TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

Project Report on

Eco-Bazar : A web portal for Sustainable Choices

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Under the Supervision of

Er. Manish Aryal

Lecturer

Department of Computer Science and Information Technology

Sagarmatha College Of Science and Technology

Sanepa, Lalitpur

Submitted by

Isha Khanal (26404/077)

Samana Silwal (26424/077)

Rewant Shrestha (26419/077)

Sagarmatha College Of Science and Technology

Department of Computer Science and Information Technology

Sanepa, Lalitpur

Submitted to

Tribhuvan University

Institute of Science and Technology

Januaray 2025

SAGARMATHA COLLEGE OF SCIENCE AND TECHNOLOGY

Affiliated to Tribhuvan University

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Isha Khanal, Rewant Shrestha and Samana Silwal entitled "Eco-bazaar - A web portal for sustainable choices" in partial fulfillment of the requirement for the degree of BSc. in Computer Science and Information Technology (BSc. CSIT) be processed for evaluation.

……

Er . Manish Aryal

Supervisor

SAGARMATHA COLLEGE OF SCIENCE AND TECHNOLOGY

Affiliated to Tribhuvan University

CERTIFICATE OF APPROVAL

This is to certify that this project prepared by Isha Khanal, Rewant Shrestha and Samana Silwal entitled "Eco-bazaar – A web portal for sustainable choices" in partial fulfillment of the requirements for the degree of BSc. CSIT has been well studied. In our opinion, it is satisfactory in the scope and quality of the required degree.

……..	……...
Head of Department	Supervisor
Mr. Pratik Timalsena	Er. Manish Aryal
Sagarmatha College of science and Technology	
……..	……...
Internal Examiner	External Examiner
Sagarmatha College of science and Technology	IOST, Tribhuvan University

ACKNOWLEDGMENT

This project has been the result of our efforts. But it wouldn't have been feasible without the assistance and backing of several people and institutions. We want to express our profound gratitude to each and every one of them.

We are really appreciative of Sagarmatha College of Science and Technology for giving us this chance to demonstrate what we have learned through this project. We also appreciate our director's efforts in creating a learning atmosphere that helped this project succeed.

We would like to express our deepest sense of gratitude and sincere thanks to our highly respected supervisor Er .Manish Aryal for his valuable guidance, encouragement and help. His useful suggestions for this project and cooperative behavior are sincerely acknowledged.

At the end, we would like to express our sincere thanks to all the friends and others who helped us directly or indirectly during this project work. This project has been a wonderful experience where we have learnt and experienced many beneficial things.

ABSTRACT

The process of finding eco-friendly products, and sustainable alternatives can often be time consuming and, requiring detailed research to identify truly green options. To simplify and, requiring detailed research to identify truly green options. To simplify this journey, Eco-Bazar has been designed and implemented as a comprehensive web portal for sustainable living. The platform incorporates a recommendations system that uses advanced algorithms to suggest eco-friendly products tailored to the user's click, view and buy with the system analyzing similiarities and patterns to provide personalized suggestions. Eco-Bazaar not only enables users to discover and purchase environmentally responsible products search, detailed information, and a seamless shopping experience, all within an intuitive interface.

The primary goal of Eco-Bazaar is to empower users to make sustainable choices effortlessly, bridging the gap between eco-conscious shoppers and responsibly sourced products. By combining convenience with environmental responsibility, Eco-Bazaar makes it easy to integrate sustainability into everyday life. The system allows users to register and login into the system allowing them to view, search, order the products that have been listed and displayed within a user-friendly interface.

Keywords: recommendation, collaborative filtering algorithms and personalized

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ER	Entity Relationship
DFD	Data Flow Diagram
IEEE	Institute of Electrical and Electronics Engineers
COD	Cash on Delivery
UI	User Interface
SSD	Solid State Drive
SQL	Structured Query Language

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Introduction

Software applications first started getting developed after the creation of fastest digital computers, initial application build works only on standalone desktop machines or local machines as we can say. Until the internet was created in about 1999, the computer network had been their common way of communication. As the quality as well as quantity of web content websites improved, these web applications started becoming more and more interactive, and eventually their functionality evolved beyond offering mere static pages. The significance of web application development rose with the advent of business over the internet. As longtime Web- servers opened everyone now working on the internet at wide-scale, new browsers and development platforms unfold. Web applications become popular when the organizations and business started introducing an application that needs a web browser to use them. E-commerce took off with the advent of the Internet and is blooming till date. Ecommerce websites are the online portals that facilitate online transactions of goods and services through means of the transfer of information and funds over the Internet.

E-commerce was partially e-mail and phone based early, days. It is online and an offline transaction. So single website can be done on the web.

The project 'Eco-bazar' is developed as a web-based application that offers an online service for users looking to purchase eco-friendly tools, equipment and products. The system provides fundamental e-commerce functionalities along with recommendations features.

1.2 Problem Statement

With the increased demand for environmentally friendly items, consumers encounter difficulties in locating credible suppliers and comparing sustainable solutions. The process is often fragmented and time-consuming, with few outlets dedicated specifically to environmentally friendly products. Eco-Bazaar intends to address this issue by providing a single web-based platform that enables consumers to browse, compare, and select the highest rated eco-friendly

products. Eco-Bazar streamlines the decision process and provides individualized recommendations, making shopping easier while promoting sustainable living.

1.3 Objectives

The objective of the project can be stated as:

- To build a recommendation system using Collaborative Filtering Algorithm to recommend construction materials to the user based on their rated products.
- To build a web portal accompanying the recommendation system to make the process of buying the construction material more efficient.

1.4 Scope and Limitation

1.4.1 Scope

The intended system must be created in a way that benefits all parties involved in it, including the end user. Because it is web-based, anyone with a device that can support a browser and an Internet connection will be able to readily use the system.

Eco-bazar as system will allow users to view and search products with their respective information. Products from different category are shown and users will be able to compare and select the products they desire manually. It allows users to create an account that allows them to buy and rate products.

Eco-bazar also allows registered users to get recommendations of related products based on their previous purchases and rating which will help them to find relevant products easily.

1.4.2 Limitations

The different limitations of this project are:

- The system may not be available to users without an Internet connection because it is web-based.
- The recommendation features is limited to the registered users that means only the registered users get recommended products based on their interest and similarity.
- Sustainability is a broad concept, but since Eco-Bazaar focuses on eco- friendly products, some users may not immediately find what they're looking for. However, the platform is designed to expand its offerings and provide personalized recommendations for a tailored experience.

1.5 Development Methodology

This project employs agile approach because of its flexibility, adaptability, and emphasis on providing value to the client in a dynamic and iterative manner. Agile methods strive to deliver small sets of software features to customers as quickly as possible in short iterations. As part of this article scope most commonly used methods will examined from the angle of their applicability, strengths and weaknesses and their adoption in industry. [1]The creation of an e-commerce platform for eco-friendly items necessitates the implementation of numerous complicated features, such as machine learning-based recommendations, incentive systems, and membership plans, all of which must be constantly refined in response to user input and industry trends.

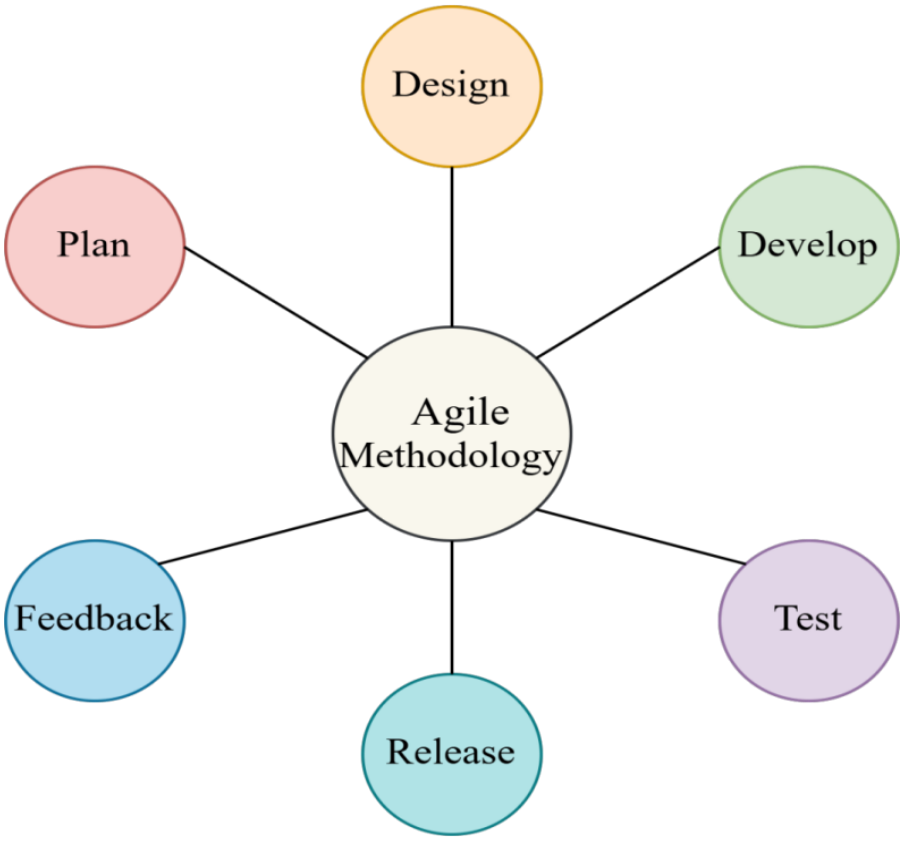


Figure 1 Agile Methodology

Reason for Selection of Agile Methodology as the development methodology

Agile methodology allows the Eco-Bazaar team to break down the project into manageable sprints, each focusing on developing and improving specific features. This iterative approach ensures that the platform can quickly adapt to changing requirements, whether they stem from evolving customer needs, new technological advancements, or shifts in the sustainability landscape. Additionally, Agile promotes collaboration between cross-functional teams, fostering innovation and ensuring that the final product is user-centric, reliable, and aligned with the project's sustainability goals.

1.6 Report Organization

The report has been prepared following the guidelines provided by Tribhuvan University. The report is separated into different chapters. Each chapter consists of various sub chapters with its content. The preliminary section of the report consists of Title Page, Acknowledgement, Abstract, Table of Contents, List of Abbreviations, List of Figures, and List of Tables.
The main report is divided into 6 chapters, which include:

1. Chapter 1: Introduction

   It includes the general overview of the system and the project as a whole. It includes the Problem Statement, Objectives, Scope/Limitations and the Development Methodology for the project and the system being developed.

2. Chapter 2: Background Study and Literature Review

   It includes the study of the current scenario/environment the system will be deployed into. It includes the study of the current trends, preferences of people, the existing systems, areas of improvement among others.

3. Chapter 3: System Analysis

   It includes the requirement and feasibility analysis of the system that can be generated through the studies presented in the previous two chapters. It will also include the Flowchart, ER and DFD for the system which specifies the workflow, entities, attributes and their relationships.

4. Chapter 4: System Design

   It includes the design of the database, forms and interface of the system. It also includes the implementation details of the selected methodology and the details of the algorithm used.

5. Chapter 5: Implementation and Testing

   It includes the details of the different design and development tools used and the implementation details of the modules presented in the form of

code snippets of functions, classes. It also includes the testing of the system with different test cases as per the requirement.

6. Chapter 6: Conclusion and Future Recommendations
   It includes the summary of the system and the project as a whole. It also includes the possibilities/aspects which the system can implement in the future.

The final part of the report consists of References and Appendices. The references are listed in accordance to the IEEE referencing standards and the Appendices includes the screenshots of the system and the major source code snippets.

Chapter 2: Background Study and Literature Review

2.1 Background Study

E-commerce has completely changed how people shop by making a wide range of things easily accessible. But there is an important cost to the environment to this convenience. Environmental degradation is a result of increased transportation emissions from express delivery, excessive packaging use, and the rising amount of product returns that wind up in landfills. Despite these obstacles, the e-commerce industry may be able to reduce its environmental impact with intentional efforts. In order to address these issues, Eco-Bazaar, an online marketplace for environmentally friendly goods, provides a space for customers to make sustainable purchases. Millennials are the generation with the fastest-increasing purchasing power and the most awareness of self-conscious consumption, which suggests that there is a growing demand for sustainable alternatives. By applying environmentally conscious techniques like eco-friendly packaging, efficient refund processes, and effective supply chain management, e-commerce businesses can meet this demand.

There are several key methods which can be utilized to make e-commerce more ecologically friendly. To make sure that a product's impact on the environment and society is taken into consideration at every point of its journey, from production to distribution, sustainable supply chain management is essential. Carbon emissions might be significantly decreased by using eco-friendly shipping methods like mass distribution and reducing the need for faster services. Additionally, providing correct product information helps prevent unwanted returns, and using eco-friendly packaging reduces waste. Reducing the environmental impact is additionally helped by effective return management, which includes enhancing return policies and using customers to avoid returns. By implementing these strategies, its footprint on the environment is reduced and its standing as sustainable and ethical company is enhanced. [2]

2.2 Literature Review

Shopping for eco-friendly products can be a challenging and time-consuming process which requires careful research. One of the most difficult aspects is identifying products that align with both sustainability goals and personal preferences.

While e-commerce has proven to be a profitable and efficient way of doing business, the market for eco-friendly products is still developing. A significant barrier lies in the complexity of sourcing sustainable goods and ensuring transparent supply chains. Additionally, traditional retail habits and lack of awareness have limited the shift toward online platforms specializing in eco-friendly products.

The global pandemic accelerated the adoption of e-commerce across all industries, including niche markets like sustainable goods. As people increasingly turn to online shopping, there has been growing acceptance of platforms offering eco-friendly products. This trend highlights the need for tailored solutions that make sustainable shopping more convenient and accessible.

In Nepal, the e-commerce market is rapidly expanding, with an estimated market value of over 25 million, projected to triple in the near future. Major investments, such as Alibaba's acquisition of daraz, underscore the competitive nature and promising future of the industry. [3] However, the eco-friendly segment of e-commerce in Nepal remains underdeveloped.

Existing platform like Eco-Cart and Earth Hero have introduced online sales for niche markets like sustainable products, but there are few systems dedicated solely to eco-friendly products. Moreover, many platforms lack advanced features such as personalized recommendations, which are key to enhancing user experience and driving customer engagement.

Eco-Bazaar aims to address the gaps by creating an e-commerce platform that combines the strengths of existing systems while introducing innovative

features like a recommendation engine. This feature will provide personalized suggestions based on user preferences (click, view and bought) making it easier for customers to find sustainable products. By integrating user-friendly design with a commitment to environmental responsibility, Eco-Bazaar seeks to transform the way people shop for sustainable goods. [4, p. 4]

Study of Existing System

1. **Eco-Cart**

Eco-Cart is a cutting-edge technology that assists consumers in making environmentally responsible purchasing decisions by estimating and off-setting their carbon footprint. It integrates seamlessly with online retailers, allowing customers to contribute to environmental programs that reduce carbon footprints, making every purchase eco-friendly. [5]

Good Features:

- Carbon Offset at checkout
- Easy Integration

Lacking Features:

- Limited Product Specific Sustainability Insights
- No Personalized Eco Recommendation

2. **Earth Hero**

Earth Hero is an eco-friendly online marketplace that sells a diverse range of sustainable products in categories such as fashion, home goods, and personal care. [6]

Good Features:

- Comprehensive Sustainability Standards
- Wide Product Range

Laking Features:

- Limited Personalized Recommendation
- Interactive Product Education

3. Eco-Mart Nepal

Nepal Eco-Mart Nepal is a local e-commerce portal that promotes sustainable and environmentally friendly items in Nepal. The portal provides a diverse selection of environmentally conscious products, such as organic groceries, reusable household products, and natural personal care items, all obtained from local producers and craftsmen dedicated to sustainability. [7]

Good Features:

- Affordable Eco-Friendly Products
- Locally Sourced Good

Lacking Features:

- Limited Online Presence
- Lack of Sustainability Certifications.

.

Chapter 3: System Analysis

3.1 System Analysis

3.1.1 Requirement Analysis

i. Functional Requirements

Customer

- View and compare products
- Create and edit their own profile
- Search for products
- Create and check orders
- View Recommendations provided

Admin

An admin should be able to perform the following activities within the system:

- Create profiles
- List the products
- Add/Update/Remove category
- Add/Update/Remove products available
- View the customer (end-users) details
- View/Maintain orders

The system offers two distinct user roles: admin and customer, each with specific functionalities. The above points outline the actions that users can perform. The use case diagram below visually represents these roles and their interactions within the system.

.

Use Case Diagram

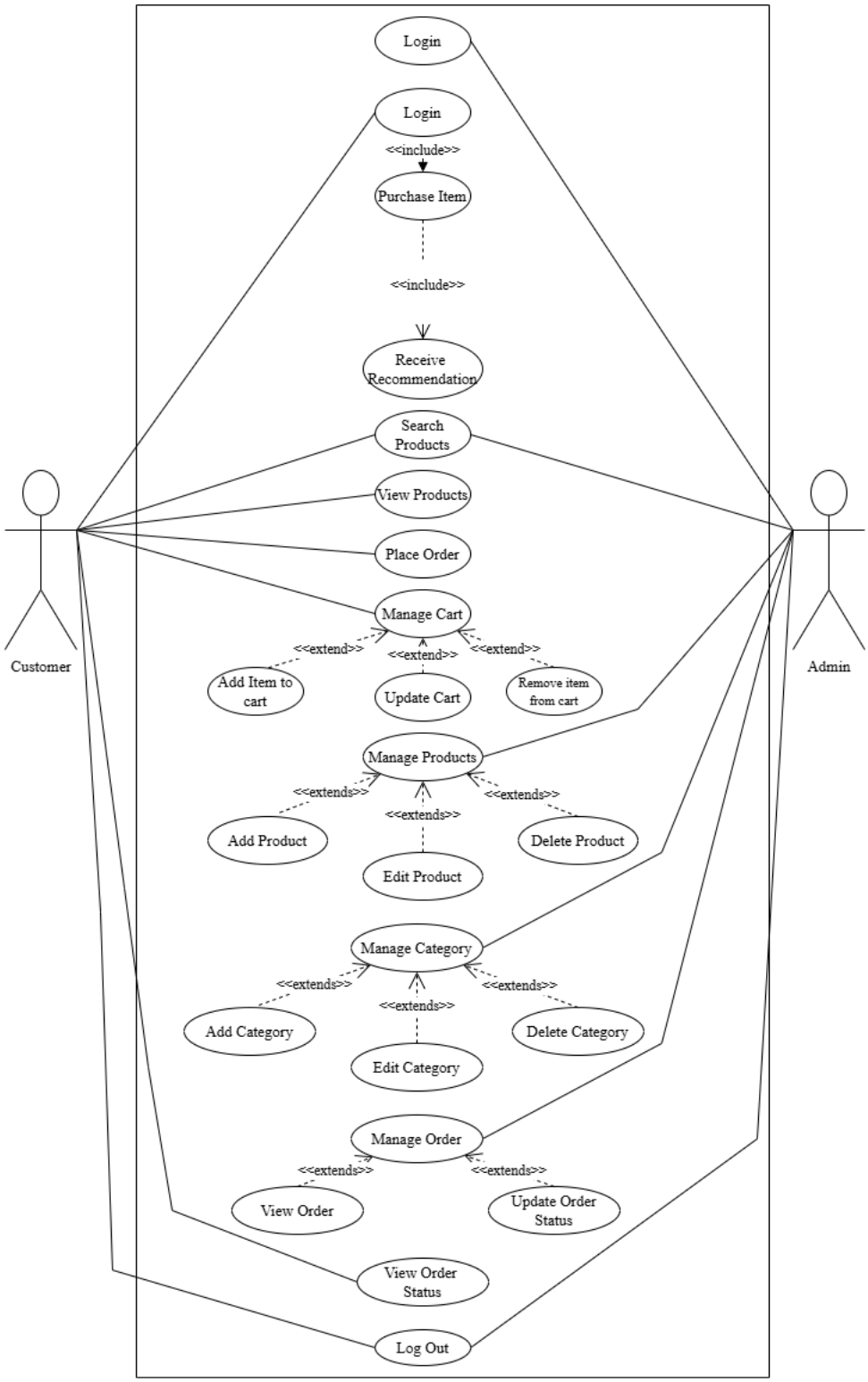


Figure 2 Use Case Diagram

Hardware Requirement

The hardware requirement for the project includes a laptop with support HTML5, CSS, and other implementation tools required for the project. The minimum system requirements for the laptop include i5 13th Gen, 8 GB RAM, 512 SSD, Windows 11 professional. As the final goal the system is to be hosted and be accessible by anyone, we have studied a few hosting providers which include World Link, Web Host with average cost rating from as low as RS. 20.00 to as high as RS. 65,000.00 with its own set of features provided accordingly which will be implemented in the future.

Software Requirement

The software requirements for the project include implementation tools like Visual Studio Code, Figma, SQL Server, all of which are open-source programs so can be utilized in the project without any additional costs.

Draw.io are to be used to prepare the necessary diagrams for the project, both of which are open source so no additional cost is required.

ii. Non-Functional Requirements

- **Performance :** It should be ensured that the system provides fast loading, rides high traffic efficiently , and also ensures excellent management of the database so as to make acess to products and user details speedy.
- **Scalability :** The developed platform has to be such that it scales with growth in a product catalog, user base, and possibly features like international shipping or new product categories.
- **Security :** Observe the best practices in cyber security, secure user authentication and data encryption, protection against SQL injection and XSS. Ensure the protection of users' data according to GDPR.

- **Usability :** Easy to use and intuitive, simple navigation. Many accessibility features should be included for all types of users that may use it, such as disabled persons.
- **Sustainability :** Green servers, powered by renewable energy or running carbon offsetting schemes, host the site, and sustainable web design practices are in place to reduce the carbon footprint of the website.

3.1.2 Feasibility Analysis

A feasibility study, also known as feasibility analysis, is an analysis of the viability of an idea. It describes a preliminary study undertaken to determine and document a project's viability. The results of this analysis are used in making the decision whether to proceed with the project or not. In short, a feasibility analysis evaluates the project's potential for success, following feasibility analysis was performed prior to working on the project.

1. Technical

Eco-Bazaar will be built with dependable, user-friendly, and scalable technologies: HTML, CSS, JavaScript, and Django for the backend, and MySQL. These technologies are well-established with a good history of support, hence the assurance of handling the anticipated load of users without affecting their seamless shopping experience. Integration of real-time data processing, secure payment systems, and other features with personalized recommendations can be achieved through a judicious combination. The scalability and flexibility required as the platform grows will be provided by cloud hosting services.

2. Operational

The Eco-Bazaar will be user-friendly and intuitive, thus requiring minimal training for users and administrators. At the back end, the platform will include an easy-to-use admin panel for managing products, orders, and customer interactions. The introduction of eco-friendly shipping and packaging options goes hand in glove with the platform's goals on sustainability, making this very attractive to the targeted audience.

3. Economic

Eco-Bazaar's development and maintenance costs are within reasonable budgetary limits. The initial investment includes the cost of development, domain registration, hosting, and marketing. These would be offset by the revenue streams that the platform generates through commissions on sales, subscription plans, and advertisement placements to ensure profit occurrence. Thus, Eco-Bazaar would stand in a better position to witness sustainable revenue growth in the long term with the rising consumer demand for eco-friendly products.

4. Schedule (Gantt Chart)

The development timeline for Eco-Bazaar is rather realistic, with a phased approach toward implementation. It aims to complete the core features of user registration, product catalog, search functionality, and secure payments within the first six months. Continuous testing and feedback loops will ensure the platform's launch on time and satisfy users' expectations.

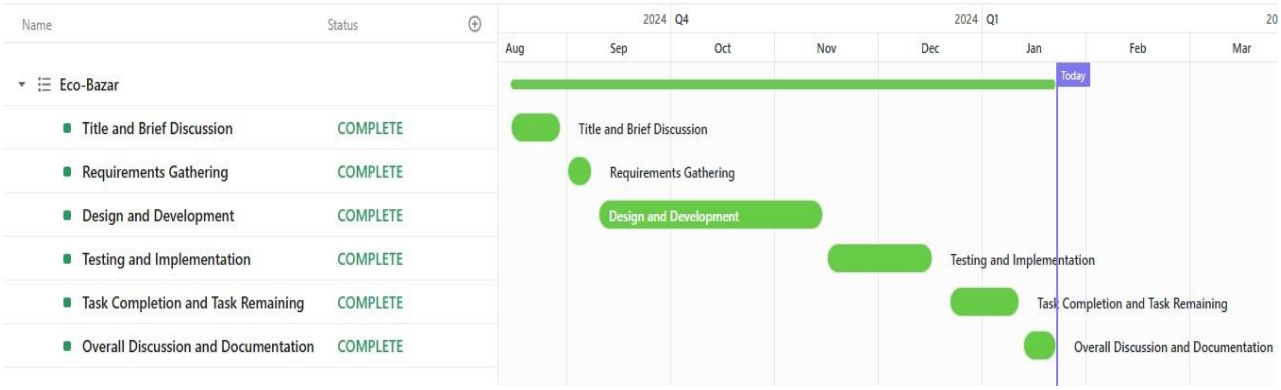


Figure 3 Gantt Chart

3.1.3 Analysis

- **Flowchart**

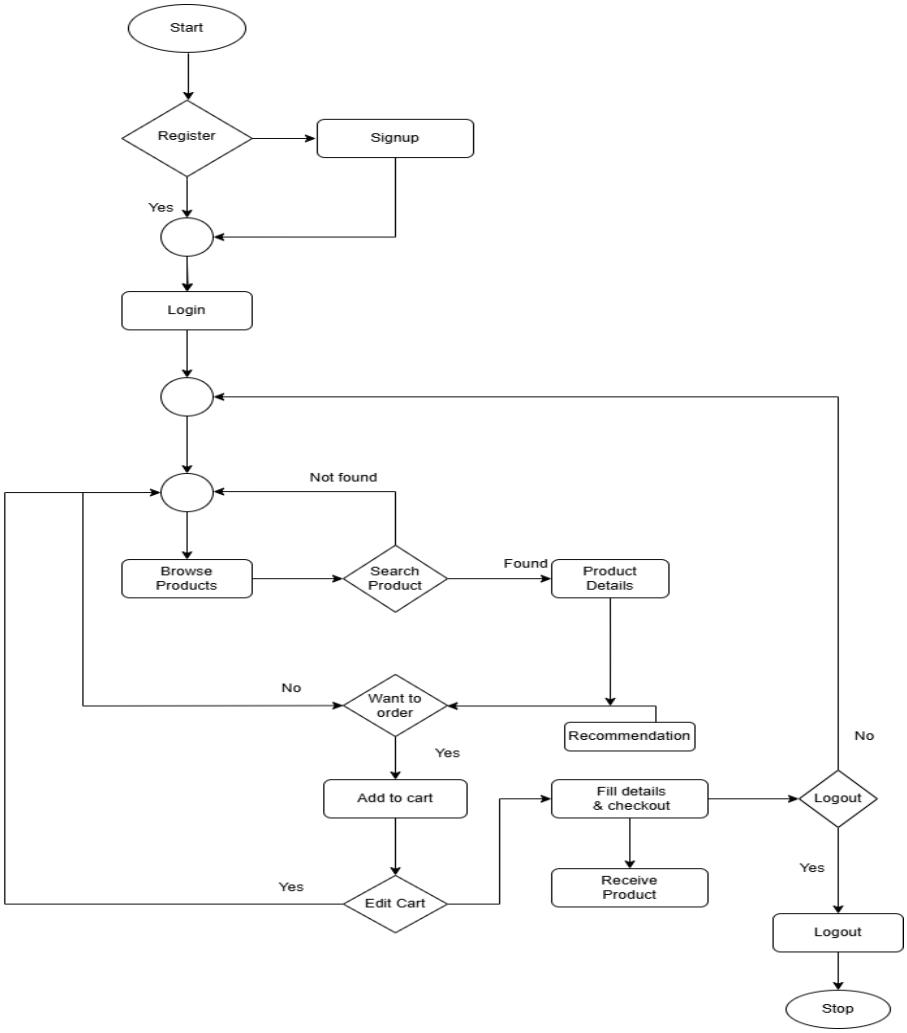


Figure 4 Flowchart for Customer

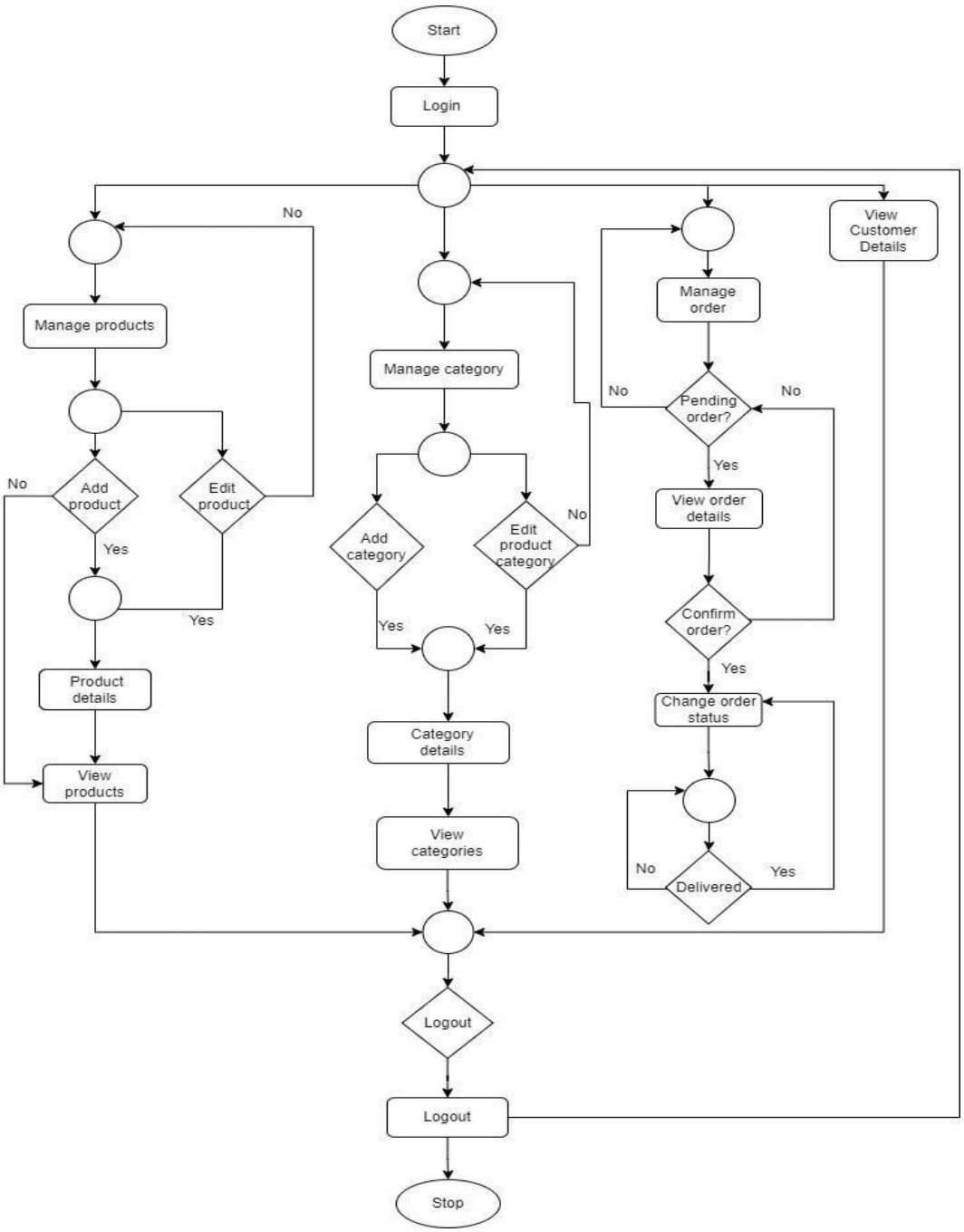


Figure 5 Flowchart for Admin

• **ER Diagram**

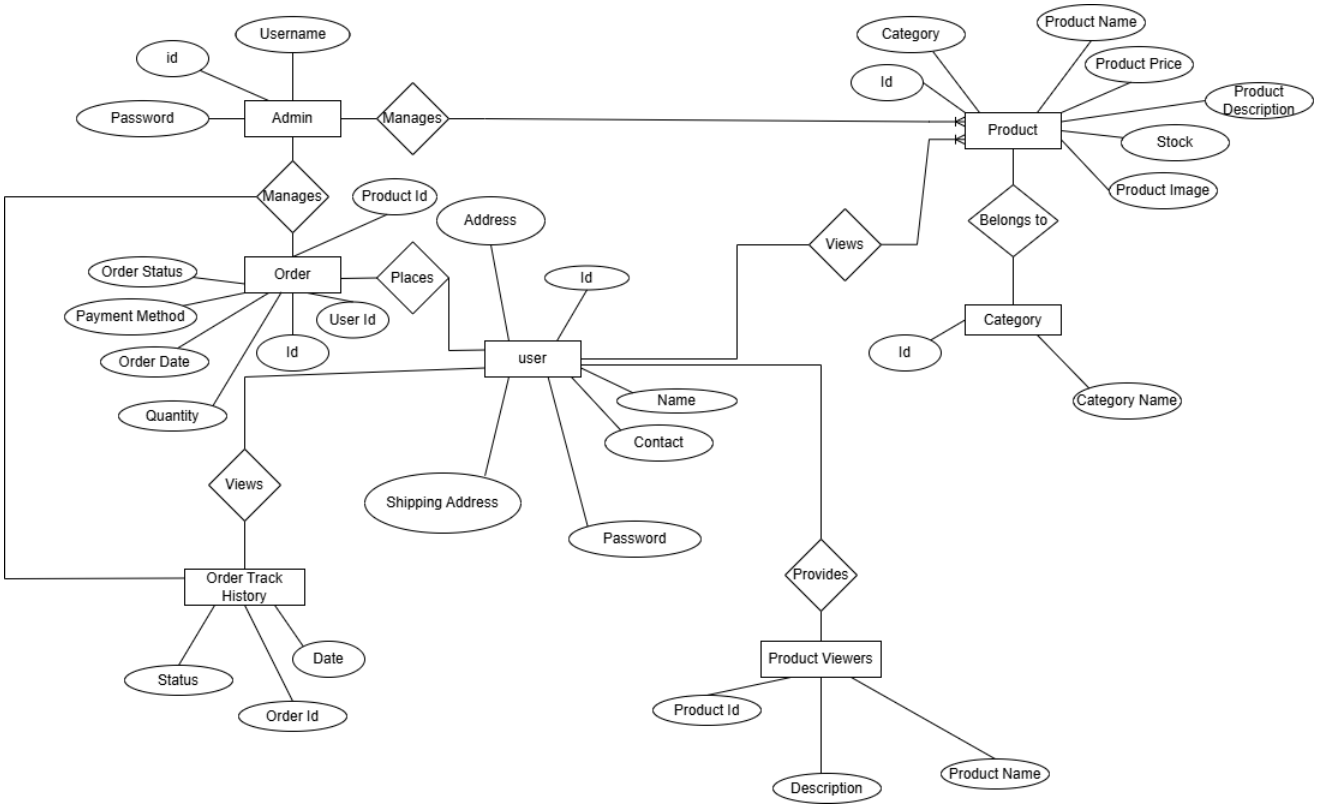


Figure 6 ER Diagram

• **DFD**

The entire system is described in the context-level data flow diagram. Every user module that runs the system is described by the 0 Level DED. The Eco-bazar system's DFD below demonstrates that both the admin and the customer can use the system.

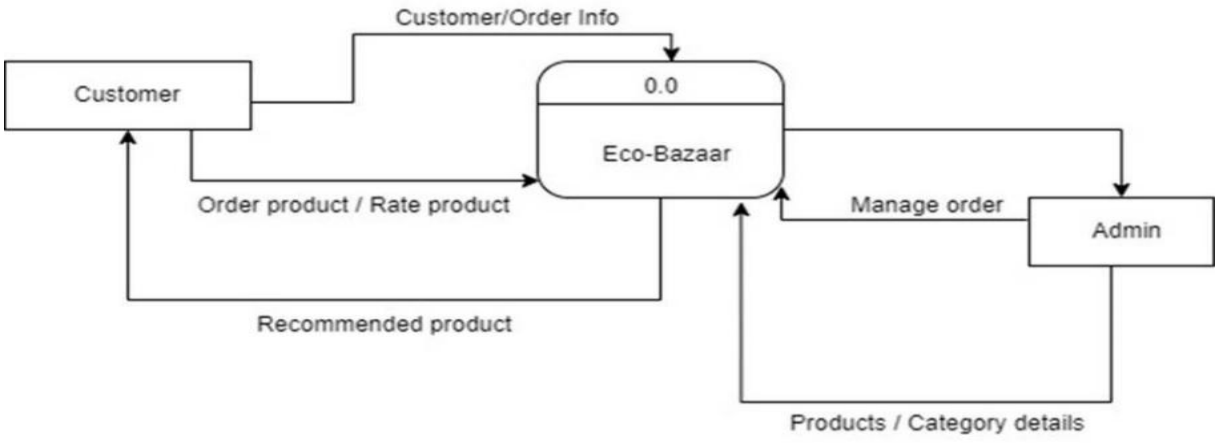


Figure 7 DFD Level 0

The DFD for Admin describe the functionality of Admin, Admin is the owner of the system.

Admin can add category, subcategory for a product and then add products accordingly.

Then the admin can manage the order.

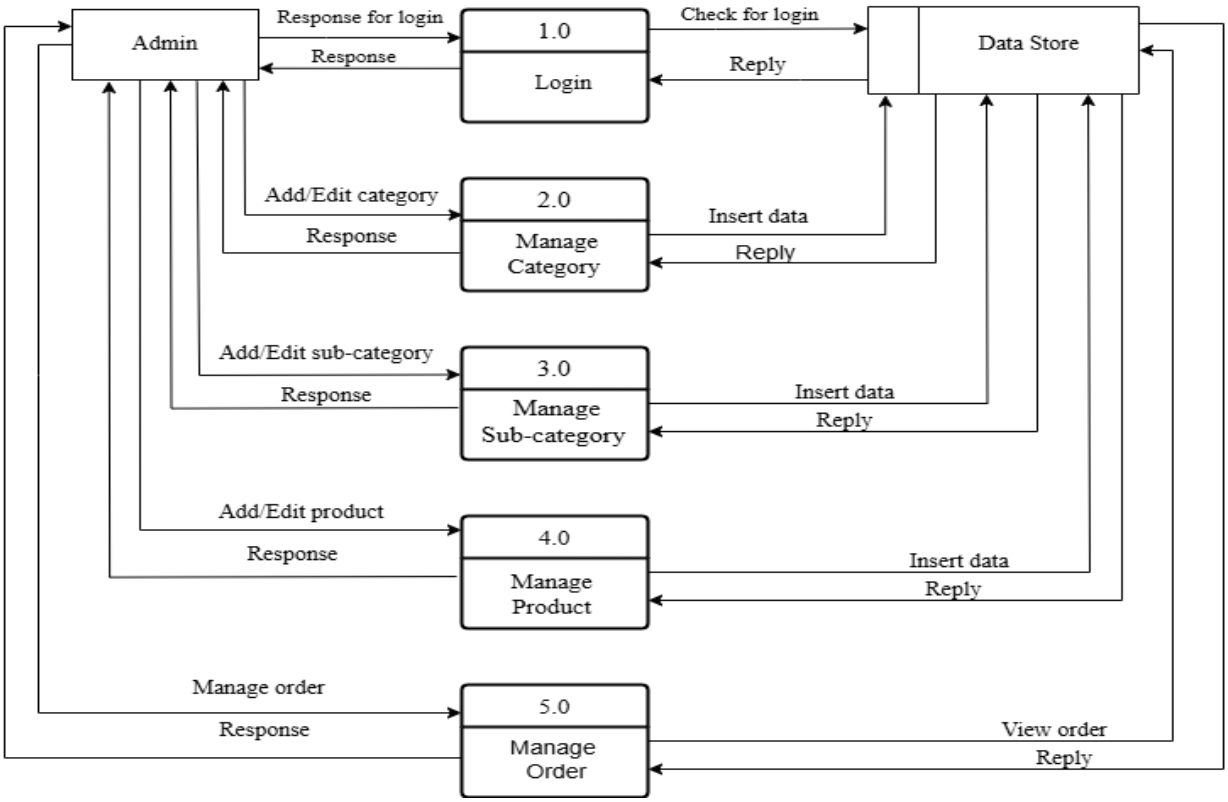


Figure 8 DFD Level 1 for Admin

DFD Level 1 for Customer

Customer include all people who operate or visit our website. The Customer can register in our system and then login. After that they can search for product, select product to buy, place an order from our website. The customer can then rate the purchased product and then view the recommendation generated accordingly.

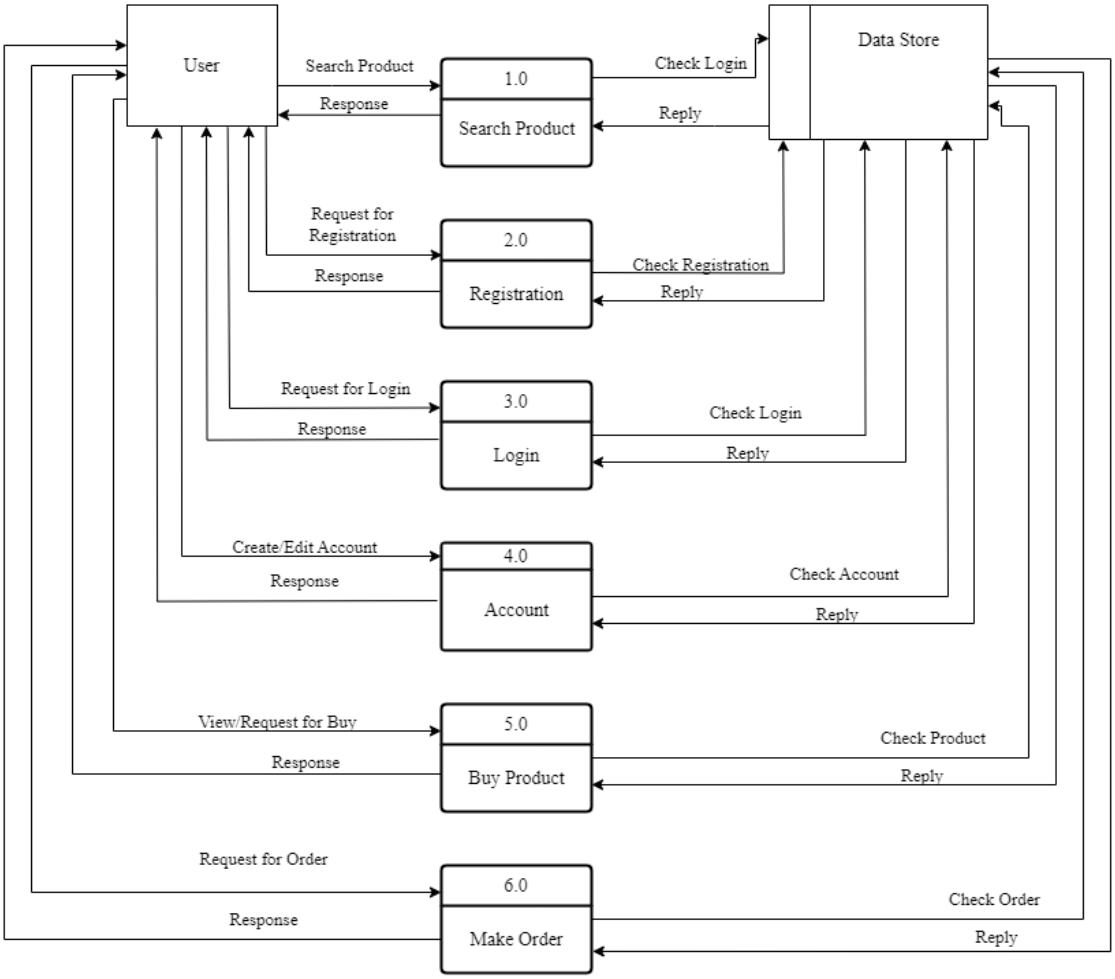


Figure 9 DFD Level 1 for Customer

Chapter 4: System Design

4.1 Design

- **Database Design**

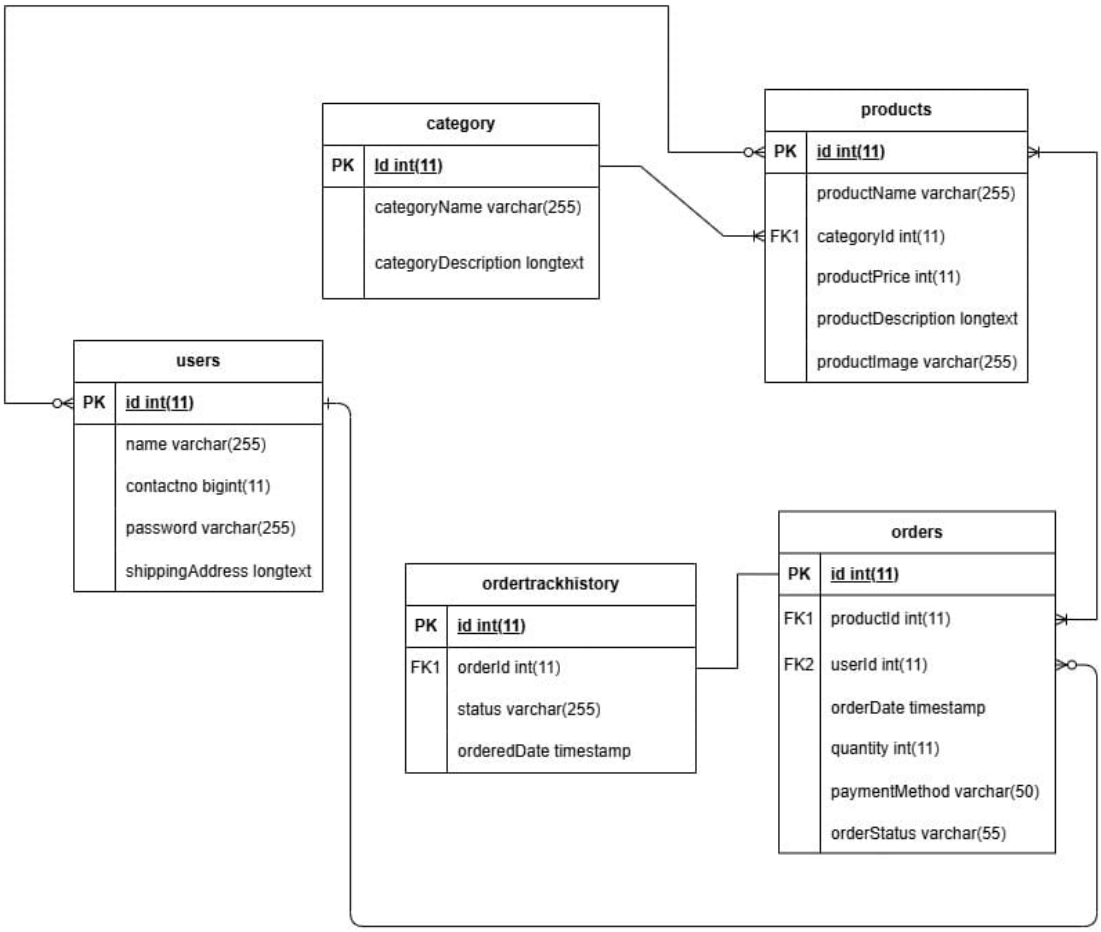


Figure 10 Database Design

- **Form Design**

Forms are an important part of a system as they are used to get data from the users. The different forms that will be required for the system include:

- **In the customer side**
  1. Form for Login & Sign Up
  2. Form for Updating profile/password information
  3. Form for Adding/Updating of Billing/Shipping information
- **In the admin side**
  1. Form for Login
  2. Form for Updating password

3. Form for Adding/Updating
    a. Product
    b. Category

The forms will be created using different elements of the <form>…</form> tag as per requirement and using JavaScript for the Validation purpose.

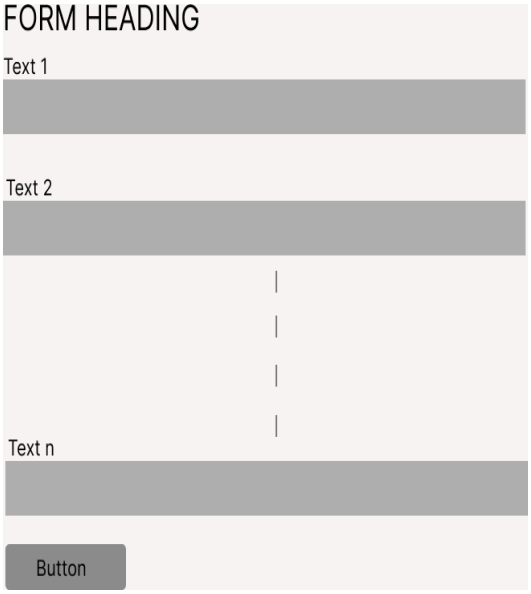


Figure 11 Basic Layout of Forms

- **Interface Design**

One of the most crucial aspects of the system is the user interface (UI/Interface), which establishes how simple it is for a novice user to comprehend the various components presented and move between them to accomplish the system's intended purpose. The following representations will be used to display our system's interface:

Figure 12 Interface of Login for Admin

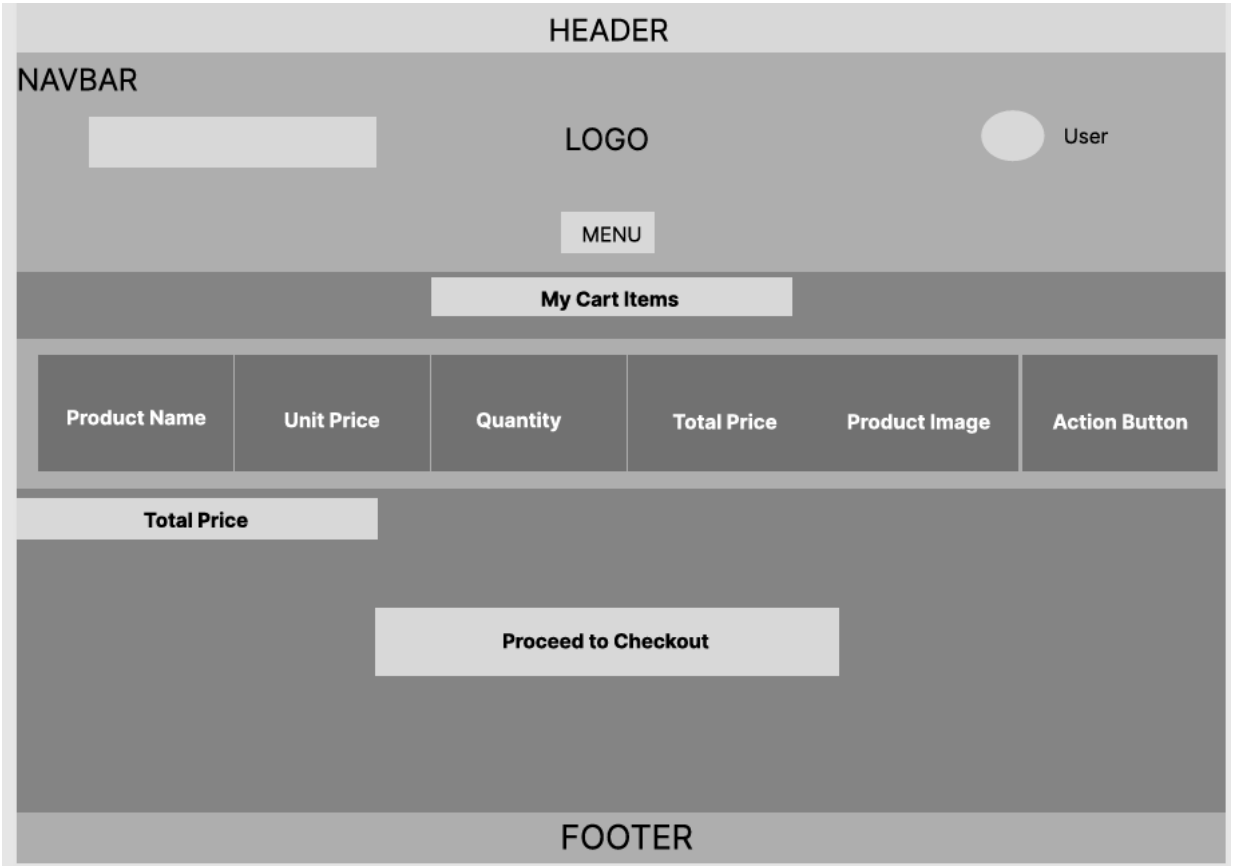


Figure 13: Interface design for cart Items

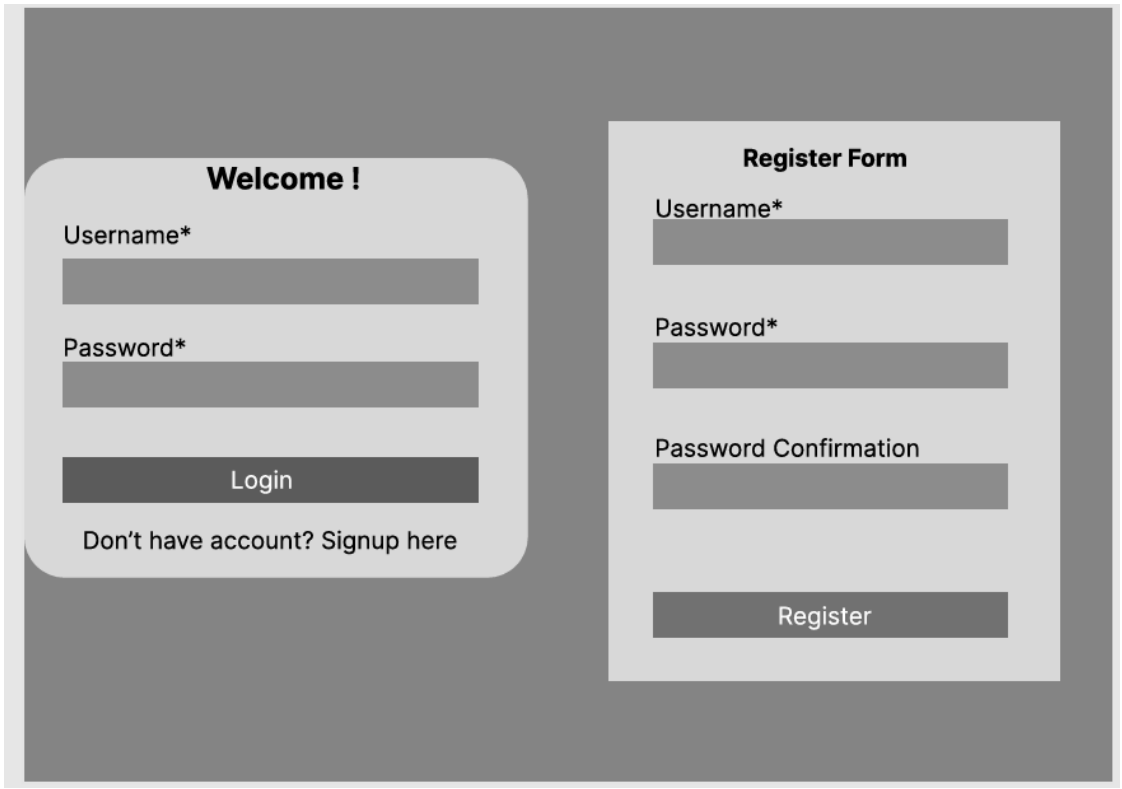


Figure 14 Interface of Login/Sign Up for Customer

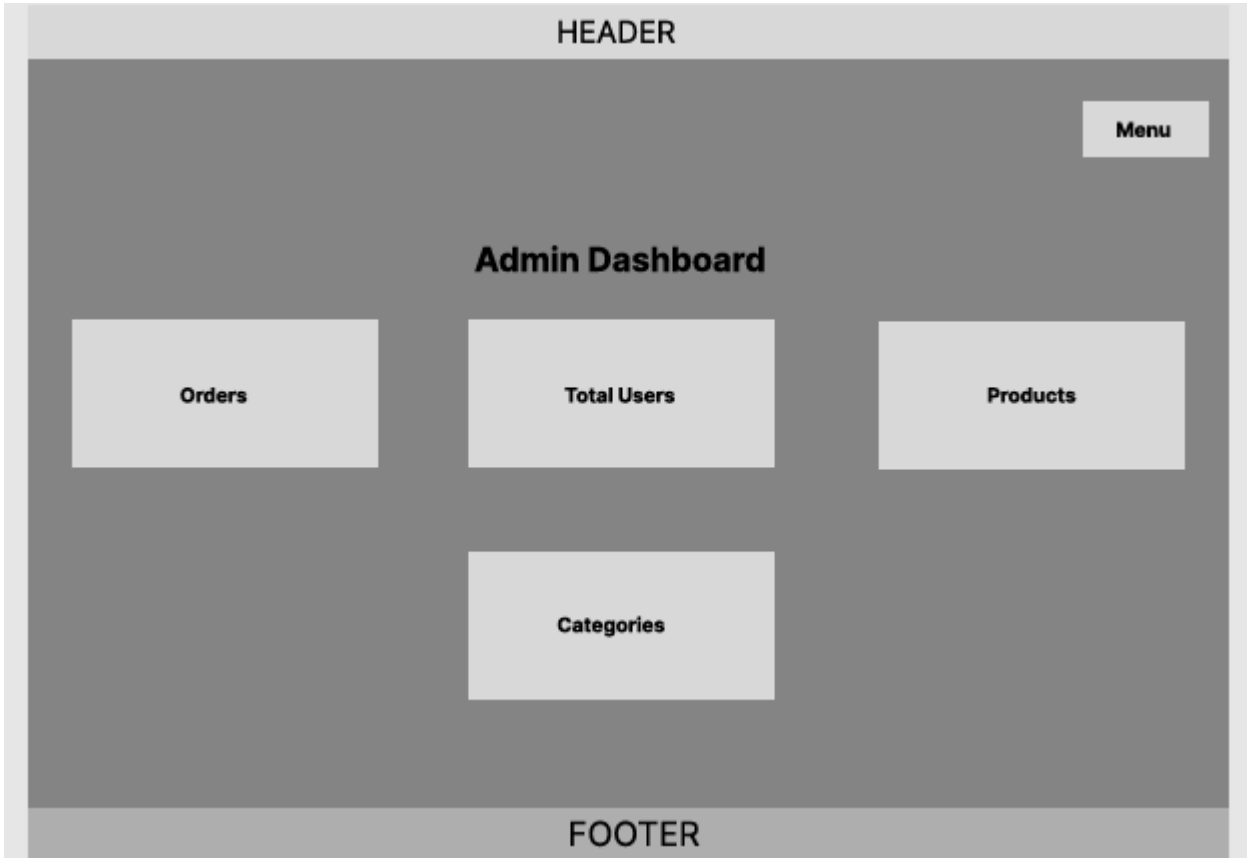


Figure 15 Interface for Admin Dashboard

After finalizing Basic layout .here is the design of our website.

Home Page

Eco-bazar , the home page is as shown below where the user first interacts with the web application after hitting with our URL.

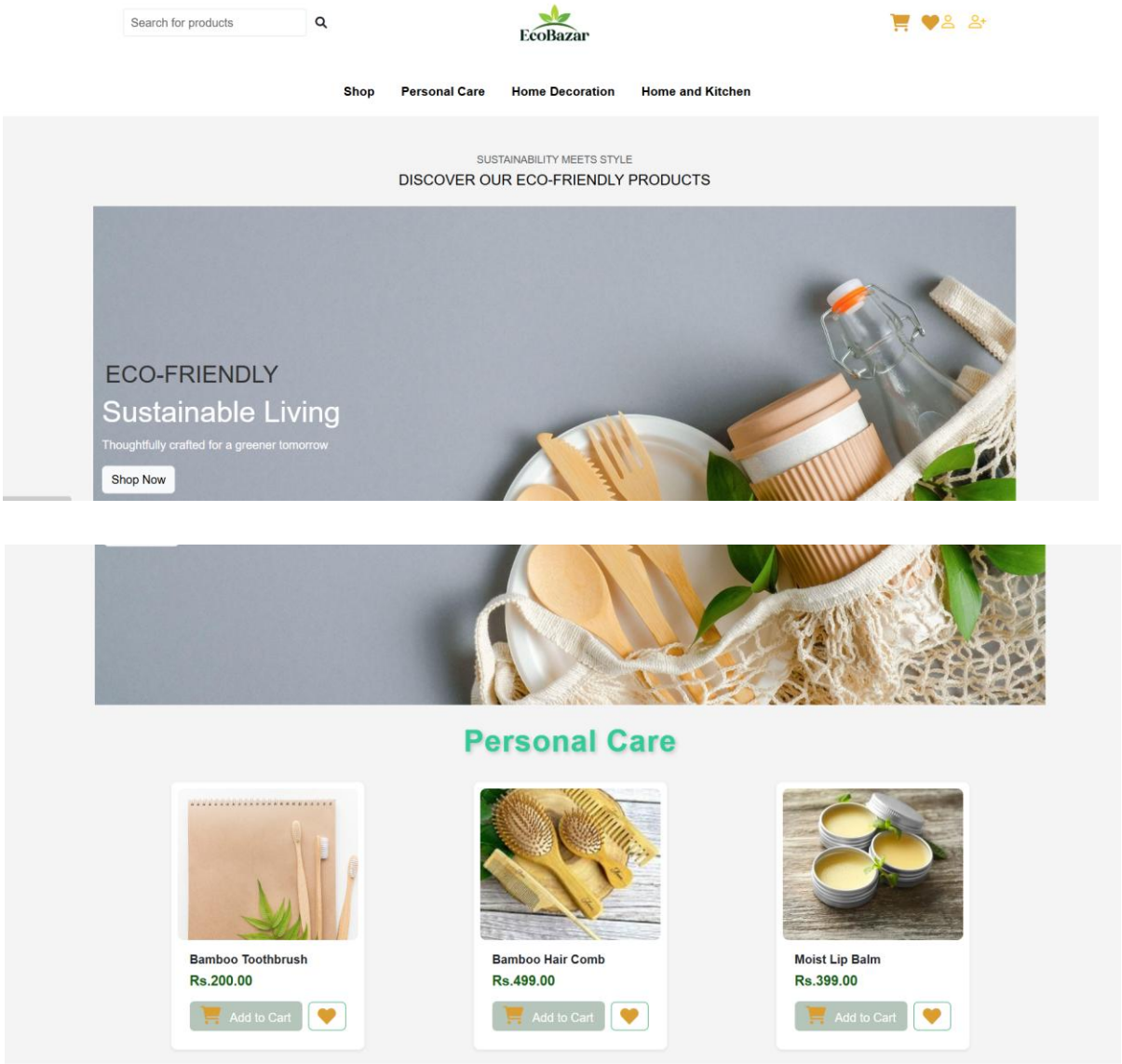


Figure 16 Home page

Login and Registration page

In Eco-Bazaar, users must log in to place orders, manage their accounts, and track purchases. New users can register by filling out a simple form with their name, email, password, and contact details. The login page allows registered users to securely access their accounts and continue shopping. Following screenshots for both login and registration page.

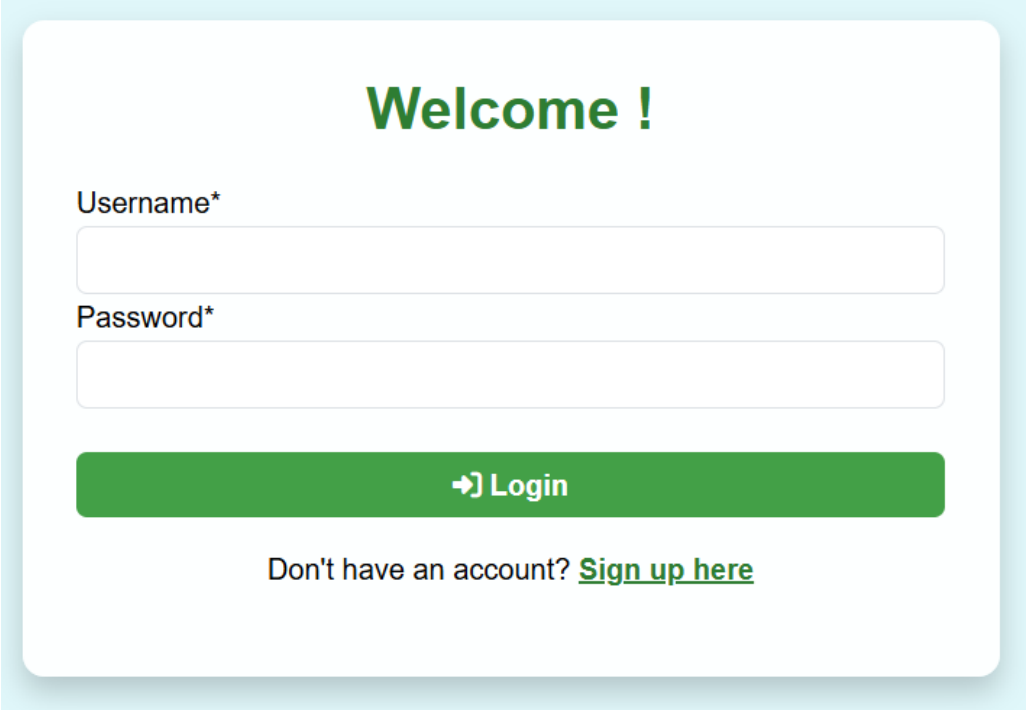


Figure 17 Login page

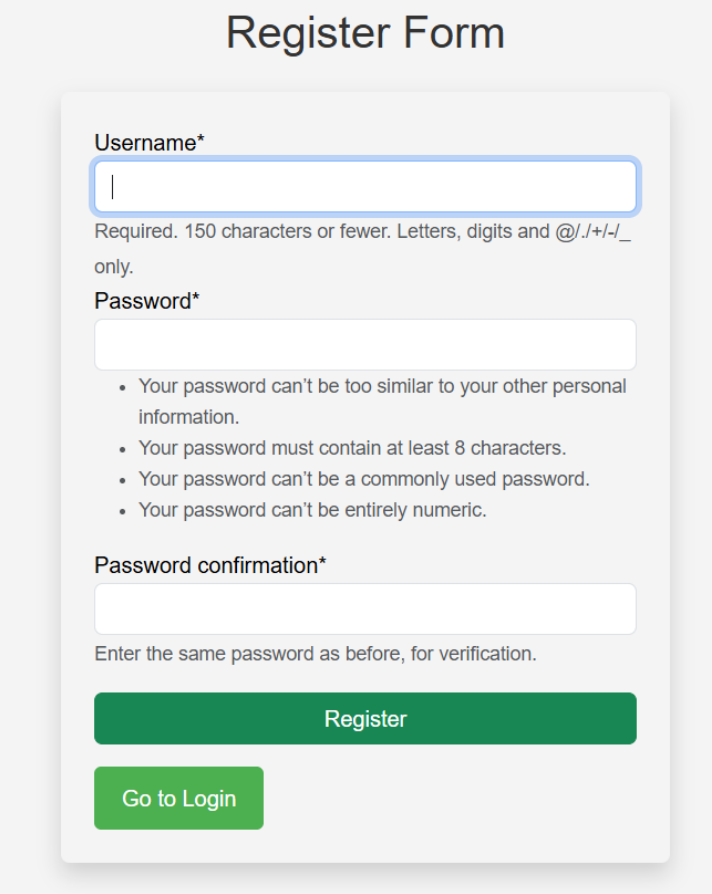


Figure 18 Registration page

4.2 Algorithm Details

Collaborative filtering

One method employed by recommender systems is called collaborative filtering (CF). Using methods that involve cooperation amongst several actors, points of view, data sources, etc., collaborative filtering is the process of searching for information or patterns. Collaborative filtering techniques are usually applied to very large data sets. Collaborative filtering has been used for a wide range of data types, such as financial data, sensing and monitoring data (e.g., in mineral exploration, environmental sensing), electronic commerce, web applications, where user data is the main focus, etc.

[9]

Item Based Collaborative Filtering

Item-item collaborative filtering is a type of recommendation system that is based on the similarity between items calculated using the rating users have given to items. It helps solve issues that user-based collaborative filters suffer from such as when the system has many items with fewer items rated. Many websites use collaborative filtering for building their recommendation system. [10]

Overview of Item Based Collaborative Filtering

Item-based collaborative filtering using cosine similarity is a popular method for generating recommendations by measuring the similarity between items based on user interactions. [8] In this approach, we represent each item as a vector where each dimension corresponds to the actions of different users, such as clicks, views, and purchases. These interactions can be weighted to reflect their significance (e.g., clicks = 1, views = 2, purchases = 3). The similarity between two items is calculated using the cosine similarity formula:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

Where A·B is dot B is the dot product of the two item vectors, A and B, and ||A|| and ||B|| represent the Euclidean norms (magnitudes) of the vectors. The cosine similarity score ranges from -1 to 1, where 1 indicates perfect similarity and 0 means no similarity. A similarity matrix is constructed by calculating the pairwise cosine similarity between every two items. This matrix is then used to recommend items to users. For example, if a user interacts with an item, the system identifies the most similar items (based on the similarity scores in the matrix) and suggests them to the user. This method is particularly effective for generating personalized recommendations, as it adapts to user behavior patterns and can handle sparse datasets efficiently by ignoring zero interactions. Here's a simplified explanation of how Item-Based Collaborative Filtering works in our project:

a) Data Preparation:

- First, the system gathers information on clicks, views, and purchases made by users when they engage with products.

- The user ID, product ID, and interaction type are displayed in the columns of the table in which this data is stored.
- The activities are then translated into numerical values (for example, clicks and views are assigned a value of 1, and purchases are assigned a value of 2).

b) Creating the Interaction Matrix:

- The system builds a table called an "interaction matrix," where each row is a user and each column is a product.
- The numbers in the table show how each user interacted with a product (or 0 if they didn't interact with it).
- This matrix helps in calculating how similar different products are to each other.

c) Calculating Product Similarity:

- To figure out which products are similar to each other, the system uses a method called cosine similarity. This method compares the interaction patterns of each pair of products. Cosine similarity uses this formula:

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

Where:

  - A and B are the interaction patterns for two products.
  - A·B is how much the products overlap in terms of user interactions.
  - ||A|| and ||B|| are the "sizes" of the products' interaction patterns.

d) Generating Product Recommendations:

- After calculating the similarities, the system suggests products to users.
- For each user, the system looks at the products they've already interacted with and calculates how similar other products are to these.
- The system recommends the products with the highest similarity scores to the user, making sure they align with the user's past interests.

e) Showing the Recommendations:

- The system shows the top-N recommended products to the user based on their interaction history and product similarities.

f) Adapting to New Interactions:

- As users continue interacting with more products, the system updates the interaction table and recalculates the recommendations, ensuring that the system stays relevant and adapts to the user's changing preferences over time.

Chapter 5: Implementation and Testing

5.1 Implementation

5.1.1 Tools Used

Design & Development Tools Visual Studio Code

Visual Studio Code is "a free-editor that helps the programmer write code helps in debugging and corrects the code using the intelli-sense method". In normal terms, it facilitates users to write the code in an easy manner. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git makes Visual Studio Code one of the most used code editors. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. [9]

In the project, Visual Studio Code was used as the main code editor to write the codes for the platform. We chose Visual Studio Code as the primary code editor for the project as it is open-source software with a user-friendly interface and support for multiple programming languages and added functionality like debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, embedded Git and so on.

.

Figma

Figma is a design platform for teams who build products together. Born on the Web, Figma helps teams create, share, test, and ship better designs from start to finish. Whether it's consolidating tools, simplifying workflows, or collaborating across teams and time zones, Figma makes the design process faster, more efficient, and fun while keeping everyone on the same page. [10]

Front End Tools

An program that allows direct interaction between users and other applications is referred to as "front-end." It needs a graphical user interface to help staff members without an IT background. It needs to be adaptable and scalable. We created this web application using the following front-end tools:

1. Hypertext Markup Language (HTML)

HTML5 is used as the front end for our project. It is a markup language that is used for structuring and presenting the content on the World Wide Web.

2. Cascading Style Sheets (CSS)

CSS3 is used for presenting our document that is written in HTML5. CSS is intended to enable the separation of presentation and content, along with layout, content, fonts and colors.

3. JavaScript (JS)

JS is used for both backend and frontend. It is characterized by dynamic, prototyped based and multi-paradigm.

Back End Tools

1. Python/Django

The Django web framework is a free, open source framework that can speed up development of a web application being built in the Python programming language.

2. MySQL

The relational database management system (RDBMS) MySQL is open-source. Data types may be associated to one another in one or more data tables that are organized in a relational database; these relations aid in the organization of the data. SQL is a programming language that programmers use to restrict user access to relational databases and to build, edit, and extract data from them.

5.1.2 Implementation Details of Modules

```
df = pd.read_csv(r"C:\Users\Dell\OneDrive\Desktop\git\college project\ecommerce\ecommerce\data\user_product_interactions_no_rating.csv")
action_map = {'click': 1, 'view': 1, 'buy': 2}
df['action_value'] = df['action'].map(action_map)
interaction_matrix = df.pivot_table(index='user_id', columns='Product_id', values='action_value', aggfunc='max', fill_value=0)
product_similarity = cosine_similarity(interaction_matrix.T)
product_similarity_df = pd.DataFrame(product_similarity, index=interaction_matrix.columns, columns=interaction_matrix.columns)
product_similarity_df.to_csv("Matrix.csv")
def recommend_products(user_id, top_n=10):

    user_interactions = interaction_matrix.loc[user_id]
    interacted_products = user_interactions[user_interactions > 0].index.tolist()


    product_scores = {}

    for product in interaction_matrix.columns:
        if product not in interacted_products:

            similarity_score = 0
            for interacted_product in interacted_products:
                similarity_score += product_similarity_df.loc[product, interacted_product] * user_interactions[interacted_product]
            product_scores[product] = similarity_score


    sorted_products = sorted(product_scores.items(), key=lambda x: x[1], reverse=True)

    top_products = [product for product, score in sorted_products[:top_n]]



    return top_products
```

The above code snippet is equivalent to the formula for the calculation of cosine similarity,

that is

$$\text{cosine similarity}(A, B) = \frac{A \cdot B}{||A|| \times ||B||}$$

Where:

- A and B are the interaction patterns for two products.
- A·B is how much the products overlap in terms of user interactions.
- ||A|| and ||B|| are the "sizes" of the products' interaction patterns.

```python
def get_matrix(user_id):
    """
    Returns the interaction matrix for a given user and the similarity matrix for each product from which
    the recommendation score was calculated.
    """

    user_interactions = interaction_matrix.loc[user_id]


    interacted_products = user_interactions[user_interactions > 0].index.tolist()


    product_matrices = {}

    for product in interaction_matrix.columns:
        if product not in interacted_products:
            similarity_matrix = {}
            for interacted_product in interacted_products:
                similarity_matrix[interacted_product] = {
                    "Similarity Score": product_similarity_df.loc[product, interacted_product],
                    "User Interaction Value": user_interactions[interacted_product]
                }
            product_matrices[product] = similarity_matrix

    return user_interactions, interacted_products, product_matrices
```

The above code snippet creates a matrix using the data from the table action in the database.

For the sample data so provided:

user_id	Product_ic	action
User_1	8	click
User_1	7	buy
User_1	19	view
User_1	20	click
User_1	5	buy
User_1	4	click
User_1	17	view
User_1	5	buy
User_1	2	click
User_1	20	buy
User_1	15	click
User_1	7	view
User_1	7	view
User_2	18	buy
User_2	19	click
User_2	9	buy
User_2	10	buy
User_2	11	view
User_2	9	buy
User_3	15	view
User_3	2	buy
User_3	9	click
User_3	3	view
User_3	17	click
User_3	20	buy

Here, we can see that there are 3 users i.e., User 1, User 2, User 3 and products id is shown.

With an assumption that the Users have purchased ,click or view certain set of products, we can see this data in the table to create a similarity score.

The similarity score of the user 7


```
{
  "user_id": "User_7",
  "user_interactions": {
    "1": 0,
    "2": 2,
    "3": 0,
    "4": 0,
    "5": 0,
    "6": 0,
    "7": 1,
    "8": 0,
    "9": 0,
    "10": 0,
    "11": 0,
    "12": 1,
    "13": 0,
    "14": 0,
    "15": 0,
    "16": 0,
    "17": 0,
    "18": 0,
    "19": 1,
    "20": 0
  },
  "interacted_products": [2, 7, 12, 19],
  "product_matrices": {
    "1": {
      "2": {
        "Similarity Score": 0.322396674072133,
        "User Interaction Value": 2
      },
      "7": {
        "Similarity Score": 0.309792381295803,
        "User Interaction Value": 1
      },
      "12": {
        "Similarity Score": 0.35366674532225,
        "User Interaction Value": 1
      },
      "19": {
        "Similarity Score": 0.385399801939445,
        "User Interaction Value": 1
      }
    },
    "3": {
```

```
"3": {
  "2": {
    "Similarity Score": 0.347829063454408,
    "User Interaction Value": 2
  },
  "7": {
    "Similarity Score": 0.319907363448104,
    "User Interaction Value": 1
  },
  "12": {
    "Similarity Score": 0.386363418840244,
    "User Interaction Value": 1
  },
  "19": {
    "Similarity Score": 0.32567556276582,
    "User Interaction Value": 1
  }
},
"4": {
  "2": {
    "Similarity Score": 0.351479745783319,
    "User Interaction Value": 2
  },
  "7": {
    "Similarity Score": 0.315611009345899,
    "User Interaction Value": 1
  },
  "12": {
    "Similarity Score": 0.346013572642267,
    "User Interaction Value": 1
  },
  "19": {
    "Similarity Score": 0.342901858873423,
    "User Interaction Value": 1
  }
},
"5": {
  "2": {
    "Similarity Score": 0.377931753061713,
    "User Interaction Value": 2
  },
  "7": {
    "Similarity Score": 0.323773760938857,
    "User Interaction Value": 1
  },
```


User_7 has interacted with particular products (2, 7, 12, and 19) with varying interaction values in this dataset, which illustrates a recommendation system. A similarity score, which gauges how similar a product is to those the user has engaged with, is used by the algorithm to determine suggestions. The user engagement value indicates the degree to which the user interacted with the products, and each product (1–20) has a similarity score with the products with which they interacted. These scores are used by the recommendation engine to rate and promote new products, giving preference to those that are more like the user's previous interactions.

```python
def index(request):
    categories = Category.objects.all()
    products = Product.objects.all().order_by('-id')
    products_by_category = {}
    for category in categories:

        products_in_category = Product.objects.filter(category=category)[:6]
        products_by_category[category] = products_in_category
    if request.user.is_authenticated:
        user_id = request.user.id
        product_list = products.values_list('id', flat=True)
        top_recommended_items = get_recommendations(user_id, model, product_list)
        recommended_products = Product.objects.filter(id__in=top_recommended_items)
        user_str = f'User_{user_id}'
        idi=recommend_products(user_str)
        int_id = [int(i) for i in idi]
        user_recommendeded_products = Product.objects.filter(id__in=int_id)

        context = {
            'products': products,
            'categories': categories,
            'products_by_category': products_by_category,
            'recommended': recommended_products,
            'user_recommendeded_products' : user_recommendeded_products,
        }

        return render(request, 'users/reccindex.html', context)
    else:
        context = {
            'products': products,
            'categories': categories,
            'products_by_category': products_by_category,

        }
    return render(request, 'users/index.html', context)
```

```
<div class="container-fluid">
  <h1
  class="text-center mt-4"
  style="font-size: 2.5rem; color: #33cc99; font-weight: bold; text-transform: capitalize; text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2); marg
>
  Recommendation Product
</h1>
<div class="row row-cols-1 row-cols-md-3 row-cols-lg-4 g-4">
  {% for item in user_recommendeded_products %}
    <div class="col">
      <div class="card shadow-sm border-light rounded-3 p-2">
        <!-- Product Image Container -->
        <div style="height: 200px; overflow: hidden;">
          <img src="/{{ item.product_image }}"
               class="card-img-top rounded-3"
               alt="{{ item.product_name }}"
               style="width: 100%; height: 100%; object-fit: cover;">
        </div>

        <!-- Card Body -->
        <div class="card-body">
          <!-- Product Name -->
          <h5 class="card-title">
            <a href="/productdetails/{{ item.id }}" class="text-decoration-none text-dark">
              {{ item.product_name }}
            </a>
          </h5>

          <!-- Product Price -->
          <h3 class="text mb-3" style="color: rgb(24, 92, 24); font-weight: bold;">
            Rs.{{ item.product_price }}
          </h3>
```

```
          <!-- Add to Cart Button -->
          <form action="/addtocart/{{ item.id }}/" method="POST" class="d-flex justify-content-between align-items-center">
            {% csrf_token %}
            <button type="submit" class="btn btn-primary flex-grow-1 me-2">
              <i class="fas fa-shopping-cart me-2"></i> Add to Cart
            </button>

            <!-- Wishlist Button -->
            <button type="button" class="btn btn-outline-danger">
              <a href="/add_to_wishlist/{{ item.id }}/" class="text-decoration-none text-danger" title="Add to Wishlist">
                  <i class="fas fa-heart"></i>
              </a>
          </button>
          </form>
          <p class="mt-3">
            <a href="/show" class="text-decoration-none text-info" title="View Recommendation Calculation">
              View how this product was recommended
            </a>
          </p>
        </div>
      </div>
    </div>
  {% endfor %}
</div>
```

The above code snippet corresponds to the condition required to receive recommendation which includes:

1. The user should be registered and logged in .
2. It shows recommendations based on the given data. When a new user (e.g., **User_8**) registers, recommendations are generated based on the data from 500 existing users.

5.2 Testing

5.2.1 Test Cases for Unit Testing

Table 1 Test Table for Register

S.N	Test Inputs	Expected Output	Actual Output	Result
1	Username : user6 Password : password123	Register success	Register success and redirected to login page	Test Successful
2	Username : user1, Password: password345	Register unsuccessful	Username already taken	Test Successful
3	‘ Username : user2 Password: password12345	Register unsuccessful	Email field in required so registration failed	Test Successful

Table 2 Test Table for Login

S.N	Test Inputs	Expected Output	Actual Output	Result
1	Username: user1 Password:password345	Login success	Redirect to home page	Test Successful

2	Username: user2 Password:password123456	Login unsuccessful	Username and password did not match	Test Successful
3	Username: Password:	Login unsuccessful	Login failed with an error message indicating that fields are required.	Test Successful

Table 3 Test Table for cart and payment

S.N	Test Inputs	Expected Output	Actual Output	Result
1	Items are should added in cart and should checkout and went to my orders.	Items should added and checkout and went to my order,	Items should added and checkout and went to my order,	Test Successful
2	Items are shouldn't in cart , it should give the message no item in cart	As result no items in cart	No items in cart	Test Successful

Table 4 Recommendation Based on User Interaction

SN	Test Input	Expected Output	Actual Output	Result
1	User_8 interacts (click, view, or buy) with 20 products	Display products in "Recommended for You" based on similarity scores	Display of recommended products is based on similarity scores	Test Successful
2	User_8 interacts with Product 2, 7, 12, 19	Display products similar to 2, 7, 12, 19 with correct similarity scores	Similar products displayed with correct similarity score calculations	Test Successful
3	User_8 interacts with Product 1, 3, 4	Display products similar to 1, 3, 4 with correct similarity scores	Similar products displayed with appropriate similarity scores	Test Successfu

5.2.2 Test Cases for System Testing

To test the proper functioning of the system, we analyze the recommendation results among 500 existing users. User_8 has interacted (clicked, bought, or viewed) with 20 products, and the system uses this interaction data to generate recommendations. The recommendations are based on the similarity scores between products and the behavior of other users. Since no new user is created, the system continues to provide recommendations by analyzing existing data patterns from the 500 users.

5.3 Result Analysis

The recommendation system generates suggestions based on the interaction data of 500 existing users. In this case, User_8 has interacted (clicked, bought, or viewed) with 20 products. The system analyzes these interactions and calculates similarity scores with other products. Based on these scores and user behavior patterns, the system provides recommendations. The results confirm that the "Recommended for You" section effectively suggests products that align with the user's past interactions, ensuring accurate and relevant recommendations.

Chapter 6: Conclusion and Future Recommendations

6.1 Conclusion

The system, Eco-Bazaar – A web portal for sustainable choices with a recommendation system based on collaborative filtering algorithm was implemented successfully. After doing background study and literature review of similar systems, we decided to integrate recommendation features in our system to make the overall e-commerce site experience better. Agile methodology was the selected development methodology as the initial requirement was concise and clear which helps to build the system. The development was further proceeded with multiple iteration (these are sprints) while making necessary changes till the intended system was developed. Testing was done in each iteration based on the requirement. Later on, unit testing was done on each component of the system. Using this system, the end user would be able to explore their sustainable products needs within a single platform with added feature of recommended. The recommendation is done by analyzing the interaction of multiple users on the same product using item-based collabartive algorithm using cosine similarity and calculating the prediction based on the similarity for different products to recommend. This ensures that the platform remains reliable and continues to meet user expectations.

6.2 Future Recommendations

As we have implemented collaborative filtering algorithm for the recommendation, it is recommended to use a large data set to make the recommendation for greater accuracy. In future, Eco-bazar will introduce a membership reward points system along with refer and earn program. Customers will earn reward points based on their purchases and can redeem them for discounts on future orders. Additionally, by referring friends to the platform, users can earn extra points, promoting both customer engagement and

eco-friendly shopping. This initiative will further enhance the shopping experience while encouraging sustainable practices within the community.

References

[1] C. M. Yassin, "Integrate and Apply the Recommendation System of Agile Methods," 2020.

[2] N. Kitukutha, "Sustainable e-commerce and environmental impact on sustainability".

[3] "S. Dyola, "E-COMMERCE TRENDS IN NEPAL AND ITS CHALLENGES".".

[4] M. R. R. Mehta, "An Analytical Study On E-Commerce For Sustainable".

[5] http://ecocart.io/.

[6] https://www.earthhero.org/.

[7] https://ecomart-hackathon.netlify.app/.

[8] https://en.wikipedia.org/wiki/Item-item_collaborative.

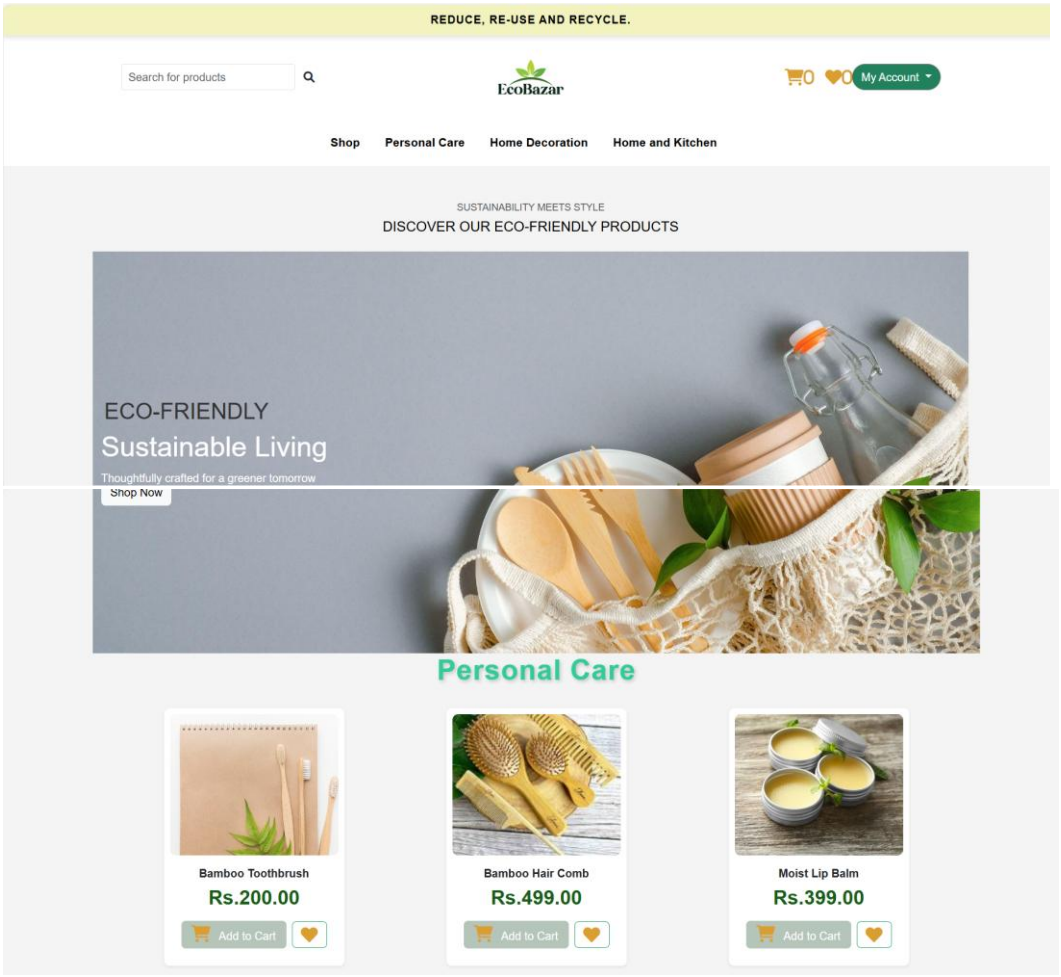
[9] https://en.wikipedia.org/wiki/Visual_Studio_Code.

[10] https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma.

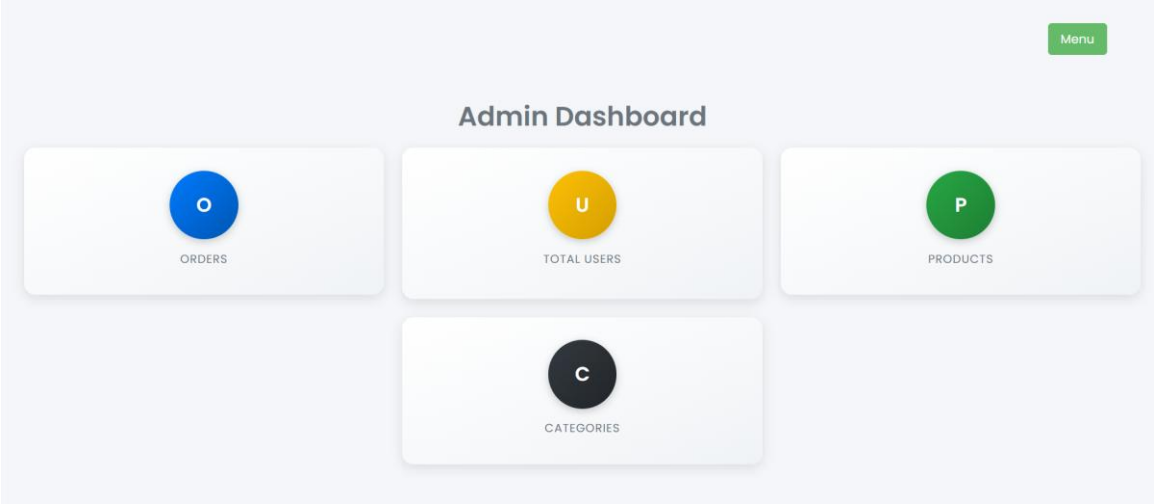
[11]

Appendices

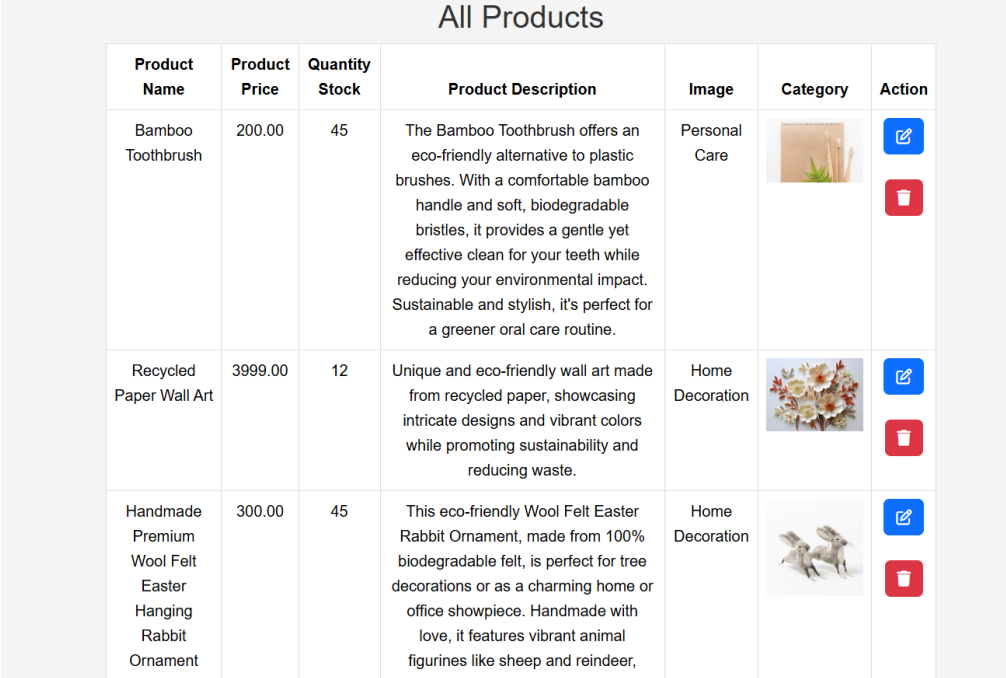
Some of the screenshots of the system are as follows:



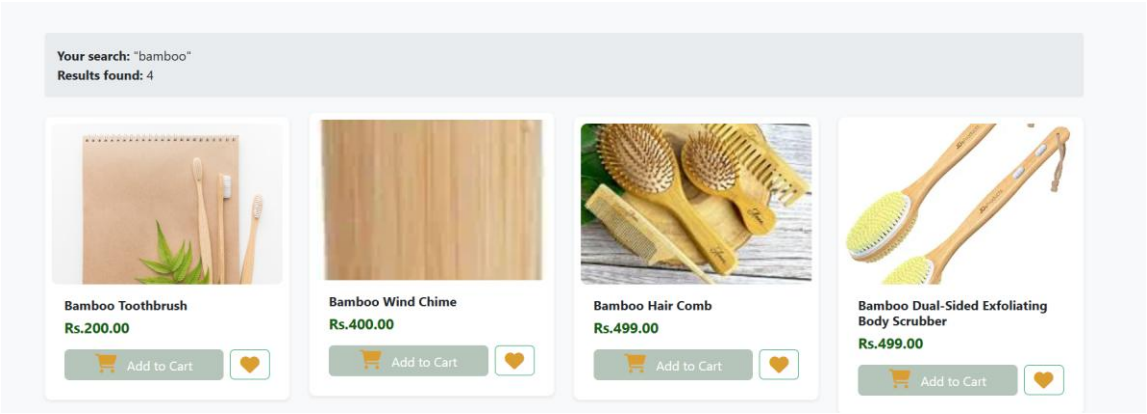
Screenshot 1 UI of the system



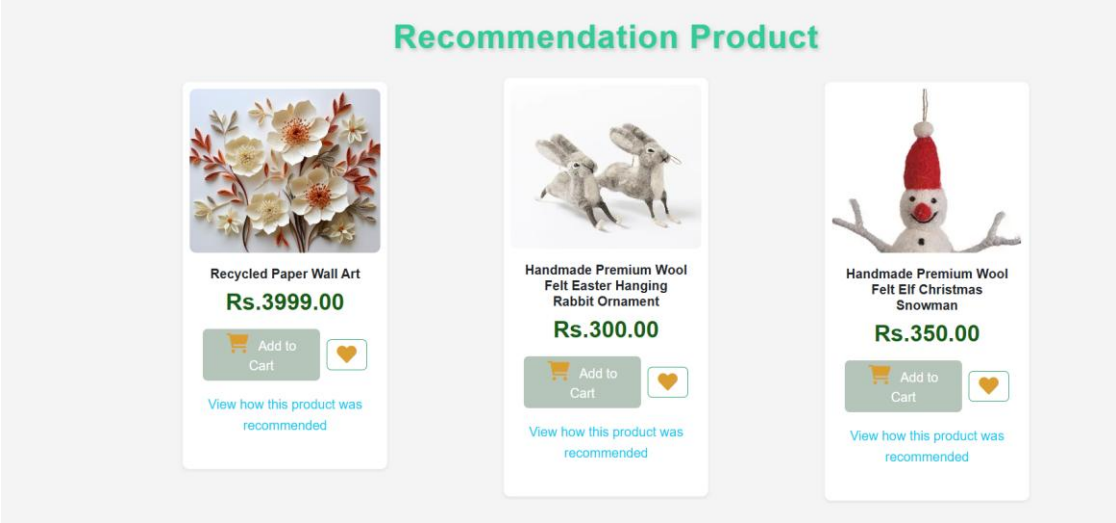
Screenshot 2 Admin Login Interface



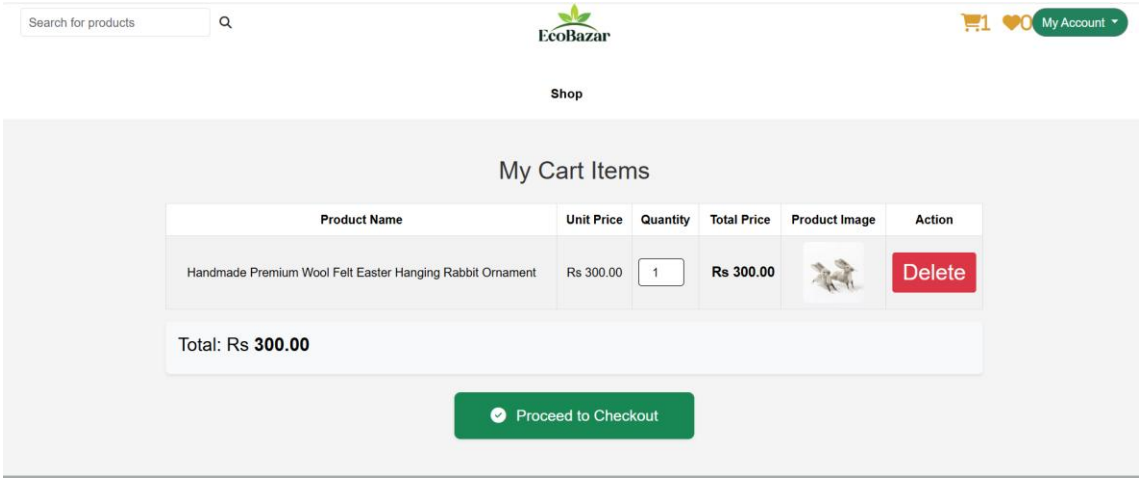
Screenshot 3 Adding Product Interface



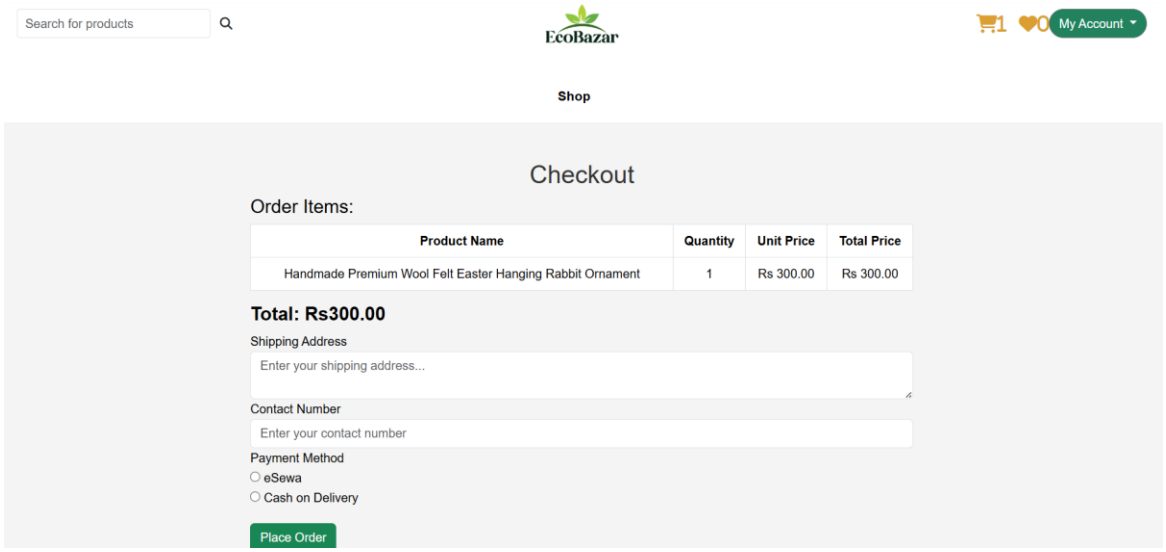
Screenshot 4 Search products result



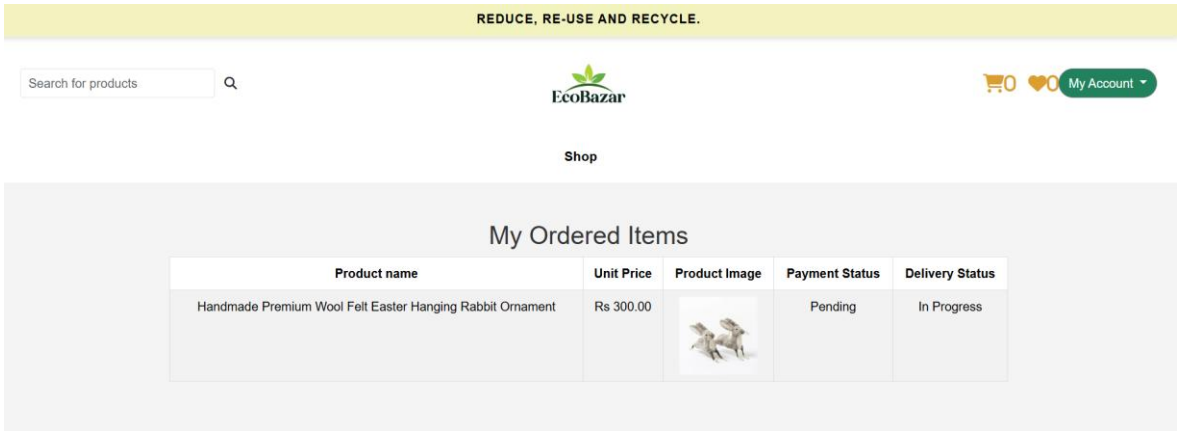
Screenshot 5 Recommendation after User Login



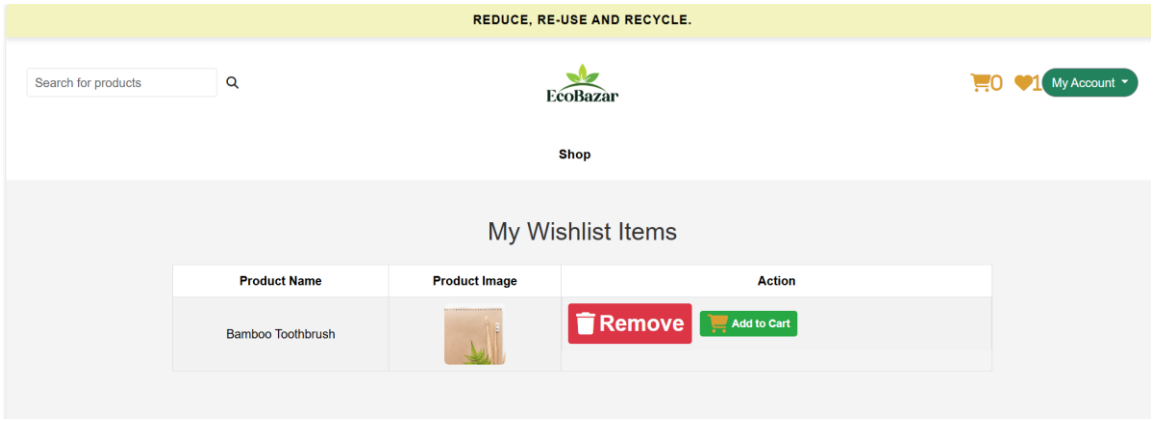
Screenshot 6 User Cart items



Screenshot 7 User checkout



Screenshot 7 User Ordered items



Screenshot 8 User wish list items