Multi-Mart

by
Mr. RAHUL JAGWAN
Mr. AMAN PAL
Mr. MANOJ MANCHIKATLA
Mr. RANJIT CHAUHAN

Under the Guidance of

Mrs. KAUSAR MANER

Department of Computer Engineering

VIDYA VIKAS EDUCATION TRUST'S

UNIVERSAL COLLEGE OF ENGINEERING KAMAN, VASAI - 401208

UNIVERSITY OF MUMBAI

2024-2025

Abstract

The development of a multi-vendor shopping platform is transforming the e-commerce industry by bringing multiple sellers together within a single digital marketplace. This system provides customers with access to a diverse range of products from different vendors in one place, making shopping more convenient and efficient. By incorporating intelligent algorithms, the platform delivers personalized product suggestions, enables price comparisons, and ensures real-time inventory updates. As a result, shoppers can make well-informed purchasing decisions, while sellers benefit from reaching a broader and more varied audience. However, several challenges must be addressed to ensure long-term success. Maintaining accurate and consistent product information across vendors is crucial for building customer trust. Additionally, optimizing order fulfillment processes is essential to meet expectations for prompt and reliable deliveries, especially on a large scale. Fraud prevention remains a key concern, requiring secure payment methods and safeguards against malicious activities to protect both buyers and sellers. Furthermore, simplifying vendor onboarding and integration is necessary to attract and retain merchants, allowing them to list and manage their products effortlessly. This report explores innovative strategies to enhance user experience and operational efficiency, examining the structure of multi-vendor platforms, their advantages, and the obstacles they face. It also outlines future developments, including improvements in security, scalability, and engagement. By incorporating advanced technologies such as augmented reality for product previews and predictive analytics for inventory management, the platform can evolve to meet changing market demands, strengthening customer relationships and vendor partnerships

Chapter 1 Introduction

A multi-vendor marketplace is an online platform that connects multiple sellers with a wide customer base. Similar to a virtual shopping mall, it offers customers access to a variety of products from different vendors, all in one place. Unlike traditional e-commerce stores that sell their own merchandise, these platforms bring together a diverse range of sellers, providing more choices and competitive pricing. By analyzing customer preferences and shopping patterns, the platform suggests relevant products, enhancing the overall shopping experience and helping buyers make well-informed purchasing decisions.

1.1 Project Idea

In today's fast-paced digital economy, online shopping has become a fundamental aspect of consumer behavior, with a growing preference for multi-vendor platforms that offer a wide selection of products and services. However, despite the convenience these platforms provide, shoppers often struggle with overwhelming choices, price differences, and difficulties in finding the best deals. This creates an opportunity to develop a multi-vendor shopping assistant that enhances the online shopping experience. The goal is to build an intelligent system that seamlessly interacts with multiple vendors, gathers product information, and offers personalized recommendations based on user preferences, browsing history, and purchasing habits. By analyzing large volumes of data from various e-commerce platforms, this shopping assistant can identify trends and suggest relevant products that match the user's needs.

Consumers today seek more customized shopping experiences, and such a system can bridge the gap between buyers and sellers by simplifying searches, enabling price comparisons, and notifying users of discounts or special promotions. Additionally, integrating real-time customer support features, such as interactive chat assistance, can improve user engagement and satisfaction.Beyond convenience, this project also aims to foster trust and transparency in online shopping by providing detailed reviews, ratings, and insights into product quality and vendor reliability. By streamlining the shopping process and ensuring reliable information, the multi-vendor shopping assistant empowers consumers to make well-informed purchasing decisions.

Chapter 2

Review of Literature

A literature survey was carried out to find various papers published in international journals related to related to multi-vendor e-commerce platforms to get the best algorithm for the same..

2.1 Existing System

In the rapidly growing e-commerce industry, traditional single-vendor marketplaces dominate, often resulting in limited product choices and higher prices due to the absence of competitive pricing. This lack of integration forces consumers to visit multiple websites to compare products, prices, and features, making the shopping process time-consuming and inefficient. Shoppers frequently encounter issues such as inconsistent product quality, unreliable delivery services, and poor customer support, all of which diminish overall satisfaction.Furthermore, many existing platforms struggle to implement effective recommendation systems that personalize the shopping experience based on individual preferences and past purchases. Complicated checkout procedures contribute to high cart abandonment rates, leading to lost sales opportunities.While some platforms have introduced automation to improve user experience, their scope is often limited to basic product suggestions or simple chat-based assistance. There remains a significant gap in creating a seamless and interactive shopping experience that efficiently connects multiple vendors.Additionally, consumers lack essential tools to manage their shopping activities, such as price tracking, product availability alerts, and streamlined purchasing options. This fragmented system not only affects customer satisfaction but also limits opportunities for platforms to innovate and enhance shopping convenience.The Multi-Vendor Shopping Assistant aims to overcome these challenges

by creating a unified marketplace where diverse vendor offerings are consolidated. Through efficient data analysis and user-driven recommendations, the platform will simplify decision-making, improve shopping efficiency, and offer a more convenient experience for consumers.

2.2 Literature Survey

A detailed examination of recent research papers in the domain of multi-vendor e-commerce platforms was carried out for this project. Table 2.1 shows a survey of the research paper done for the project.

Table 2.1 – Literature Survey table

Sr. No.	Paper Name	Year of Publication	Author(s)	Publication	Proposed Work	Research Gap
1	On-Demand Multi-Agent Basket Picking for Shopping Stores	2023	Mattias Tiger, David Bergström, Simon Wijk Stranius, Evelina Holmgren, Daniel de Leng, Fredrik Heintz	IEEE International Conference on Robotics and Automation (ICRA)	Proposed a new method for basket-aware multi-agent systems to improve customer satisfaction in automated shopping stores	Current state-of-the-art methods are not optimized for basket membership or customer satisfaction in real-time systems
2	AI Multi-Agent Shopping System	2023	Pragyan Nanda, Srikanta Patnaik	Decision Analytics Journal	Proposed a multi-agent coalition framework for optimal e-commerce order fulfilment, focusing on cost minimization, lead time reduction, and customer	Existing systems lack collaboration between sellers, leading to inefficiencies in cost and delivery speed
3	Multi-Agent Coalition-Based Approach for Order Fulfilment in E-commerce	2022	Michael Green, Linda White	IEEE	A reinforcement learning approach that adapts trading strategies based on market conditions.	Needs improvement in real-time processing and decision-making.
4	AI Based Multi-Agent Online Shopping System	2022	Priyanka Bingi, Ambika Guttikonda, Shweta Madgundi, V.S. Patki	International Research Journal of Engineering and Technology	Proposed an AI-based multi-agent system to automate and simplify the online shopping process, including price negotiation	Limited user-friendly interfaces in existing e-commerce systems, lacking intelligent shopping behavior

				(IRJET)	and product recommendation	simulation.
5	The Rising Trends of Smart E-Commerce Logistics	2021	Hicham Kalkha, Azeddine Khiat, Ayoub Bahnasse, Hassan Ouajji	IEEE Access	Explores the impact of digitalization on e-commerce logistics using ICT like IoT, AI, Blockchain, and Cloud computing	Identifies underutilization of technologies like computer vision for product quality inspection and limited research on accessibility for people with disabilities. Suggests further exploration of deep learning for Vehicle Routing Problems (VRP) optimization
6	A Multi-Agent Coalition-Based Approach for Order Fulfilment in E-commerce	2021	Pragyan Nanda, Srikanta Patnaik	Decision Analytics Journal	Proposes a multi-agent coalition framework to optimize e-commerce order fulfilment by minimizing cost, lead time, and maximizing	Research lacks deep exploration of agent interaction complexities and real-world dynamic constraints like fluctuating demand and supplier competition

2.3 Problem Statement and Objective

The problem in a multi-vendor system arises from managing multiple sellers and their diverse products while ensuring a seamless experience for both buyers and sellers.

- Co-ordination Between Buyer and Seller
- User Authentication
- Tracking the Delivery
- Vendor Performance visualization
- Customer Interest Survey for Recommendation
- Reliable Shopping Experience
- Creating Successful Events
- Real-time Payment Integration

2.4 Project Scope

The scope of the **Multi Mart** project defines the core features and functionalities of the multi-vendor e-commerce platform. It focuses on building an intuitive interface that allows customers to explore and purchase products from multiple vendors effortlessly. Vendors will have dedicated dashboards to oversee inventory, monitor sales, and handle order processing. The platform will integrate secure payment gateways, such as Stripe, to ensure safe and reliable transactions.Additionally, the project includes developing a robust backend system to manage user accounts, product catalogs, order fulfillment, and customer feedback. Security measures will be implemented to safeguard user data, while customer support features will be incorporated to enhance the overall shopping experience. The platform is designed to be scalable, allowing for future expansion and the introduction of additional features as needed.
Ultimately, **Multi Mart** aims to deliver a seamless and convenient shopping experience for customers while equipping vendors with efficient tools to manage and grow their online businesses.

Chapter 3

Proposed System
This chapter includes a brief description of the proposed system and explores the different modules involved along with the various models through which this system is understood and represented.

3.1 Analysis/Framework/ Algorithm

The main objective of the multi-vendor shopping assistant is to improve the overall shopping experience by offering tailored recommendations to users while optimizing vendor operations..
Key areas of analysis include:
1. User Behavior Analysis: Examining customer preferences, purchasing habits, and interaction history to deliver personalized product suggestions..
2. Vendor Performance Analysis: Analyzing sales data, customer reviews, and inventory turnover to help vendors refine their product offerings and pricing strategies.
3. Market Trends: Monitoring e-commerce trends and consumer preferences to ensure the platform stays competitive and adapts to changing demands.

Framework
The framework for the multi-vendor shopping agent consists of the following key components:
Frontend Interface: A user-friendly interface built with technologies like React.js that allows users to browse products, manage accounts.
Backend Services: A robust backend developed with Node.js and MongoDB, responsible for managing user data, vendor information, product catalogs, and order processing.
AI Recommendation Engine: Leveraging machine learning algorithms to analyze user data and provide personalized product recommendations.
API Integrations: Integrating third-party APIs for payment processing, shipping logistics, and possibly social media for enhanced user engagement.

Recommendation Algorithm (Collaborative Filtering):
This method evaluates user behavior and preferences to generate product suggestions. By applying similarity measures, such as cosine similarity, it identifies users with comparable interests and recommends products they have positively interacted with.

Search Algorithm (k-NN):
The **k-Nearest Neighbors (k-NN)** algorithm enhances product search by matching user queries with vectorized product representations. Utilizing distance metrics like Euclidean distance, it identifies the most relevant products based on query

similarity.

Data Encoding and Classification:
To analyze user data and product attributes, encoding techniques such as one-hot encoding are applied, ensuring compatibility with machine learning models. Classification algorithms categorize products effectively, improving search accuracy and product organization.

Sentiment Analysis:
By evaluating customer reviews and ratings, sentiment analysis determines overall user sentiment (positive, negative, or neutral). Using Natural Language Processing (NLP), this technique extracts insights from feedback, assisting vendors in optimizing product offerings and platform improvements.

3.2 System Requirements

This section will provide the user the required specification of the hardware and software components on which the proposed system is to be implemented.

3.2.1 Hardware Requirements

This subsection will provide the minimum requirements that must be fulfilled by the hardware components. The hardware requirements are as follows: -

- Processor (CPU):Intel Core i3 (7th generation or newer)
- Memory (RAM): 4 GB RAM
- Storage: 256 GB SSD
- Graphics (GPU): Integrated graphics (sufficient for non-intensive workloads)
- Operating System: Windows 10 (64-bit)
- Network Requirements: Moderate Speed Internet is sufficient

: -

- Processor (CPU):Intel Core i3 (7th generation or newer)
- Memory (RAM): 4 GB RAM
- Storage: 256 GB SSD
- Graphics (GPU): Integrated graphics (sufficient for non-intensive workloads)
- Operating System: Windows 10 (64-bit)
- Network Requirements: Moderate Speed Internet is sufficient

3.2.2 Software Requirements

This subsection will provide the versions of software applications that must be installed. The software requirements are as follows: -

- Backend: Node.js
- Frontend: React.js
- Database: MongoDB
- APIs: Stripe payment API

3.3 Design Details

The multi-vendor shopping platform follows a layered architecture, consisting of the presentation, application, data, and integration layers to ensure seamless user interaction and efficient system functionality. The design incorporates a structured database schema for effective data management, ensuring smooth relationships between users, vendors, and transactions.To maintain security and protect user information, the system implements measures such as data encryption and multi-factor authentication. Additionally, the architecture is built for scalability, leveraging microservices and load balancing to enhance performance and support future expansion.By integrating these design principles, the platform provides a reliable, secure, and user-friendly e-commerce environment that optimizes both user and vendor experiences.

3.1 System Architecture

As illustrated in Figure 3.1, the system architecture details the available actions for both users and sellers after they log in or register. Once authenticated, users can explore and purchase products, proceed with payments, track their orders, and provide feedback through reviews. Meanwhile, sellers can manage their product listings, apply discounts, organize promotional events, and oversee order processing.The platform also enables direct communication between users and sellers, allowing discussions related to products or order-related queries. The flowchart concludes with an **"END"** point, signifying the completion of all possible interactions within the system.

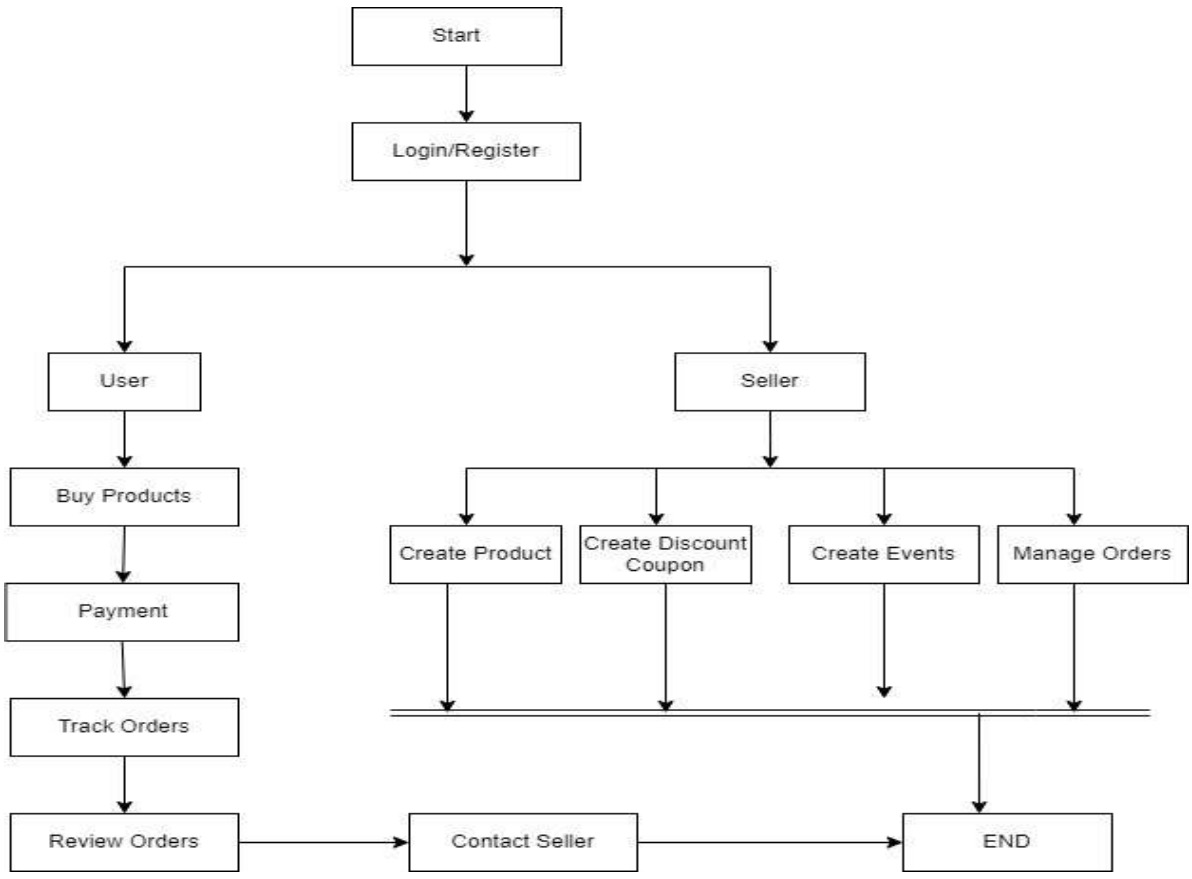

Figure. 3.1 – System Architecture

3.3.2 Details of Modules

The multi-vendor shopping agent is designed with various integrated modules to improve the overall shopping experience, simplify vendor interactions, and optimize product searches. These modules work together to provide seamless navigation, efficient product discovery, and smooth transaction processing.

1. User Registration and Authentication
2. Product Search and Recommendation
3. Vendor Management
4. Order Processing and Payment Integration
5. Review and Rating System

A. User Registration and Authentication

This module enables users to register and log in securely while ensuring the protection of their personal information. User details are stored in a secure database, with authentication mechanisms in place to safeguard data through encryption and secure login practices. Additionally, the system may support social media login options to enhance convenience and accessibility..

B. Product Search and Recommendation

The product search functionality is designed to analyze user preferences and past interactions to deliver relevant search results. By interpreting search queries effectively, the system enhances product discovery and refines recommendations over time. The recommendation engine adapts based on user interactions, ensuring continuous improvement in suggestion accuracy.

C. Vendor Management

This module provides vendors with the necessary tools to oversee their product listings, monitor inventory, and handle order processing efficiently. Vendors can update product details, adjust pricing, and manage availability in real time. Additionally, the system offers access to sales analytics and customer feedback, enabling vendors to refine their offerings and enhance overall business performance.

D. Order Processing and Payment Integration

After a user selects a product, this module manages the entire transaction process, from cart management to order confirmation and shipping coordination. It ensures a seamless checkout experience by integrating multiple payment gateways, providing secure transactions, and enhancing user confidence. Additionally, the system streamlines order tracking and delivery logistics to improve efficiency and customer satisfaction.

E. Review and Rating System

Following a purchase, users can submit reviews and ratings for both products and vendors. This feedback system assists other shoppers in making informed decisions while offering vendors valuable insights for improving their products and services. Additionally, the platform analyzes review data to enhance product recommendations and refine the overall shopping experience.

3.4 Data Model and Description

Data Model describes the relationship and association among data which includes Entity Relationship Model.

3.4.1 Entity Relationship Model

Figure 3.2 Multi-Vendor Shopping Agent, we will model the main entities and relationships involved in such a system. A multi-vendor platform involves interactions between users (buyers), sellers (vendors), products, orders, payments, and reviews.

Entities in Model:

1. **User**: Represents the buyers who purchase products from the platform.
2. **Seller**: Represents the vendors who list their products for sale.
3. **Product**: Represents the goods listed for sale by the sellers.
4. **Order**: Represents the orders placed by the users for one or more products.
5. **Payment**: Represents the payment transactions made by users for their orders.
6. **Review**: Represents feedback or ratings given by users for the products they purchased.

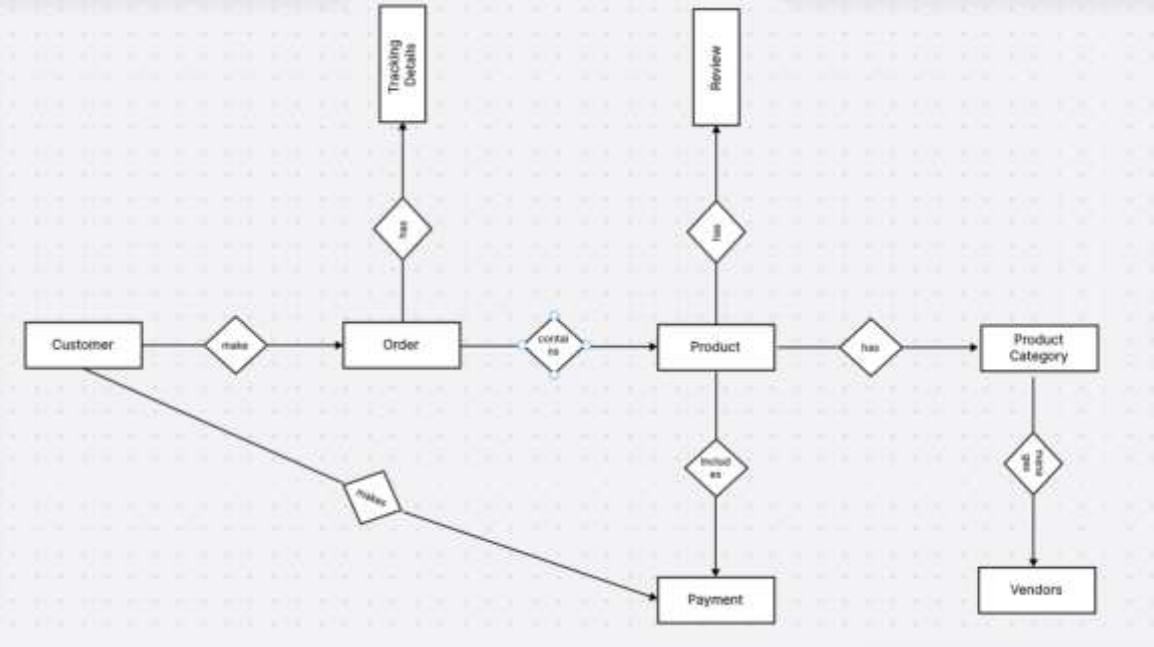

Figure 3.2 - Entity Relationship Diagram

3.5 Fundamental Model

The fundamental model provides a comprehensive overview of the project, illustrating the relationships between various entities, their attributes, and the flow of data within the system. It defines how users, vendors, products, and transactions interact, ensuring a structured and efficient platform design. By mapping out these connections, the model helps in understanding system functionality and optimizing data management for seamless operations.

3.5.1 Data Flow Model

The Data Flow Diagram (DFD) visually represents how data moves through the system, depicting its processes, data inputs, outputs, and storage. It outlines the interactions between users, vendors, and system components, showcasing the logical flow of information. Using standardized symbols and notations, the DFD effectively illustrates how different

entities exchange data, ensuring clarity in system design and functionality..

DFD LEVEL 0

Figure 3.3 illustrates a **Level 0 Data Flow Diagram (DFD)** for the multi-vendor system, providing a high-level representation of data movement and system interactions. It outlines the exchange of information between external entities, such as vendors and customers, and the core system components.The **User Interface (UI)** serves as the primary interaction point, enabling users to perform essential actions like registration, product management, and order placement. The UI communicates with the backend, which processes business logic and manages data transactions. The backend further interacts with the **API layer**, ensuring seamless data exchange between the UI and the database.This structured flow highlights how key data elements—such as product listings, user accounts, and orders—are processed within the system. By maintaining a clear and simplified representation, the Level 0 DFD effectively captures the fundamental functionality of the multi-vendor platform.

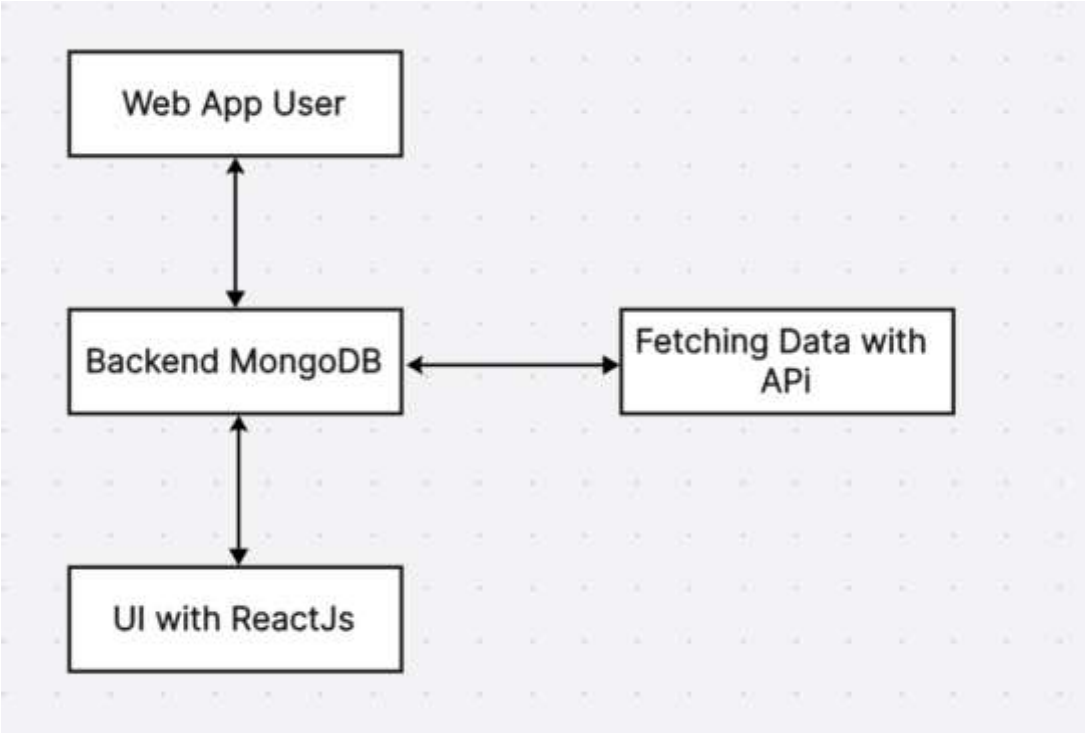

Figure 3.3 – DFD Level 0

DFD Level 1

Figure 3.4 presents a **Level 1 Data Flow Diagram (DFD)** for the multi-vendor system, offering a more detailed breakdown of the processes outlined in the Level 0 DFD. It illustrates the flow of data between key components, including the **User Interface (UI), backend, API, and database**, while mapping out core system functionalities.At this level, specific processes such as **user registration, product management, order processing, and payment handling** are depicted. Users interact with the UI to perform actions like account creation, product listing, and order placement. The backend is responsible for managing business logic and data transactions, ensuring seamless communication with the database for information retrieval and updates.The diagram captures how user inputs are processed and how responses are generated, demonstrating the system's dynamic interactions. By visualizing these dependencies, the Level

1 DFD provides a clearer understanding of the operational flow within the multi-vendor platform, ensuring an efficient and responsive shopping experience.

.

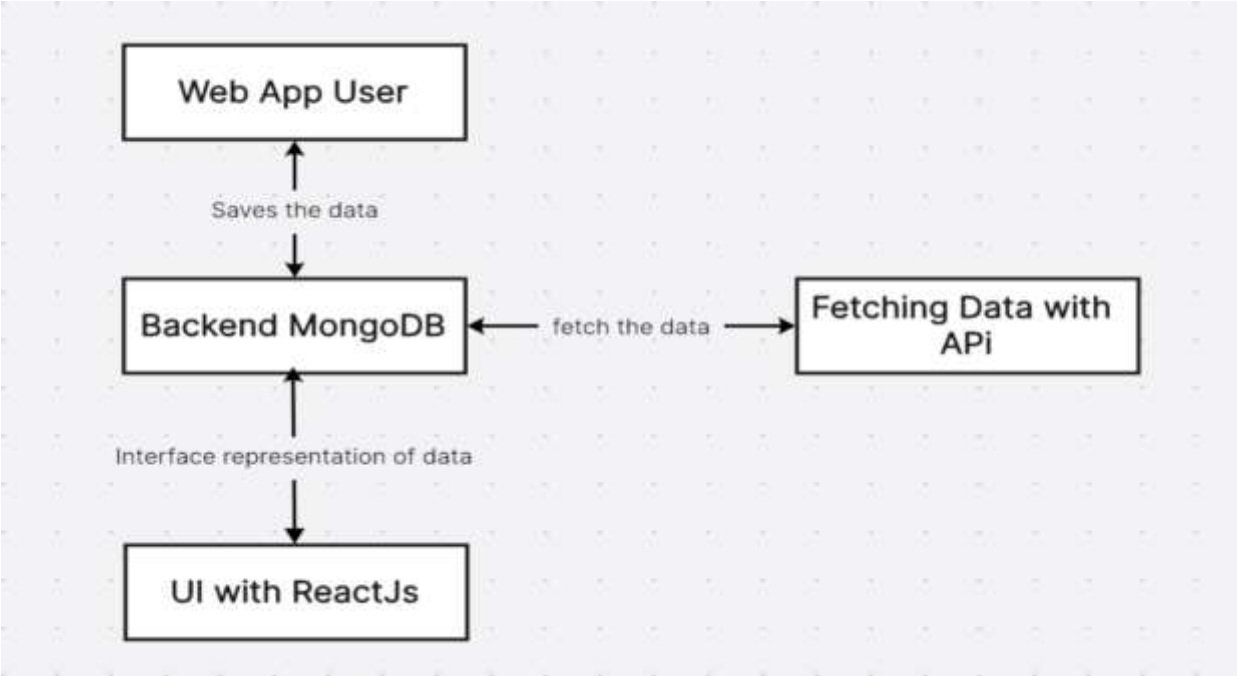

Figure 3.4 – DFD Level 1

DFD LEVEL 2

Figure 3.5 presents a **Level 2 Data Flow Diagram (DFD)** for the multi-vendor system, offering a detailed representation of data movement and interactions between key components, including the **User Interface (UI), backend services, APIs, and database**.At this level, users perform various actions such as **browsing products, placing orders, and managing vendor profiles** through the UI. These actions trigger backend processes via **API calls**, where specialized services like **order management and product management** handle the requests. The APIs act as intermediaries, ensuring seamless communication between the UI and backend while managing data transactions efficiently.This diagram highlights how data is **processed, stored, and retrieved**, demonstrating the structured flow of information between system components. By showcasing the dependencies and interactions among users, backend services, and data repositories, the Level 2 DFD provides a **comprehensive understanding of the platform's operational workflow and data handling mechanisms**.

.

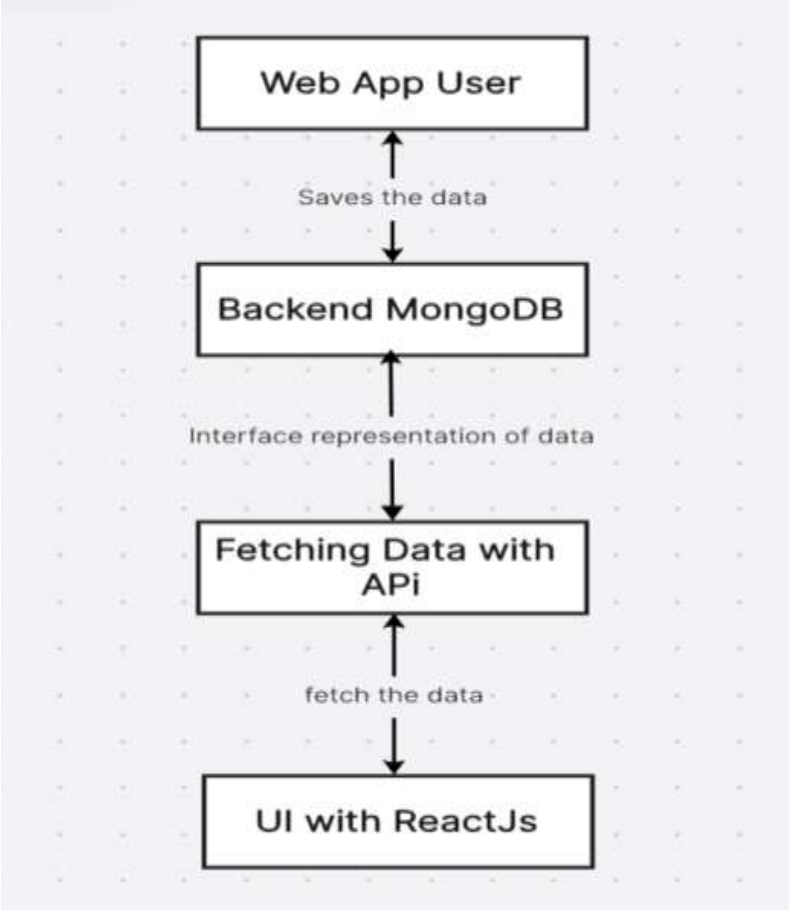

Figure 3.5 – DFD Level 2

Methodology

The development of the multi-vendor shopping system follows a structured approach to ensure efficiency, scalability, and reliability. The process begins with **requirement gathering**, where user needs, business objectives, and system functionalities are identified through discussions, surveys, and research. This step helps define the overall scope of the project. Next, in the **system design** phase, the platform's structure is planned, including database architecture, API integration, and user interface layouts, ensuring seamless interaction between components. The **development** phase follows, where the front-end, back-end, and database are implemented based on the finalized design. Using an iterative approach, incremental updates and improvements are made to enhance performance and usability. After development, the **testing** phase is conducted, involving unit testing, integration testing, and user acceptance testing (UAT) to identify and resolve any errors, ensuring smooth functionality. Once testing is completed, the **deployment** phase takes place, where the system is launched on a live server with necessary configurations and security measures. A backup plan is also established to address any potential issues. Finally, **ongoing maintenance and updates** are performed to optimize performance, enhance security, and introduce new features based on user feedback and technological advancements.

Chapter 4

Result and Discussion

This chapter includes the snapshots of the actual outputs that were seen by the user and this chapter also contains the results of the proposed system.

4.1 Proposed System Result

The proposed system for the multi-vendor website aims to enhance the online shopping experience by providing a seamless platform for both customers and vendors. It simplifies product discovery, purchasing, and vendor management, ensuring a user-friendly and efficient marketplace. Customers can browse a diverse range of products, compare prices, add items to their cart, and complete secure transactions. Vendors can manage their product listings, track sales, and respond to customer inquiries. The platform also includes order tracking, customer support, and review systems to improve trust and transparency. Figure 4.1 displays the GUI of the Home Page of the Multi-Vendor Website, which features functionalities such as browsing product categories, searching for specific items, viewing featured deals, accessing user accounts, and managing vendor dashboards. This intuitive interface ensures a smooth and engaging experience for all users, fostering a competitive and dynamic e-commerce environment.

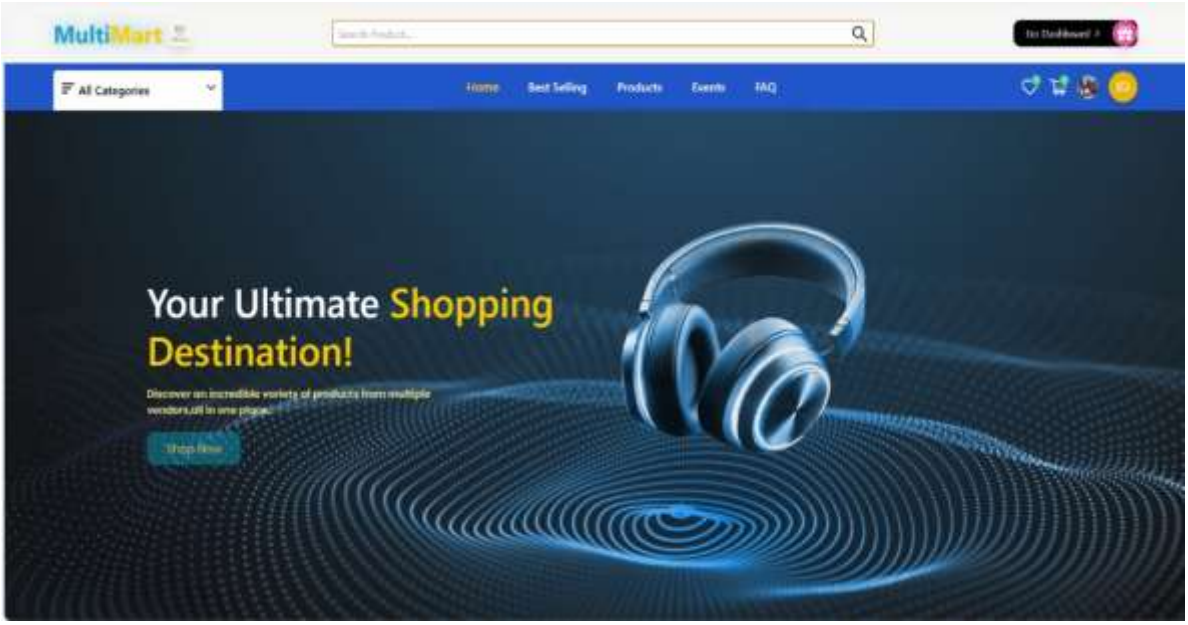

Figure 4.1 GUI for Home Page

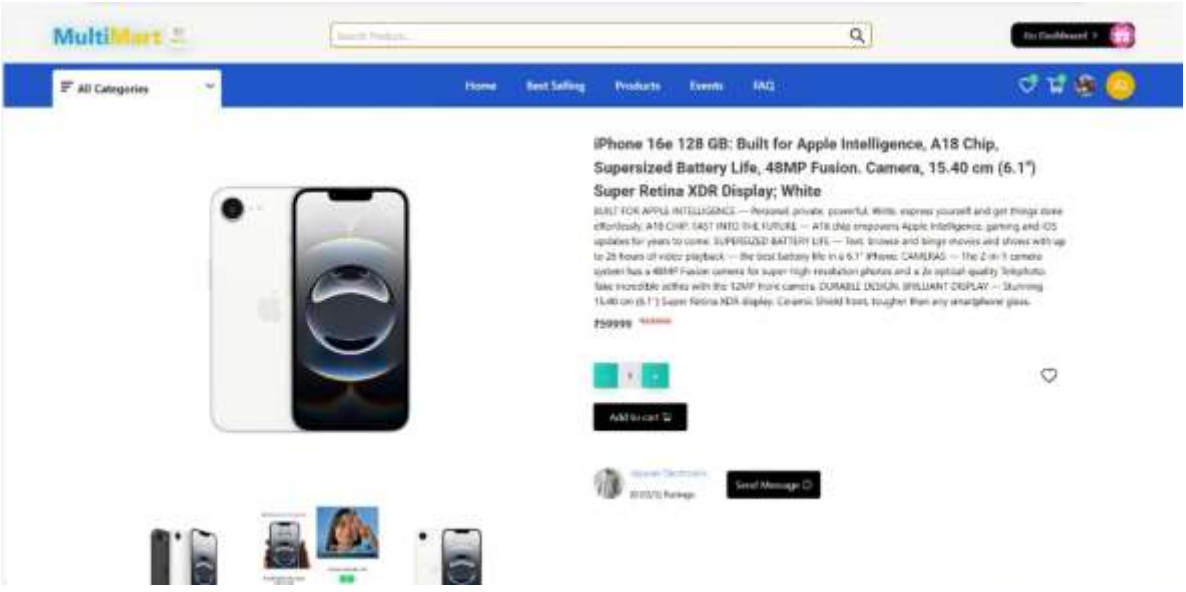

Figure 4.2 GUI for Product Page

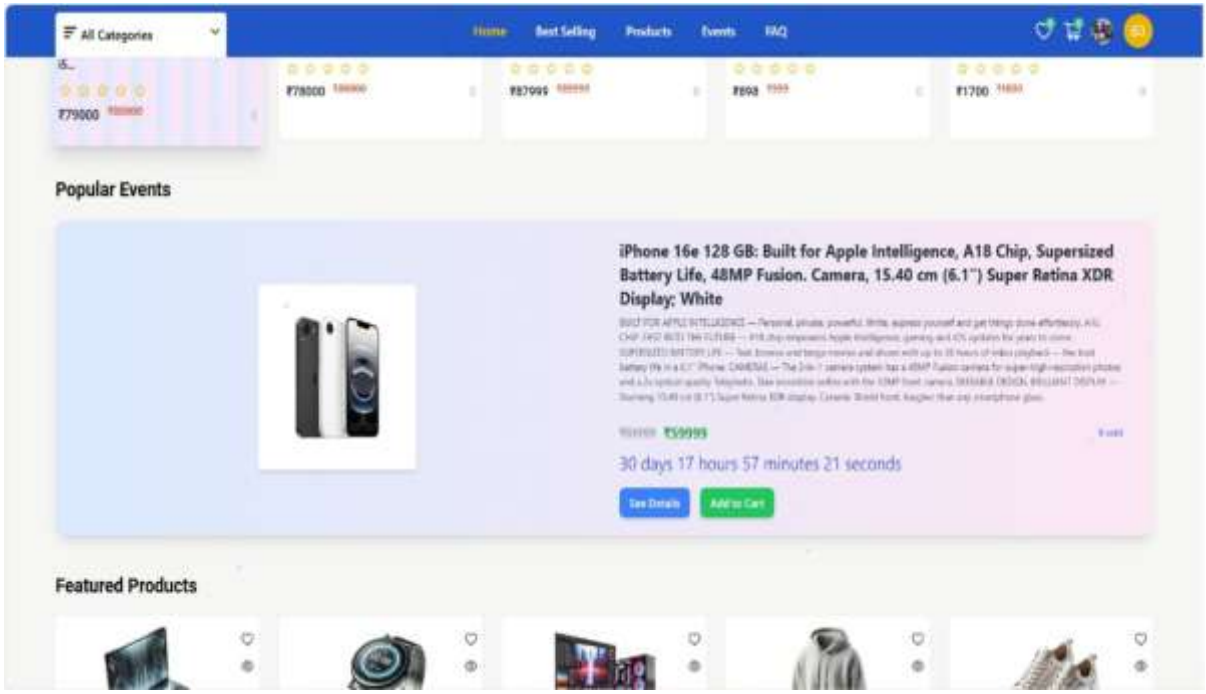

Figure 4.3 GUI for Popular Events

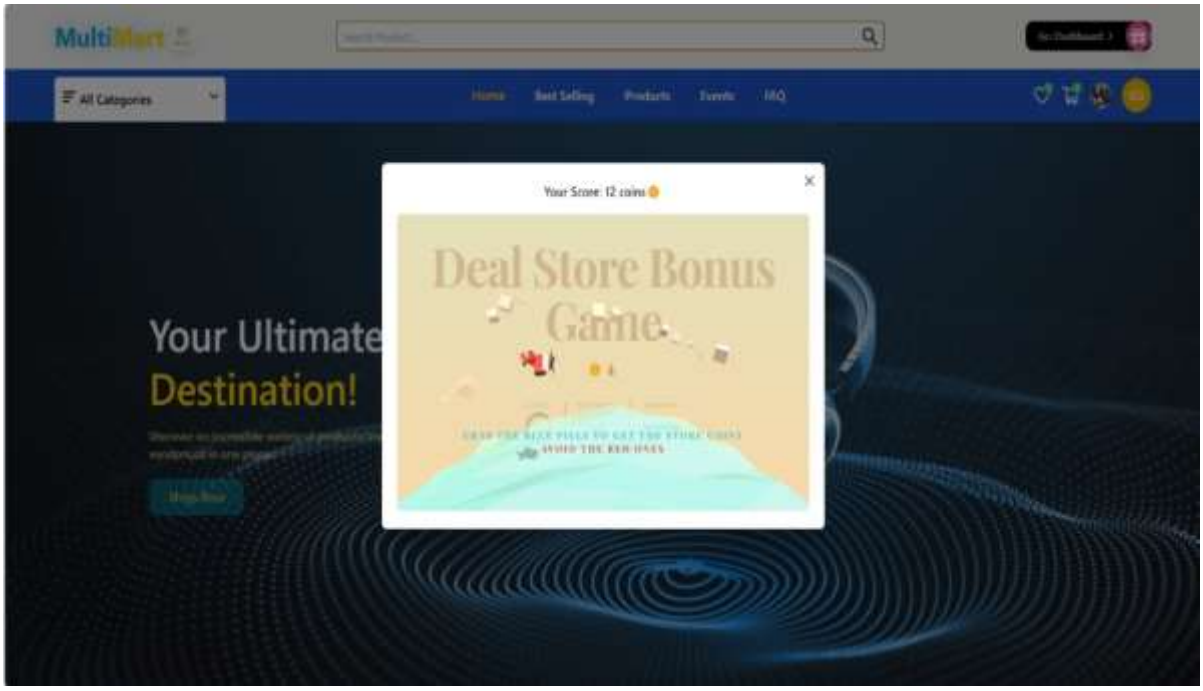

Figure 4.4 GUI for Generating Coupons Code

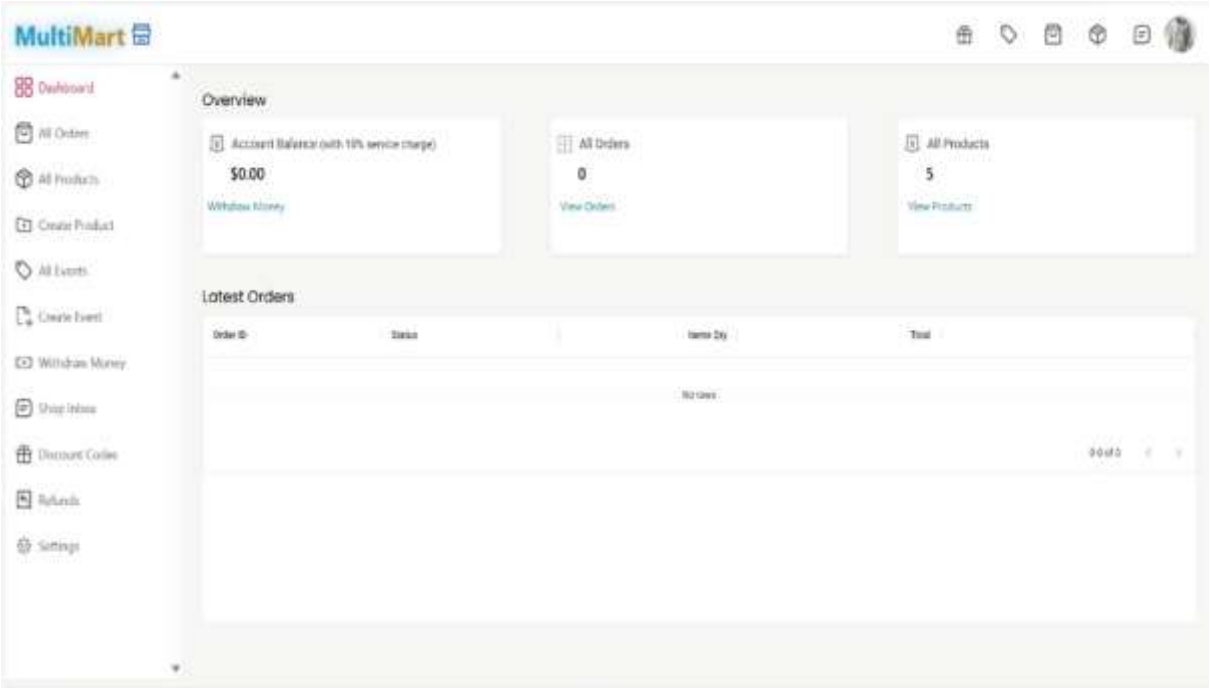

Figure 4.5 GUI of SELLER DASHBOARD

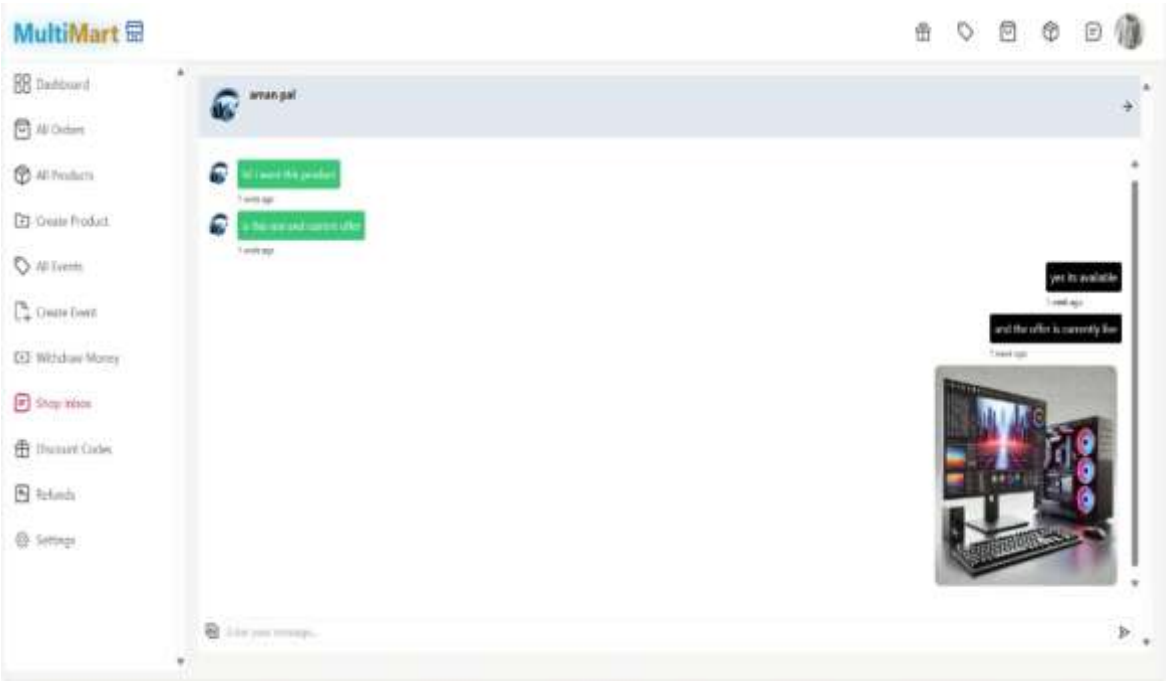

Figure 4.6 GUI for Connecting Seller and User

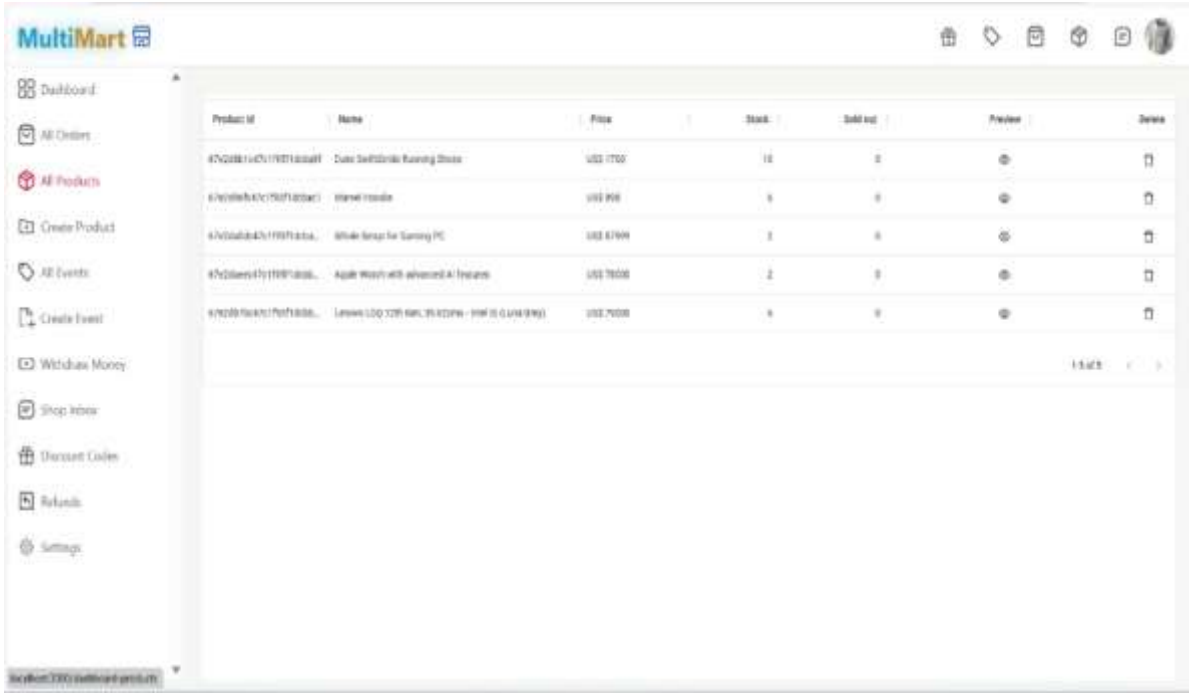

Figure 4.7 GUI of Seller Dashboard For All Products

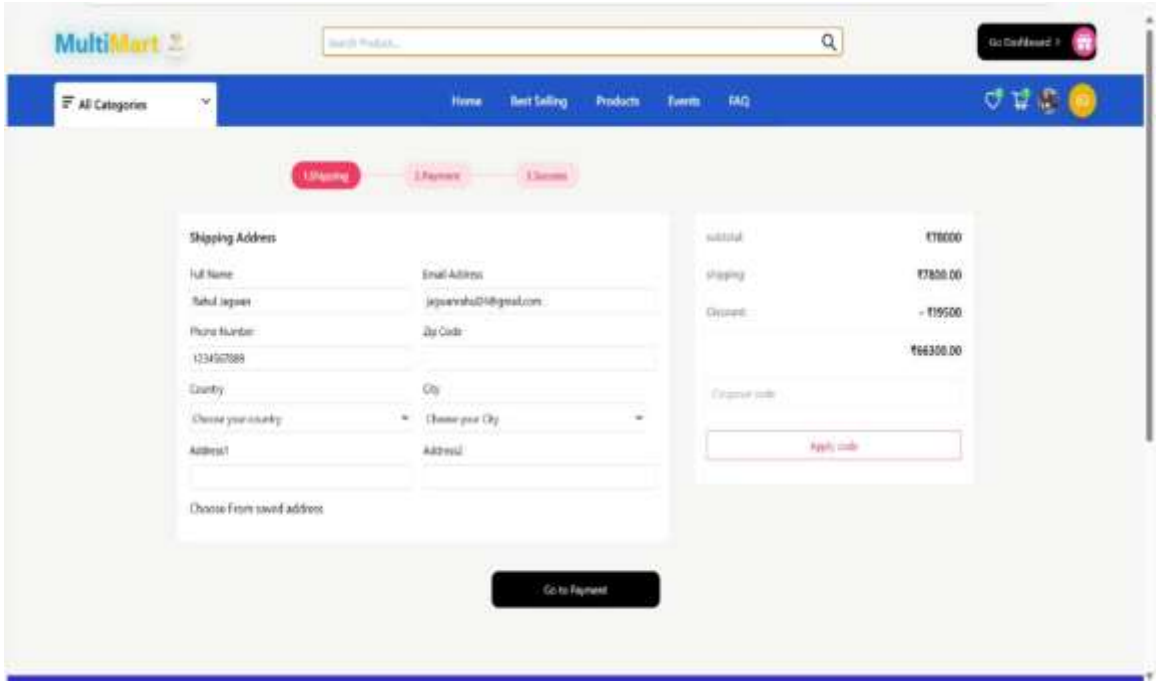

Figure 4.8 GUI for Shipping

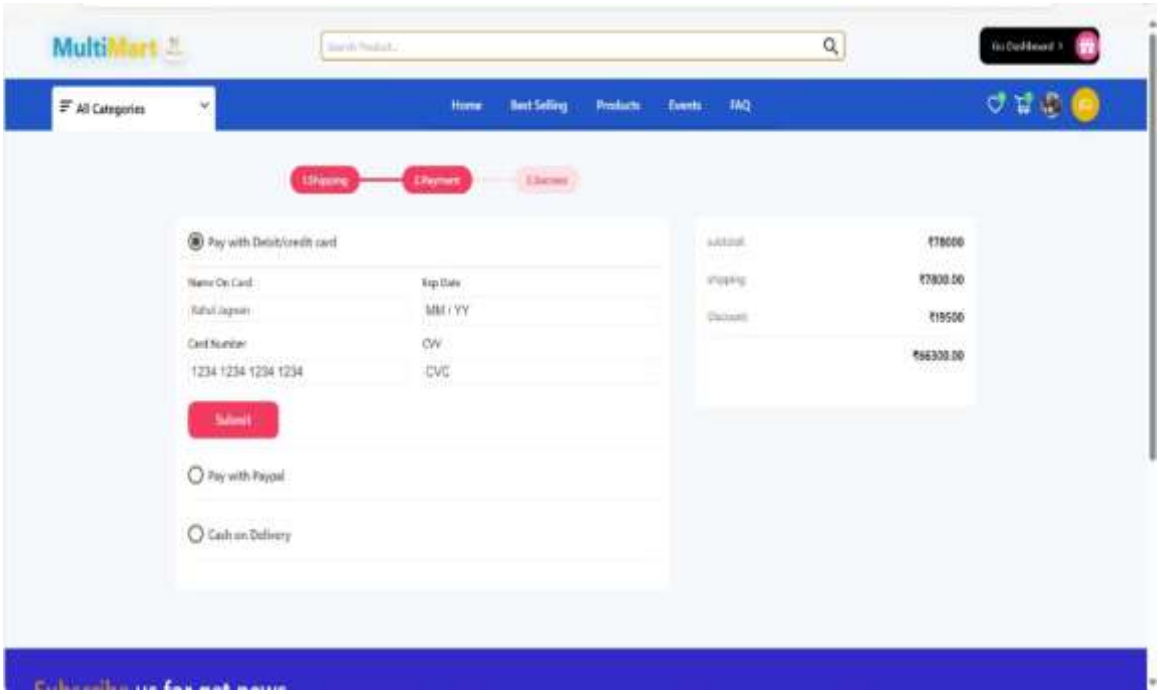

Figure 4.9 GUI for Payment

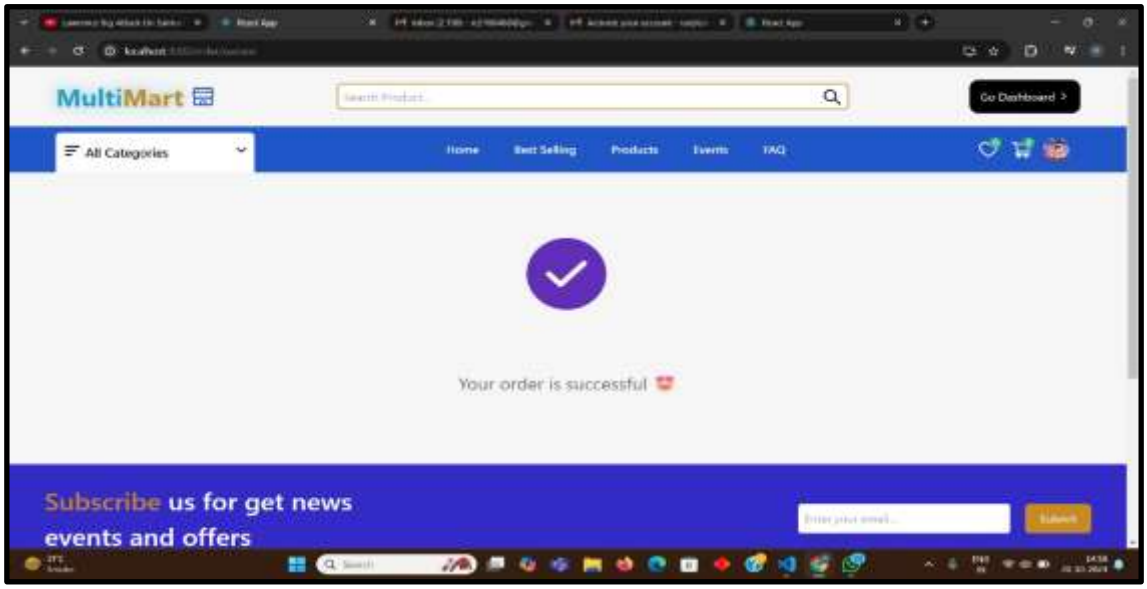

Figure 4.10 GUI of Payment Successful

Parameter	Existing Systems	MULTI-MART
User Interface	Basic and non-responsive UI	Modern, responsive, and user-friendly UI
Vendor Management	Limited vendor tools and support	Comprehensive vendor dashboard with advanced features
Payment Processing	Basic payment options with security concerns	Secure payment gateways (e.g., Stripe) with multiple options

Product Management	Limited product categorization and filtering options	Advanced product management with rich filtering and search options
Customer Support	Minimal support channels	Multiple support channels (chat, email, phone)
Order Management	Manual order tracking and processing	Automated order processing with real-time tracking
Scalability	Difficulty in scaling with increased vendors/customers	Designed to scale easily with performance optimization
Data Analytics	Basic sales and traffic reports	In-depth analytics dashboard for vendors and admin
Security Features	Basic security measures	Enhanced security protocols, including SSL and data encryption
Report Generation	Limited mobile support	Fully responsive design for mobile devices
Mobile Compatibility	Minimal integration options	Robust API for seamless integration with third-party services
Integration with Third-Party Services	Basic roles for users	Granular role-based access control for better management

4.2 Proposed system versus existing system

The table below is a comparison table that outlines the key parameters of an existing multi-Mart system versus a proposed multi-Mart system. This table highlights differences in features, performance, and user experience.
Table 4.1 – Comparison between existing and proposed system.

Conclusion

In conclusion, the development of the multi-vendor shopping platform marks a significant step in improving the online shopping experience. By implementing an efficient system for product discovery, order management, and vendor interactions, the platform ensures a seamless and user-friendly marketplace. The structured approach to data management, secure transactions, and a well-designed user interface enhances both customer satisfaction and vendor operations. Additionally, continuous improvements and scalability strategies allow the system to adapt to changing consumer preferences and market trends. As e-commerce continues to evolve, this platform will serve as a valuable solution for shoppers seeking convenience and vendors aiming to expand their businesses. Overall, the project addresses existing challenges in online retail while setting the foundation for future enhancements in digital commerce..

Appendix

1 **ReactJS:** React is employed for building the user interface, offering a dynamic and responsive front-end for the mobile and web applications

.

2 **MongoDB**: React is employed for building the user interface, offering a dynamic and responsive front-end for the mobile and web applications

.

3 **Express:** Express is the back-end framework used to build the server and manage API routes, handling the communication between the front-end and the database.

.

4 **Stripe API**: The Stripe API is a powerful tool for integrating payment processing into web applications, allowing developers to securely handle transactions, subscriptions, and invoicing. With comprehensive documentation and libraries for various programming languages, Stripe makes it easy to implement payment solutions with a smooth user experience.

5 **NodeJS:** Node.js serves as the runtime environment for building and running the server-side code, ensuring the app is fast and efficient.

This tech stack ensures a powerful, real-time, and responsive system for both user interface and back-end processes.

References

[1] Mattias Tiger, David Bergström, Simon Wijk Stranius Evelina Holmgren, Daniel de Leng and Fredrik Heintz,“ On-Demand Multi-Agent Basket Picking for Shopping Stores ” ,IEEE,2023.

[2] Mushtaq Ahamed K. S, Mahesh N, Prasanna Kumar P, Thirukkumaran R, “AI Multi-Agent Shopping System”,Decision Analytics Journal,2023.

[3] Pragyan Nanda, Srikanta Patnaikb, “Multi-Agent Coalition-Based Approach for Order Fulfilment in E-commerce”,Elsevier ,2023.

[4] Prashant Gawande, Dhaval Ingale, Gauri Thorat, Nikhil Pawar, Hemant Bhattad, “AI Multi Agent Shopping System”, IJRASET,2021.

[5] Hicham Kalkha, Azeddine Khiat, Ayoub Bahnasse, Hassan Ouajji, Shweta Madgundi, V.S. Patki, “The Rising Trends of Smart E-commerce Logistics”,IEEE,2021.

[6] Pragyan Nanda, Srikanta Patnaik, “A Multi-Agent Coalition-Based Approach for Order Fulfilment in E-commerce”, Decision Analytics Journal,2021.