. [Online].

[10]. Electronic Product Environmental Assessment Tool (EPEAT), *“EPEAT Criteria for Green Electronics”*, Global Electronics Council, 2023. [Online]

**GITHUB LINK OF PROJECT**:

CHANDRU M (110521205007): <https://github.com/chandru2621/Integration-of-B2C-C2B-B2B-model-in-ecommerce-website>

DEVIPRIYA J(110521205008): <https://github.com/Devipriya11J/Integration-of-B2C-C2B-B2B-model-in-ecommerce-website->

LOKESH D(110521205304): <https://github.com/Lokesh2004-dj/Integration-of-B2C-C2B-B2B-model-in-ecommerce-website->

