MULTI-VENDOR ECOMMERCE WEBSITE

Submitted By

Md.Jahid Hasan

221-35-959

Supervised By

Mr. Khalid Been Badruzzaman Biplob

Lecturer (Senior Scale) Department

of Software Engineering

Daffodil International University

This project report has been submitted in fulfilment of the requirements for the degree of **Bachelor of Science in Software Engineering**

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Md.Jahid Hasan

Date of Birth : 16/10/2002

Title : Multi Vendor Ecommerce Website

Academic Session : 2022-2026

I declare that this thesis is classified as:

☒ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*

☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)*

☐ OPEN ACCESS I agree that my project to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Project is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the Project for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the Project for academic exchange.

Certified by:

_____ _____ (Student's Signature) (Supervisor's Signature)

_____ _____

Student ID Name of Supervisor

Date: Date:

NOTE: * If the Project is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

PROJECT DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City,
Ashulia.Dhaka,Bangladesh

Dear Sir,

CLASSIFICATION OF Project AS RESTRICTED

Please be informed that the following project is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name
Project Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, *Daffodil International University* with its copy attached to the thesis.

SUPERVISOR'S DECLARATION

I/We* hereby declare that I/We* have checked this project and in my/our* opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

(Supervisor's Signature)

Full Name : Mr. Khalid Been Badruzzaman Biplob

Position : Lecturer (Senior Scale)

Date : 27-11-2025

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

_____

(Student's Signature)

Full Name : Md.Jahid Hasan

ID Number : 221-35-959

Date : 27 November 2025

APPROVAL

This project titled on "**Multi Vendor Ecommerce Website**", submitted by **Md.Jahid Hasan (ID: 221-35-959)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. S. M. Hasan Mahmud
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

A.H.M Shahariar Parvez
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Tapushe Rabaya Toma
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Khalid Been md. Badruzzaman Biplob
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Dr. Md Sazzadur Rahman
Professor
Institute of Information technology
Jahangirnagar University, Bangladesh

MULTI VENDOR ECOMMERCE WEBSITE
Md.Jahid Hasan

Project submitted in fulfillment of the requirements for
the award of the degree of
Bachelor of Science/Master of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to Almighty Allah for giving me the strength, patience, and ability to successfully complete this project.

I would like to express my deepest appreciation and heartfelt thanks to my respected supervisor and teachers for their continuous guidance, valuable suggestions, and constant encouragement throughout the entire development of this project. . They were essential in the success of this project as they provided support and motivation.

I am also thankful to Department of Software Engineering Daffodil International.

University, to avail to me the opportunity and resources required to conduct this project.

My academic experience of knowledge and skills earned me the following. much

assisted me in the formation of this system.

Thanks to my peers and friends, who were very cooperative, had fruitful discussions. and

professional assistance in this project work. Their support and cooperation ensured.

The process of development simplified and made more enjoyable.

Lastly, I would like to thank my family members with my whole heart as they have offered unconditional.

help, motivation, and impetus during my schooling. Without their support,

without this it could not have been accomplished.

DEDICATION

I hence announce that I have accomplished this project under the supervision of "Mr. Khalid Been Badruzzaman Biplob" "Lecturer(Senior Scale) Department of Software Engineering Daffodil International University. And also state that whole record has not been presented in any other place to obtain my degree.

ABSTRACT

The present project is aimed at designing and developing a Multi-Vendor E-commerce Platform where people have an opportunity to purchase goods online in a secure and efficient manner. The system allows the customer to create an account, search, add to the cart, place orders, pay online, and track the orders. Vendors are able to control their products and orders, and the overall system activities are controlled by the admin. The overall objective of the project is to deliver a secure, convenient, and scalable web-based shopping system with the help of the latest web technologies.

TABLE OF CONTENT

DECLARATION

TITLE PAGE

ACKNOWLEDGEMENTS **5**

DEDICATION **6**

ABSTRACT **7**

TABLE OF CONTENT 8

LIST OF TABLES

LIST OF FIGURES 11

LIST OF SYMBOLS

LIST OF ABBREVIATIONS

LIST OF APPENDICES **12**

CHAPTER 1 INTRODUCTION **13**

1.1 Background 13

1.1.1 Context and Relevance 13

1.1.2 Problem Identification 13

1.1.3 Purpose and Justification

1.1.4 Scope

1.2 Project Planning and Initiation

Feasibility Study (Step-by-Step) 17

1.3 Target User Profile and Tentative Elicitation Process 20

1.3.1 Target User 20

1.3.2 User profile 20

1.3.3 Elicitation Process 22

1.4 Project Block Diagram 22

1.5 System Requirements 23 1.5.1 Hardware Requirements .

1.5.2 Software Requirements .

1.5.3 Constraints and Dependencies 23
1.6 Project Scheduling 24
1.7 Summary 24
CHAPTER 2 DESIGN AND IMPLEMENTATION 27
2.1 Introduction 27
2.2 Functional Requirements 27
2.3 Non-Functional Requirements 28
2.3.1 Performance 28
2.3.2 Reliability 29
2.3.3 Portability 30
2.4 Object-oriented System design using UML 31
2.4.1 Use Case Diagram 31
2.4.2 Case Description 33
2.4.3 Activity Diagram 49
2.4.4 Sequence Diagram 55
2.4.5 Class Diagram 61
2.4.6 ER Diagram 62
2.5 Coding: Appendix A 62
2.6 Summary 63
CHAPTER 3 SOFTWARE TESTING 65
3.1 Introduction 65
3.2 Testing Features 65
3.2.1 Feature to Be Tested 66
3.3 Testing Strategies 66
3.3.1 Test Approach 68
3.3.2 Pass/Fail Criteria 68
3.4 System Testing (Test Cases with Report) 68
3.5 Summary 73
CHAPTER 4 DEPLOYMENT AND MAINTENANCE 73
4.1 Introduction 73
4.2 Try to follow the SRLC (software release life cycle).

CHAPTER 5 USER MANUAL **74**

5.1 Introduction 74

5.2 Project Functionalities

5.3 Summary 74

6.5 Conclusion 74

REFERENCES **74**

LIST OF FIGURES

Figure No.	**Title**	**Page No.**
Figure 1.1	System Use Case Diagram	10
Figure 1.2	Customer Use Case Diagram	11
Figure 1.3	Admin Use Case Diagram	12
Figure 1.4	Vendor Use Case Diagram	13
Figure 2.1	Activity Diagram of User Registration	15
Figure 2.2	Activity Diagram of Login Process	16
Figure 2.3	Activity Diagram of Add to Cart	17
Figure 2.4	Activity Diagram of Checkout and Payment	18
Figure 3.1	Sequence Diagram of Order Process	20
Figure 3.2	Sequence Diagram of User Authentication	21
Figure 4.1	Class Diagram of the System	23
Figure 5.1	ER Diagram of the Database	25

LIST OF APPENDICES

Appendix A: Title 19 Appendix B: Title 20

CHAPTER 1 INTRODUCTION

1.1 Background

As the internet technology has increased swiftly, online shopping is being part and parcel of everyday living. The e-commerce sites enable customers to purchase goods conveniently any time wherever they are. A multi-vendor e-commerce system will allow other sellers to sell their products using the same platform and this opens a broader market to both the consumers and the sellers.

The proposed project is Multi-Vendor E-commerce System, and it will offer a secure, easy to use and efficient online shopping platform. Shoppers are able to search and browse products and add them to the cart, online purchases, and track orders. Vendors are able to control their product and the administrators are able to control users, vendors and overall system activities. The system will guarantee a smooth running, security and reliability in online business.

1.1.1 Context and Relevance

- The application of e-commerce systems has witnessed a surge in usage over the past several years because of the easy access to the internet and smart phones. Online shopping has become more preferable to people due to its time-saving nature, convenience, and a broad range of products. Simultaneously, companies are also moving to online platforms in a bid to increase their market share and reach greater numbers of people.
- The Multi-Vendor E-commerce System will be very applicable in the digital economy today. It enables various sellers to sell their products in the same platform and still, customers can easily browse, search, compare and buy products. Such system is helpful to the small business, enhances the customer experience and is significant in the development of online business and online services.

1.1.2 Problem Identification

- In modern business and consumer environment, several challenges exist as businesses and consumers strive to deal with.

- handling online shopping, product choice and communication with suppliers.

The following are some of the identified key issues:
Inefficient Product Management:

- Most of the available e-commerce opportunities lack the means of sorting, classifying and handling a high volume of items effectively. This usually causes both the sellers and the ones who buy to have problems navigating and locating desired products.

- Absence of Real-Time communication:

- There is a tendency to have delays when responding to customers by the sellers. Current

- platforms do not have inbuilt live chat infrastructure which is required to assist in solving queries immediately, which impacts the customer experience and confidence.

- Handling of Orders and Payments:

- Orders processing and tracking payments is not always automated, and also may be manual.

- and contributing to failures, delays and dissatisfaction of customers.

- Limited Fraud Detection:

- The e-commerce sites are susceptible to fraud cases such as frauds related to accounts, frauds related to payments and fake product descriptions. The current systems might not have AI-based surveillance to identify suspicious activity.

- Sub-Optimal User Experience:

- The users might find it difficult to use complex navigation, bad interface design, and

- long loading delays, decreasing the level of engagement and retention.

Problem in Vendor Management:
In the case of multi-vendor, dealing with multiple vendors, their products and so on.

- Without a centralised and analytical system, transactions may be difficult.

1.1.3 Purpose and Justification:

- The objective of this project is to develop a Multi-Vendor E-commerce Platform that can provide a smooth and safe online shopping experience for both customers and vendors. The platform should be designed to simplify and speed up the management of products, automate order processing, allows real time communications through live chat and employ AI-based fraud detection for secure transactions.
- Justification:
- · User/consumer experience by having handy search, preview and purchase products of their choice – Ability to navigate easily will not keep a user waiting or guessing.
- · Effective VENDOR Management : Sellers are able to manage numerous products, initiating orders, contacting with the customers and this enlarges efficiency and prevents operational errors.
- · Safe Transactions: Buyer and seller can trust each other because of the presence of AI-enabled anti-fraud system and auto payment verification.
- · Real-Time Assistance: The unified live chat customer support will enable instant response to the customer queries resulting in increased satisfaction and loyalty.
- · Market Leader: Delivering a feature rich and state of the art platform is not only in response to the limitations behind existing e-commerce solutions, but will also help your business grow its competitive advantage.

1.1.4 Scope :

- This project is being revolving on scope based which define its limitations, features and affecting result. Multi-Vendor E-commerce Platform will include:
- Product Management:
- · Vendors can add, edit and delete products.
- · Organization and categorization of items to make searching simpler.
- User Management:
- Vendor and customer registration and authentication through.
- Firebase and JWT.
- User and vendor profile management.
- Receiving and Processing Orders:
- Orders processing orders track automatically using the process tracking service.
- Seamless payment integration for safe and easy payments.
- Live Chat Support:
- Live chat between users and suppliers to discuss.
- Fraud Detection:
- Automated AI watch on suspicious behavior, and fraud. transactions.
- Responsive Design:
  Platform accessible what are desktops and tablets, and also mobile.

Review and Rating System:

- Customers too can write reviews about the products and its seller to get better.
- Out of Scope:
- Delivery and implementation of the physical aspects.
- Offline Payment options like Cash onDelivery.
- Integrate with outside marketing or analytics tools (version one).
- 1.2 Project Planning and Initiation
- Purpose of Project Planning and Initiation – The purpose of a Project Planning and Initiation is to define the objectives, schedules, resources and plans for execution that will enable the success completion of the project. This phase bootstraps development, identifies threats, and understands the working mechanism.
- Project Objectives:
- The project will be founded on Agile method, which allows iterative development, continuous feedback and improvement of the product.
- Tasks accommodated in isolatable parts: backend, frontend, database and integration..

1.2 Project Planning and Initiation :

The project will be developed according to the Agile methodology, permitting both incremental development by periodic feedback and improvement.

Activities are separated into organized sub-activities: backend, frontend, database, and linking.

Resource Planning:

: Human Resources: Project developer (you), guide/supervisor.

Software Resources: Node. js, Express. js, React, Tailwind CSS & DaisyUI, MongoDB and Firebase.

Hardware Requirements: Laptop / Desktop with minimum 8GB RAM, internet connectivity.

Timeline and Milestones:

Needs Analysis & Design: 2 weeks

Backend Development: 3–4 weeks

Frontend Development: 3–4 weeks

Integration & Testing: 2 weeks

Deployment & Documentation: Last 1 week

Risk Management:
Technical Risks: Glitches, compatibility problems, server outages.

Mitigation: Regular code review, testing and backup systems.
Time risks (time to develop the product in case of unexpected problems).

Mitigation: Good Task allocation, planning with buffer time.

Initiation:

Project officially start with gathering of requirements and design documents to assure that it aligns with objectives.

So that the project goals, tasks and deliverables are well communicated to stakeholders and team members.

Feasibility Study (Step-by-Step)

Feasibility Study: Which determines if a project is feasible and the goal can be achieved considering constraints like time, cost, technology. The feasibility is stepwise performed in this project:

1. Technical Feasibility:

- Platform is based on MERN stack (MongoDB, Express. js, React, Node. js) with Firebase for authentication.
- All of the necessary set of tools and technologies are present and work with each other.
- Teams can handle frontend, backend and fraud detection based on AI

2. Economic Feasibility:

- Most of the development cost is zero as open-source tools and technologies are used.
- Cheap cloud-based hosting and storage.
- By solving real e-commerce problems at scale, this is a project that gives a big bang for the buck.

3. Operational Feasibility:

- The system will improve **user experience, vendor management, and security**.
- Customers and vendors can use the platform easily with minimal training.
- Live chat and automated processes enhance operational efficiency.

4. Schedule Feasibility:

- Timeline Project Timeline will span 12–14 weeks through defined design, development, integration, testing and deployment milestones.
- Realistic in that can be achieved with the available resources and project complexity

Phase 1 Preliminary Analysis & Project Scope Definition:

Objective:

In Phase 1, the project is reviewed to understand and clarify expectations, identify critical issues, and properly determine scope so that we can map out a road for development.

1. Preliminary Analysis:

- Drove requirement by doing research in available e-commerce platforms.
- Key issues identified: Inefficient product management No real-time communication Manual order processing (orders often slow to arrive) Limited fraud detection Poor user experience
- Studied target users (consumers and merchants) and their requirements.
- Evaluated potential technologies (MERN stack, Firebase and AI tools) for compatibility and feasibility.

2. Project Scope Definition:

- **In Scope:**
- Creating and managing multiple vendor products (add, edit, delete and categorize)
- User authentication and profile management
- Secure order and payment handling
- Live chat support for instant Customer-NetworkPartner communication
- AI-based fraud detection
- Desktop and Mobile Responsive Design
- **Out of Scope:**
- Physical delivery and logistics management
- Payment options other than online payment (cash on delivery)
- Integration with third-party marketing tools

3. Outcome of Phase 1:

- A clear comprehension of the project's goals, scope, and limitations.
- Described The scope and functions of the system.
- Laid a basis for system design, development planning and risk analysis in the next phase.

Phase 2 Market Feasibility Analysis (or Market Research):

Objective:

Analysis of the market's demand, competition and potential for Multi-Vendor E-commerce Platform.

.

Key Points:

- Market Demand: The growth of online shopping is exponential, and it's causing the demand for marketplaces to lead.
- Competition Conclusion: We have found that current even the most competitive companies do not provide real time support or fraud detection with AI which is left out of market.
- The customers looking for diversity and separate sellers who are searching from easy ways to sell their products.
- So what nowThe market (add about the strong potential etc.) and the project can address unmet needs giving a competitive edge.

Phase 3 Technical Feasibility Analysis:

Objective:

To make sure that the project is technically feasible using existing technology and resources.

Key Points:

- Technology Used: MERN stack (MongoDB, Express. js, React, Node. js), Firebase, AI libraries for fraud detection.
- Skills Required: Full stack development, database handling, AI implementation.
- Resource Availability: Tools, hardware and software all available in the market.
- Conclusion: The project is technically feasible and can be set up with the available resources and technologies. .

Phase 4 Financial Feasibility Analysis:

Objective:

To determine how much the project will cost and budget for it

Key Points:

- Development Costs Developed : Using MERN stack, Firebase etc, thereull be very less impact on development.
- Host & Maintain: Budget cloud hosting and database services.
- ROI: Gap-in-the market consideration from a platform perspective with discovered revenue potential.
- Conclusion: This project is economically viable with a low development cost and high potential value.

1.3 Target User Profile and Tentative Elicitation Process:

1. Target User Profile:

- **Customers:**
  - Age: 18–45 years
  - Tech-savvy and frequent online shoppers
  - Interested in a variety of products and secure transactions
- **Vendors:**
  - Small to medium business owners ○ Selling multiple product categories online ○ Require efficient product, order, and customer management

2. Tentative Elicitation Process:

- Surveys & Questionnaires: Gather likes and dislikes from future potential users.
- Interviews: to get the issues and demand from Vendors operations.
- Observation - Study available e-commerce platforms to find out gaps in feature set and use experience.
- Analysis: Aggregate and analyse collected information in order to establish functional and non-functional requirements Both are fine.

1.3.1 Target User

1. Customers:

- Age: 18–45 years
- Frequent online shoppers
- Looking for variety, convenience, and secure transactions

2. Vendors:

- Small to medium business owners
- Selling multiple product categories
- Need easy product, order, and customer management

1.3.2 User profile

Table 0: User Profile for Multi-Vendor E-commerce Platform

User Class	Note on Characteristics
Type of user	Customer / Vendor
Age range	Customer: 18–45 years Vendor: 25–50 years
Frequency of use	Customer: Daily/Weekly Vendor: Daily
Mandatory	Customer: Yes, to make purchases Vendor: Yes, to manage products and orders
Computer experience	Customer: Basic to moderate Vendor: Moderate to advanced
Education	Customer: High school and above Vendor: High school, diploma, or higher
Goal	Customer: Buy products conveniently Vendor: Manage products, orders, and communicate with customers
Language skills	Customer: Basic English / Local language Vendor: Moderate English / Local language
Number of users	Customer: 500–1000 (initial) Vendor: 10–50 (initial)
Training	Customer: Minimal, guided by UI Vendor: Short training or tutorial
Other system use	Customer: Other e-commerce platforms Vendor: Other online sales or management tools
Way of working	Customer: Browsing, searching, ordering Vendor: Uploading products, monitoring orders, responding to customers

1.3.3 Elicitation Process

The elicitation is conducted to collect the requirements needed from interested users and stakeholders in order to tailor a system with an ideal fit. For the Multi-Vendor Ecommerce Platform here is the process:

- Surveys & questionnaires:
- Distributed to potential customers to learn about their shopping habits, frequency and pain points.
- Gathered feedback about what users were looking for in features and expected from a usability standpoint.
- 2 Interviews:
- 2.4 Interviews with Vendors to hit the issues in product management, order processing and customer communication.
- 2.5 Collected user stories on functionality required to operate efficiently.
- 3 Observation:
- 3.4 Researched other OOS online shopping websites for usages and common problem study on user interface, workflow etc.
- 3.5 Noticed some missing features like live chat support and fraud detection.
- 4 Document Analysis:
- 4.4 Read and compared reports, online reviews stats about e-commerce trends and consumer expectations.
- 5 Analysis and Consolidation:
- 5.4 Aggregated data from various sources, used to establish specs (functional/non-functional).
- 5.55 Features are prioritized by user needs and project feasibility ○ .

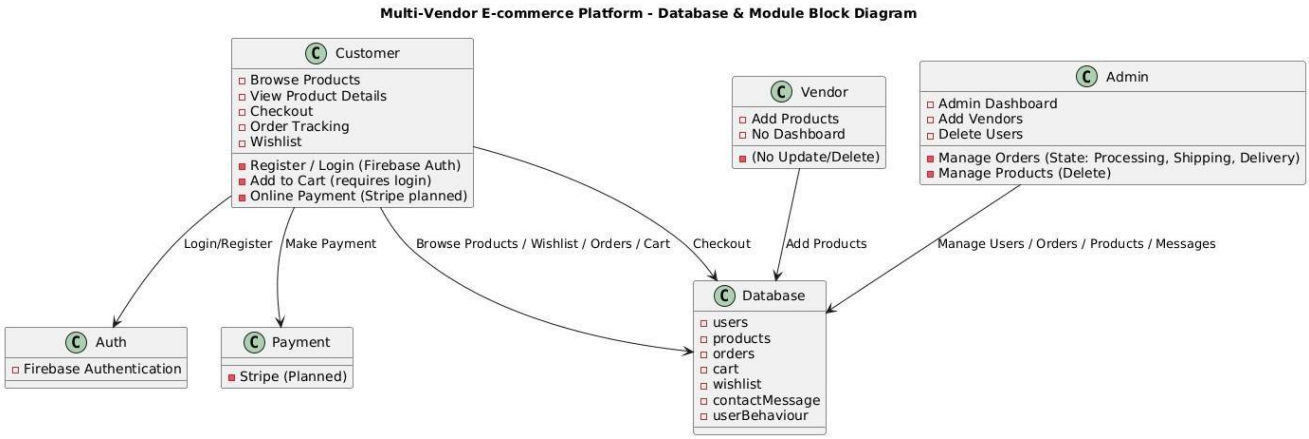
1.4 Project Block Diagram

Figure 1: System Block Diagram

1.5 System Requirements

1.5.1 Hardware Requirements

- Minimum 8 GB RAM, 256 GB storage
- Processor: Intel i5 or equivalent
- Internet connection for development and cloud services



- Optional: Laptop/Desktop with GPU if AI-based features are added

1.5.2 Software Requirements

- **Backend:** Node.js, Express.js, JavaScript
- **Frontend:** React, Tailwind CSS, DaisyUI **Database:** MongoDB
- **Authentication:** Firebase Authentication
- **Payment Integration:** Stripe (planned)
- **Operating System:** Windows 10/11, macOS, or Linux
- **Development Tools:** VS Code, Postman, Git

2. Network Requirements: Stable internet connection for API requests, cloud database, and Firebase

- Port availability for local development (usually 3000 for frontend, 5000 for backend)

3. Other Requirements:

- Browser compatibility (Chrome, Firefox, Edge)
- Optional: Node Package Manager (npm) or Yarn for dependency management

1.5.3 Constraints and Dependencies

Constraints: 1. **Technical**

Constraints:

- Firebase authentication is used; JWT implementation is not included yet. ○ Stripe integration for payments is planned but not yet implemented.
- Vendors cannot update or delete products; no vendor dashboard.

2. **Operational Constraints:**

- Live chat support is not included in the current version. ○ Product approval workflow is limited; admin manually manages users and products.

3. **Time Constraints:** ○ Project timeline must align with academic submission deadlines.
4. **Resource Constraints:**
    - Limited number of developers (solo or small team). ○ Development environment relies on existing hardware and internet availability.

Dependencies: 1. **Software**

Dependencies:

- Node.js, Express.js, React, Tailwind CSS, DaisyUI ○ MongoDB as the database ○ Firebase for authentication ○ Stripe (future payment integration)

2. **External Services:**
    - Firebase for authentication and user management ○ Stripe for online payment processing (planned)
3. **Browser & Platform Dependencies:**
    - Platform accessible on modern browsers (Chrome, Edge, Firefox) ○ Responsive design for desktop, tablet, and mobile devices

1.6 Project Scheduling

1. Time Frame:

The project is planned over **12–14 weeks** with the following milestones:

Phase	Duration	Tasks
Requirement Analysis & Design	2 weeks	Requirement gathering, system design, database design
Backend Development	3–4 weeks	MongoDB setup, API development, Firebase integration
Frontend Development	3–4 weeks	React UI development, Tailwind CSS/DaisyUI styling
Integration & Testing	2 weeks	API integration, functional & usability testing
Deployment & Documentation	1 week	Final deployment, report and documentation

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13

Requirement Analysis & Design	■	■											
Backend Development			■	■	■	■							
Frontend Development							■	■	■	■			
Integration & Testing											■	■	
Deployment & Documentation													

Risk Management

Risk	Likelihood	Impact	Mitigation
Delay in development	Medium	High	Proper planning, task prioritization, buffer time
Technical issues (bugs, API errors)	Medium	Medium	Regular testing, code review, backup environment
Payment integration issues	Low	High	Implement Stripe carefully, test in sandbox mode
Limited user feedback	Medium	Medium	Conduct surveys & test with sample users
Resource limitation	Medium	Medium	Optimize workflow, use opensource tools

1.7 Summary

This section discusses the planning, feasibility and scheduling of the Multi-Vendor Ecommerce Platform project. It starts with problem analysis listing problems within existing e-commerce systems, including ineffective tracking of product, poor control for merchants and the absence of real-time system support. Mission and goal The reason of this project and why you should use it are to create a secure, user-friendly solution that is easy to manage orders and wishlists as well as features needed to manage vendors.

The System Boundary The system includes the features of product browsing, cart management, checkout and Firebase authentication and admin control but excludes live

chat and vendor dashboards for version1. The project feasibility study statement says that the proposed plant is feasible in technical, financial, operational and scheduling terms.

The target users are both the consumers and merchants and their profile/need is identified through surveys, interviews or observation. The chapter also describes the system requirements, limitations, and dependencies necessary for a smooth development. Lastly, a project plan with milestones, Gantt chart and risk management tactics means your project will be done fast & on time.

.

CHAPTER 2 DESIGN AND IMPLEMENTATION

2.1 Introduction

This section will focus on the design and development of Multi-Vendor Ecommerce Platform. It explains how the system architecture, database design and user interfaces are created as per project specification. It also involved the technical implementation (back-end & front-end), such as authentication, product management, order processing and admin console. Through a set of well-tested design principles and excellent yielding practical deployment strategies, the system secures a strong, userintimiating shopping experience between customers, vendors and administrator.

2.2 Functional Requirements :

FR ID	Description	Stakeholder
FR01 – Registration	Customers and vendors must register before using the platform. Registration is handled via Firebase authentication.	Customer, Vendor

FR02 – Login	Customers and vendors must log in to access the platform. Admin login is required to manage users and products.	Customer, Vendor, Admin
FR03 – Add Profile Info	Users can add personal infromation.	Customer, Vendor
FR04 – Update Profile Info	Users can update their profile information anytime.	Customer, Vendor
FR05 – Browse Products	Customers can browse products by category or search.	Customer
FR06 – View Product Details	Customers can view detailed information about a product.	Customer
FR07 – Add to Cart	Customers can add products to the cart. Login is required.	Customer
FR08 – Wishlist	Customers can add products to their wishlist for future reference.	Customer
FR09 – Checkout	Customers can place orders by checking out products from the cart.	Customer
FR10 – Online Payment	Customers can make online payments (Stripe planned) for their orders.	Customer
FR11 – Order Tracking	Customers can track the status of their orders (Processing, Shipping, Delivery).	Customer
FR12 – Add Products	Vendors can add new products to the system.	Vendor
FR13 – Manage Users	Admin can add vendors, delete users, and manage system users.	Admin
FR14 – Manage Products	Admin can delete or approve products in the system.	Admin
FR15 – Manage Orders	Admin can update order states and manage all orders.	Admin

2.3 Non-Functional Requirements

Non-functional requirements define the **quality attributes** and constraints of the system rather than specific behaviors. For the Multi-Vendor E-commerce Platform, the key nonfunctional requirements include:

Performance:

The platform should support more than one user at a time with no perceivable lag.

As the product search and order processing should be instantaneous.

Reliability:

The system needs to be highly available (HA), and must exhibit minimal downtime.

It is important to keep order, payment and customer data in an as did not hacked state.

Security:

Firebase User Authentication needs to ensure the protection of account details.

Online payments (Stripe) need to be secured and encrypted.

Usability:

User interaction should form intuitiveness and smoothness on desktops, tablets, and smartphones alike.

Customers and Vendors - Minimal learning curve.

Maintainability:

We need an easy way to update the features, database and UI without having an impact onto other modules.

Scalability:

The platform needs to scale as the number of users, products and sellers increase.

2.3.1 Performance

The Multi-Vendor E-commerce Platform must provide fast and resposi ve to ensure a smooth user experience. Key performance requirements include:

- **Response Time:** Pages, product searches, and order processing should load within 2– 3 seconds under normal usage.
- **Concurrency:** The system should support multiple simultaneous users without slowdowns.
- **Efficiency:** Backend APIs and database queries should be optimized to handle high traffic efficiently.

- **Resource Usage:** The system should efficiently use server and database resources to prevent bottlenecks.

2.3.2 Reliability

Key reliability requirements include:

- **Availability:** The system should be accessible to users 24/7 with minimal downtime.
- **Data Integrity:** All transactions, orders, and user data must be accurately recorded and protected from corruption.
- **Error Handling:** The system should gracefully handle errors without crashing and provide meaningful messages to users.
- **Backup & Recovery:** Regular database backups and recovery mechanisms should be in place to prevent data loss.

2.3.3 Portability

Key points include:

- **Cross-Platform Compatibility:** The platform should work on various operating systems such as Windows, macOS, and Linux.
- **Browser Compatibility:** It should be accessible via major web browsers like Chrome, Firefox, and Edge.
- **Responsive Design:** The system should function properly on desktops, tablets, and mobile devices.
- **Ease of Deployment:** The platform should be easy to deploy or migrate to different servers or cloud environments.

2.4 Object-oriented System design using UML

2.4.1 Use Case Diagram

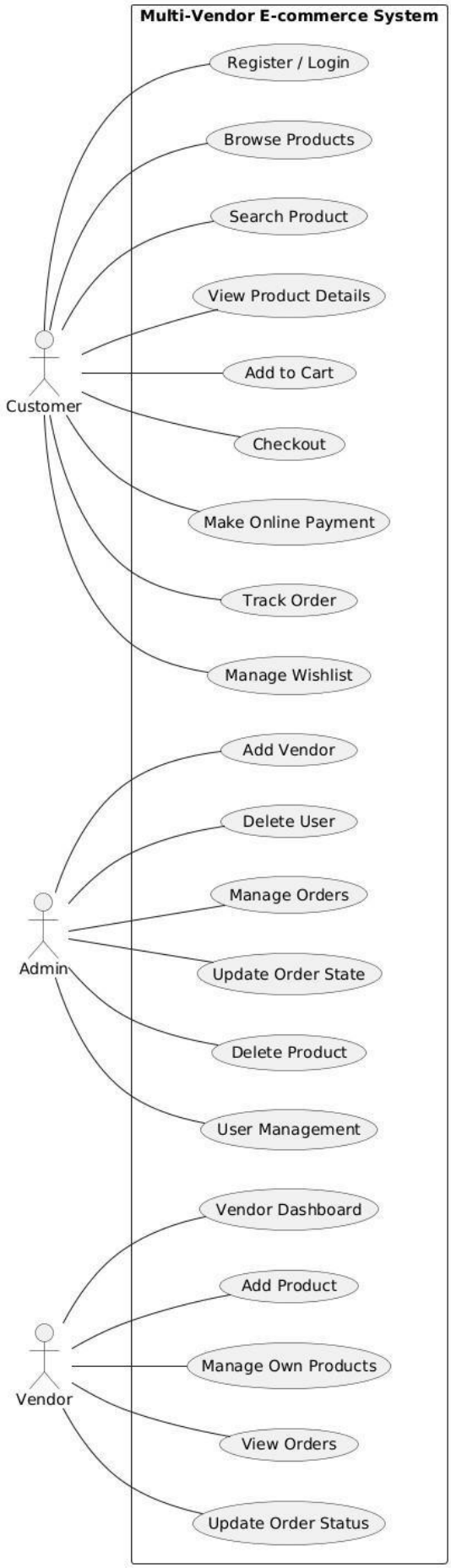


Figure 2: Use case Diagram

2.4.2 Case Description

Case Description-01: Registration

<table>
<tr><td>Use Case</td><td>Registration</td></tr>
<tr><td>Goal</td><td>Users can register and create an account to access the system.</td></tr>
<tr><td>Precondition</td><td>User must have internet access and open the Multi-Vendor Ecommerce website/app.</td></tr>
<tr><td>Success End Condition</td><td>Notification: “Successfully Registered!”</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Registration Failed”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer
Firebase Authentication</td></tr>
<tr><td>Trigger</td><td>User clicks on the “Register” button to create a new account.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User presses the “Register” button.</td></tr>
<tr><td>2.</td><td>System provides the Registration Form.</td></tr>
<tr><td>3.</td><td>User enters information (Name, Email, Password).</td></tr>
<tr><td>4.</td><td>Press “Submit” Button.</td></tr>
<tr><td>5.</td><td>Information saved</td></tr>
<tr><td>6.</td><td>System displays notification: “Successfully Registered!”</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a. Try Again!!</td></tr>
<tr><td>4.1</td><td>The user Did not fill up the details!</td></tr>
<tr><td></td><td>4.1.a. Checked By the system & Notify by “Please! Fill Up the Box”.</td></tr>
<tr><td>5.1</td><td>Email already exists</td></tr>
<tr><td></td><td>5.1.a. System notifies: “Registration Failed – Email Already Exists”</td></tr>
</table>
</td></tr>
</table>

	6.1	The system Doesn't save the details.
		6.1.a. Notification: "Details did not Save"
Quality Requirements	• Registration must complete within 15 minutes. • System should respond within 3–5 seconds.	

Case Description-02: Login

Use Case	Login	
Goal	Users can sign in to the system using valid credentials.	
Precondition	User must have a registered account and active internet connection.	
Success End Condition	Notification: "Login Successful!"	
Failed End Condition	Failed End Condition: Notification: "Login Failed!""	
Primary Actors: Secondary Actors:	Customer Firebase Authentication	
Trigger	User presses the Login button	
Description / Main Success Scenario	1.	User opens Login Page.
	2.	User enters Email and Password.
	3.	User enters information (Name, Email, Password).
	4.	User presses Sign In.
	5.	System sends credentials to Firebase Auth.
	6.	Firebase Auth validates credentials.
	7.	System displays notification: "Login Successful!"

<table>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Empty email/password field</td></tr>
<tr><td></td><td>1.1.a. System notifies: “Please fill in all fields!”</td></tr>
<tr><td>4.1</td><td>Network issue</td></tr>
<tr><td></td><td>4.1.a. “Network Error! Try Again.”</td></tr>
<tr><td>5.1</td><td>Invalid credentials</td></tr>
<tr><td></td><td>5.1.a. “Incorrect Email or Password!”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Login must complete within 15 minutes.
• System should respond within 3–5 seconds.</td></tr>
</table>

Case Description-03: Browse Products

Use Case	Browse Products
Goal	Users can browse and explore available products.
Precondition	Products must exist in the database and user must have internet access.
Success End Condition	Product list displayed successfully.
Failed End Condition	Notification: “Failed to Load Products!”
Primary Actors: Secondary Actors:	Customer System, MongoDB Database
Trigger	User opens the website/app home page or product page.

<table>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User opens the website/app.</td></tr>
<tr><td>2.</td><td>System fetches all available products from the database.</td></tr>
<tr><td>3.</td><td>System displays products with price, images, and categories.</td></tr>
<tr><td>4.</td><td>User scrolls and explores different products.</td></tr>
<tr><td>5.</td><td>User selects a product to view details.</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>Database not responding</td></tr>
<tr><td></td><td>1.1.a. System shows: “Unable to Load Products!”</td></tr>
<tr><td>4.1</td><td>No products available</td></tr>
<tr><td></td><td>4.1.a. System notifies: “No Products Found!”</td></tr>
<tr><td>5.1</td><td>Slow network</td></tr>
<tr><td></td><td>5.1.a. System shows loader animation until data loads.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>• Product list should load within 3 seconds.
• Images must load smoothly and be optimized.
• Users must be able to scroll without lag.</td></tr>
</table>

Case Description-04: Search Products

Use Case	Search Products
Goal	sers can search for products using keywords.
Precondition	Products must exist in the database, and user must be on the browse page.
Success End Condition	Relevant products displayed based on keyword.

<table>
<tr><td>Failed End Condition</td><td>Notification: “No Products Found!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>User types a keyword in the search bar.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User types a keyword in the search bar.</td></tr>
<tr><td>2.</td><td>System fetches matching products from the database.</td></tr>
<tr><td>3.</td><td>System displays search results with product name, image, price, and category.</td></tr>
<tr><td>4.</td><td>User can select a product to view details.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>No matching products</td></tr>
<tr><td></td><td>1.1.a. System shows: “No Products Found!”</td></tr>
<tr><td>4.1</td><td>Network error</td></tr>
<tr><td></td><td>4.1.a. System shows: “Unable to fetch results. Try Again.”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Search result must load within 3 seconds.
• Search must return accurate results.</td></tr>
</table>

Case Description-

05: View Product Details

Use Case	View Product Details
Goal	Users can view full information about a product.
Precondition	Product must exist in the database, and user must select a product from the list.
Success End Condition	Product details displayed successfully.
Failed End Condition	Notification: “Failed to Load Product Details!”

<table>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>User clicks on a product.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User clicks a product from browse/search results.</td></tr>
<tr><td>2.</td><td>System fetches product details from the database.</td></tr>
<tr><td>3.</td><td>System displays product image, name, price, stock, description, and category.</td></tr>
<tr><td>4.</td><td>User sees the Add to Cart option.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Product not found</td></tr>
<tr><td></td><td>1.1.a. “Product Not Available!”</td></tr>
<tr><td>4.1</td><td>Database error</td></tr>
<tr><td></td><td>4.1.a. “Error Loading Product!”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Details must load within 2–4 seconds.
• All information clearly visible.</td></tr>
</table>

Case Description-06: Add to Cart

Use Case	Add to Cart
Goal	Users can add a selected product to their shopping cart.
Precondition	User must be logged in, and product must be in stock.
Success End Condition	Notification: “Product Added to Cart!”
Failed End Condition	Notification: “Failed to Add Product!”
Primary Actors: Secondary Actors:	Customer System, MongoDB Database

<table>
<tr><td>Trigger</td><td>User clicks Add to Cart button.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User clicks Add to Cart.</td></tr>
<tr><td>2.</td><td>System checks if the user is logged in.</td></tr>
<tr><td>3.</td><td>System adds the product to the cart in MongoDB.</td></tr>
<tr><td>4.</td><td>Cart is updated, and notification is displayed.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>User not logged in</td></tr>
<tr><td></td><td>1.1.a. Redirect to login & notify: “Please Login to Add Items!”</td></tr>
<tr><td>4.1</td><td>Database error</td></tr>
<tr><td></td><td>4.1.a. “Database Error! Try Again”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Cart update must complete within 3–5 seconds.
• Data must be securely saved in database.</td></tr>
</table>

07: Checkout

Use Case	Checkout
Goal	Users can place an order for items in the cart.

Case Description-

Precondition	Cart has items, and user is logged in.
Success End Condition	Order summary displayed, ready for payment.
Failed End Condition	Notification: “Checkout Failed!”
Primary Actors: Secondary Actors:	Customer System, MongoDB Database
Trigger	User clicks Checkout.

<table>
<tr><td>Description / Main
Success
Scenario</td><td><table>
<tr><td>1.</td><td>User opens checkout page.</td></tr>
<tr><td>2.</td><td>System shows cart items and total amount.</td></tr>
<tr><td>3.</td><td>User enters shipping information.</td></tr>
<tr><td>4.</td><td>User confirms the order.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Shipping info empty</td></tr>
<tr><td></td><td>1.1.a. “Please fill shipping info!”</td></tr>
<tr><td>4.1</td><td>Database error</td></tr>
<tr><td></td><td>4.1.a. “Unable to fetch cart details!”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Checkout must complete within 5 seconds.
• Shipping info must be validated.</td></tr>
</table>

Case Description-

08: Online Payment

<table>
<tr><td>Use Case</td><td>Online Payment</td></tr>
<tr><td>Goal</td><td>Users can pay for their order online.</td></tr>
<tr><td>Precondition</td><td>Checkout must be completed.</td></tr>
<tr><td>Success End Condition</td><td>Payment successful & order confirmed.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Payment Failed!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Customer
Payment Gateway (Stripe)</td></tr>
<tr><td>Trigger</td><td>User selects online payment method.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User selects Online Payment.</td></tr>
<tr><td>2.</td><td>System initiates Stripe PaymentIntent.</td></tr>
<tr><td>3.</td><td>User enters payment details.</td></tr>
<tr><td>4.</td><td>System updates order status & notifies user.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Payment declined</td></tr>
<tr><td></td><td>1.1.a. “Payment Failed!”</td></tr>
<tr><td>4.1</td><td>Network issue</td></tr>
<tr><td></td><td>4.1.a. “Payment Timeout. Try Again!”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Payment process ≤ 5 seconds.
• Transactions must be secure & PCI-compliant.</td></tr>
</table>

-

Case Description

09: Track Order

Use Case	Track Order
Goal	Users can track the status of their orders.
Precondition	Order must exist for the user.
Success End Condition	Order status displayed successfully.
Failed End Condition	Notification: “Unable to Track Order!”
Primary Actors: Secondary Actors:	Customer System, MongoDB Database
Trigger	User clicks Track Order**.**
Description / Main Success Scenario	1. User opens Track Order page. 2. System fetches order status from database. 3. System displays current status: Processing → Shipping → Delivery.

Case Description-

Alternative Flows	<table><tr><td>1.1</td><td>Order not found</td></tr><tr><td></td><td>1.1.a. “Order Not Found!”</td></tr><tr><td>4.1</td><td>Network/database issue</td></tr><tr><td></td><td>4.1.a. “Unable to fetch order status.”</td></tr></table>
Quality Requirements	• Status must load ≤ 3 seconds. • Status updates must be accurate.

10: Wishlist

Use Case	Wishlist
Goal	Users can add products to their wishlist.
Precondition	User must be logged in.
Success End Condition	Notification: “Product Added to Wishlist!”
Failed End Condition	Notification: “Failed to Add Product!”
Primary Actors: Secondary Actors:	Customer System, MongoDB Database
Trigger	User clicks Add to Wishlist.

-

<table>
<tr>
<td>Description / Main Success Scenario</td>
<td>
<table>
<tr><td>1.</td><td>User clicks Add to Wishlist.</td></tr>
<tr><td>2.</td><td>System adds product to wishlist in database.</td></tr>
<tr><td>3.</td><td>System shows notification.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Alternative Flows</td>
<td>
<table>
<tr><td>1.1</td><td>User not logged in</td></tr>
<tr><td></td><td>1.1.a. Redirect to login.</td></tr>
<tr><td>4.1</td><td>Network/database issue</td></tr>
<tr><td></td><td>4.1.a. “Unable to add product.”</td></tr>
</table>
</td>
</tr>
<tr>
<td>Quality Requirements</td>
<td>Wishlist update ≤ 3 seconds. Data securely stored.</td>
</tr>
</table>

-

Case Description 11: Add Vendor (Admin)

Use Case	Add Vendor (Admin)
Goal	Admin can add a new vendor.
Precondition	Admin logged in.
Success End Condition	Notification: “Vendor Added Successfully!”
Failed End Condition	“Failed to Add Vendor!”
Primary Actors: Secondary Actors:	Admin System, MongoDB Database
Trigger	Trigger: Admin clicks Add Vendor**.**
Description / Main Success Scenario	1. Admin opens Add Vendor page. 2. Admin fills vendor information. 3. System saves vendor in database.
Alternative Flows	1.1 Required fields empty 1.1.a. “Please fill all fields!” 4.1 Network/database issue 4.1.a. “Unable to update role.”
Quality Requirements	• Operation ≤ **5 seconds**. • Vendor info must be securely stored.

12: Manage Users (Admin)

Use Case	Manage Users (Admin)

Case Description-

<table>
<tr><td>Goal</td><td>Admin can view,role update and delete users.</td></tr>
<tr><td>Precondition</td><td>Admin logged in.</td></tr>
<tr><td>Success End Condition</td><td>User deleted successfully or updated.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Action Failed!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Admin
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>Admin selects Manage Users.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Admin opens Manage Users page.</td></tr>
<tr><td>2.</td><td>System displays all registered users.</td></tr>
<tr><td>3.</td><td>Admin selects user to delete.</td></tr>
<tr><td>4.</td><td>System removes user and confirms.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>User not found</td></tr>
<tr><td></td><td>1.1.a. “User Not Found!”</td></tr>
<tr><td>4.1</td><td>Network/database issue</td></tr>
<tr><td></td><td>4.1.a. “Unable to process request.”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Action ≤ 3 seconds.
• Data securely updated.</td></tr>
</table>

Case Description-

13: Manage Orders (Admin)

<table>
<tr><td>Use Case</td><td>Manage Orders (Admin)</td></tr>
<tr><td>Goal</td><td>Manage Orders</td></tr>
<tr><td>Precondition</td><td>Orders exist.</td></tr>
<tr><td>Success End Condition</td><td>Order status updated.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Failed to Update Order!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Admin
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>Admin opens Manage Orders page.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Admin opens Manage Orders page from dashboard</td></tr>
<tr><td>2.</td><td>System shows all orders.</td></tr>
<tr><td>3.</td><td>Admin selects order and updates status: Processing → Shipping → Delivered.</td></tr>
<tr><td>4.</td><td>System saves update and confirms.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>No orders found</td></tr>
<tr><td></td><td>1.1.a. “No Orders Available!”</td></tr>
<tr><td>4.1</td><td>Database error</td></tr>
<tr><td></td><td>4.1.a. “Unable to update order.”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Update ≤ 3 seconds.
• Status changes must reflect accurately.</td></tr>
</table>

14:Delete Product (Admin)

Case Description-

<table>
<tr><td>Use Case</td><td>Delete Product (Admin)</td></tr>
<tr><td>Goal</td><td>Admin can delete any product from the system.</td></tr>
<tr><td>Precondition</td><td>Product exists.</td></tr>
<tr><td>Success End Condition</td><td>Product deleted successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Failed to Delete Product!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Admin
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>Admin clicks Delete Product.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Admin opens Product List from dashboard</td></tr>
<tr><td>2.</td><td>Admin selects a product.</td></tr>
<tr><td>3.</td><td>Clicks Delete.</td></tr>
<tr><td>4.</td><td>System removes product from database and confirms.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Product not found</td></tr>
<tr><td></td><td>1.1.a. “Product Not Available!”</td></tr>
<tr><td>4.1</td><td>Database error</td></tr>
<tr><td></td><td>4.1.a. “Unable to delete product.”</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>• Delete operation ≤ 3 seconds.
• Database must maintain integrity.</td></tr>
</table>

Case Description-

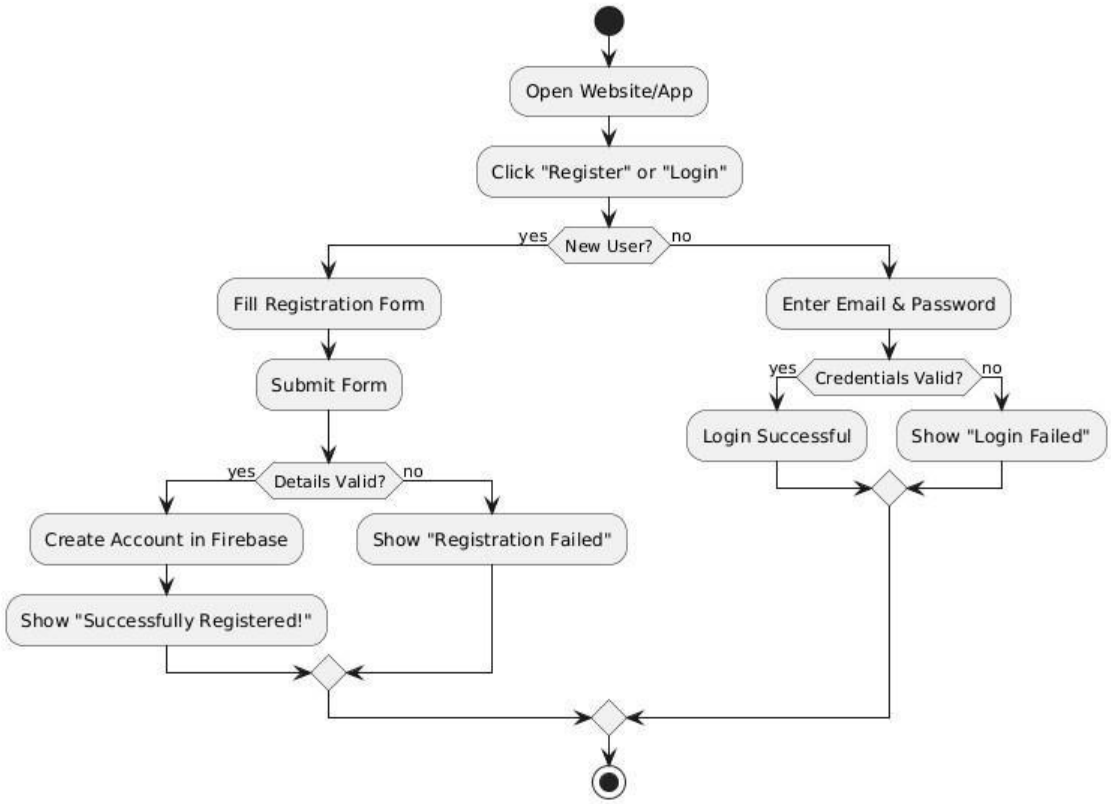
Case Description

-15: Add Product (Vendor)

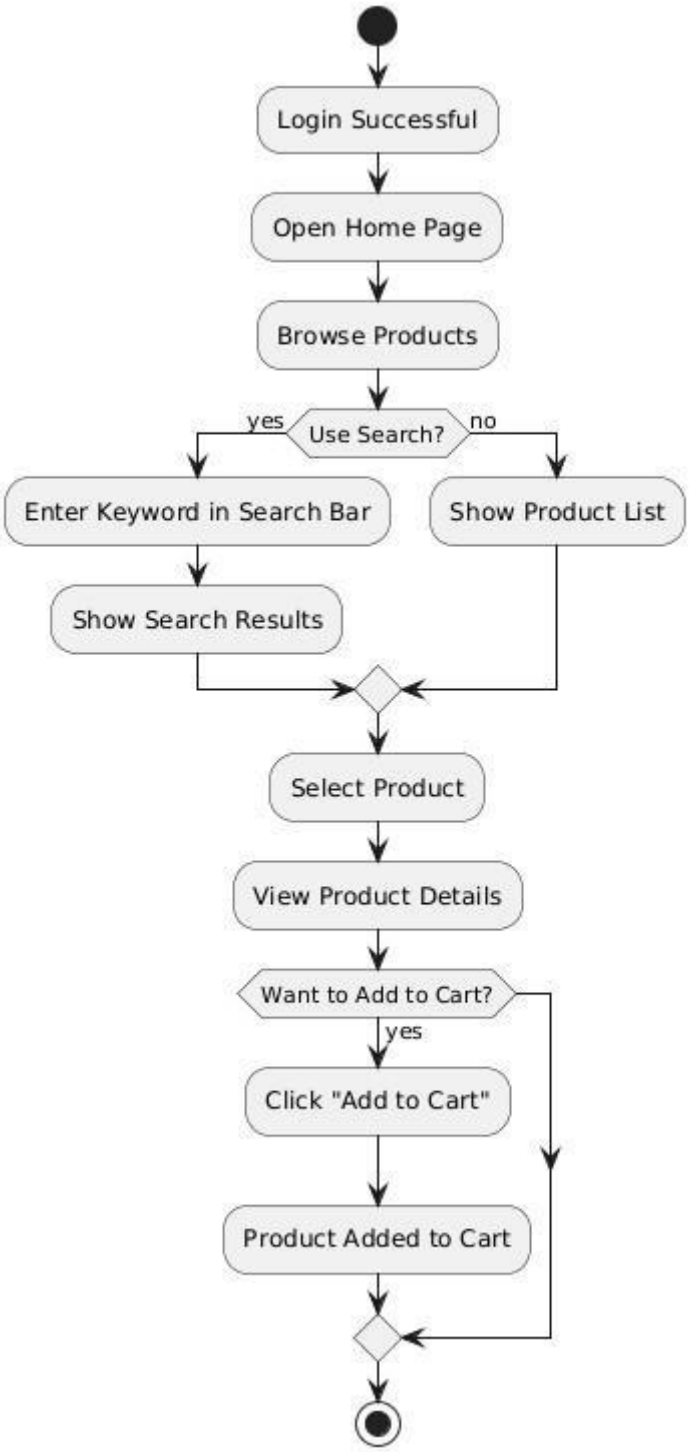
<table>
<tr><td>Use Case</td><td>Add Product (Vendor)</td></tr>
<tr><td>Goal</td><td>Vendor can add new products.</td></tr>
<tr><td>Precondition</td><td>Vendor logged in.</td></tr>
<tr><td>Success End Condition</td><td>Notification: “Product Added Successfully!”</td></tr>
<tr><td>Failed End Condition</td><td>Notification: “Failed to Add Product!”</td></tr>
<tr><td>Primary Actors:
Secondary Actors:</td><td>Admin
System, MongoDB Database</td></tr>
<tr><td>Trigger</td><td>Vendor clicks Add Product.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>Vendor opens Add Product page.</td></tr><tr><td>2.</td><td>Vendor fills product details: name, price, stock, category, image.</td></tr><tr><td>3.</td><td>Vendor submits form.</td></tr><tr><td>4.</td><td>System saves product in database and confirms.</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1.1</td><td>Required fields empty</td></tr><tr><td></td><td>1.1.a. “Please fill all fields!”</td></tr><tr><td>4.1</td><td>Database/network error</td></tr><tr><td></td><td>4.1.a. “Unable to add product.”</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>• Add product ≤ 5 seconds.
• Product data stored securely.</td></tr>
</table>

2.4.3 Activity Diagram

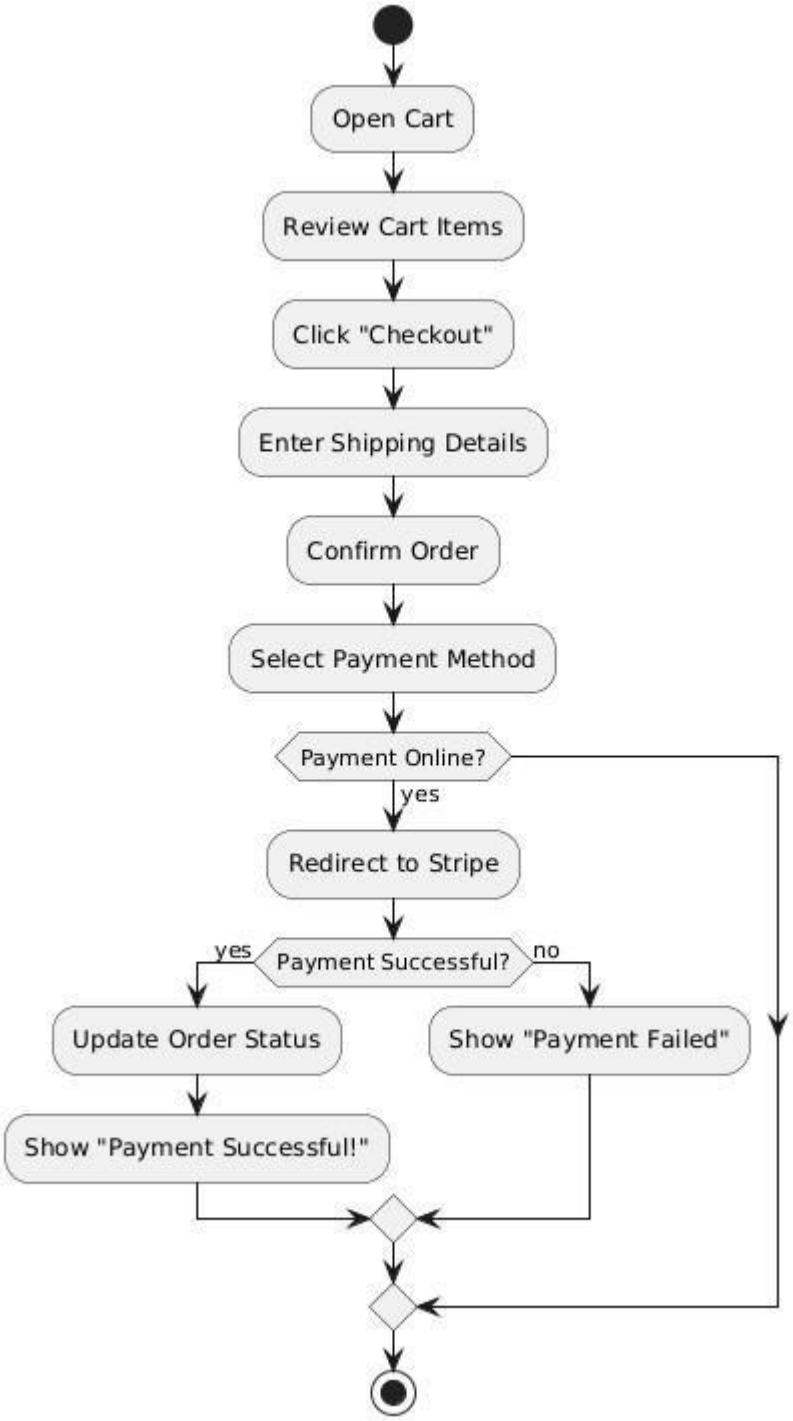
1.Customer – Registration & Login Activity Diagram:



2.Customer Browse, Search & View Product Activity Diagram:

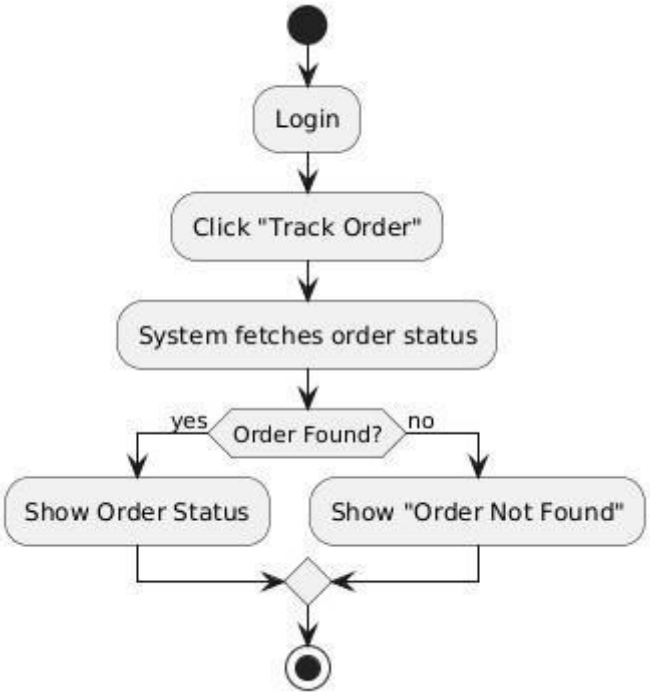


3.Customer Checkout & Payment Activity Diagram:

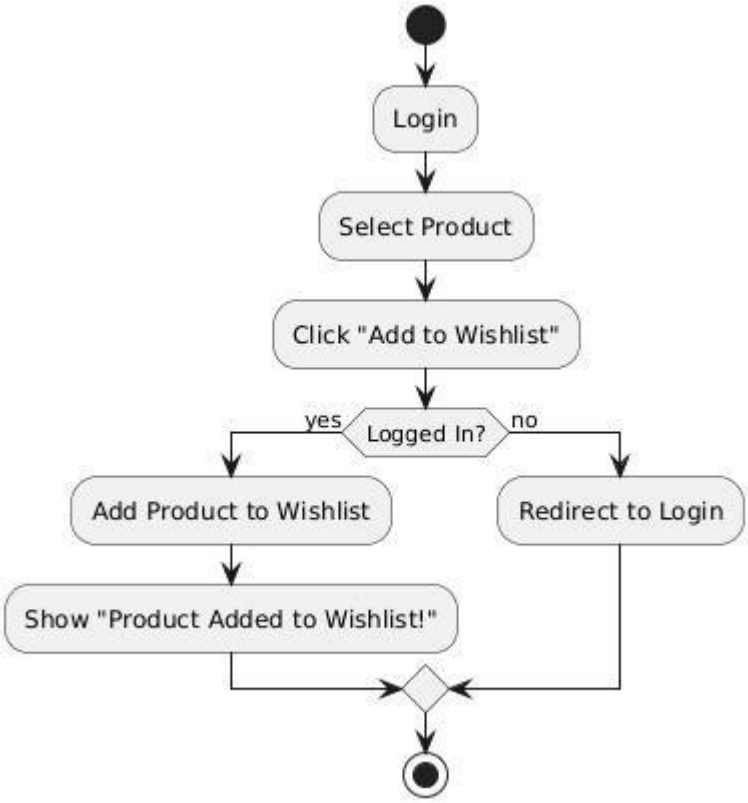


–

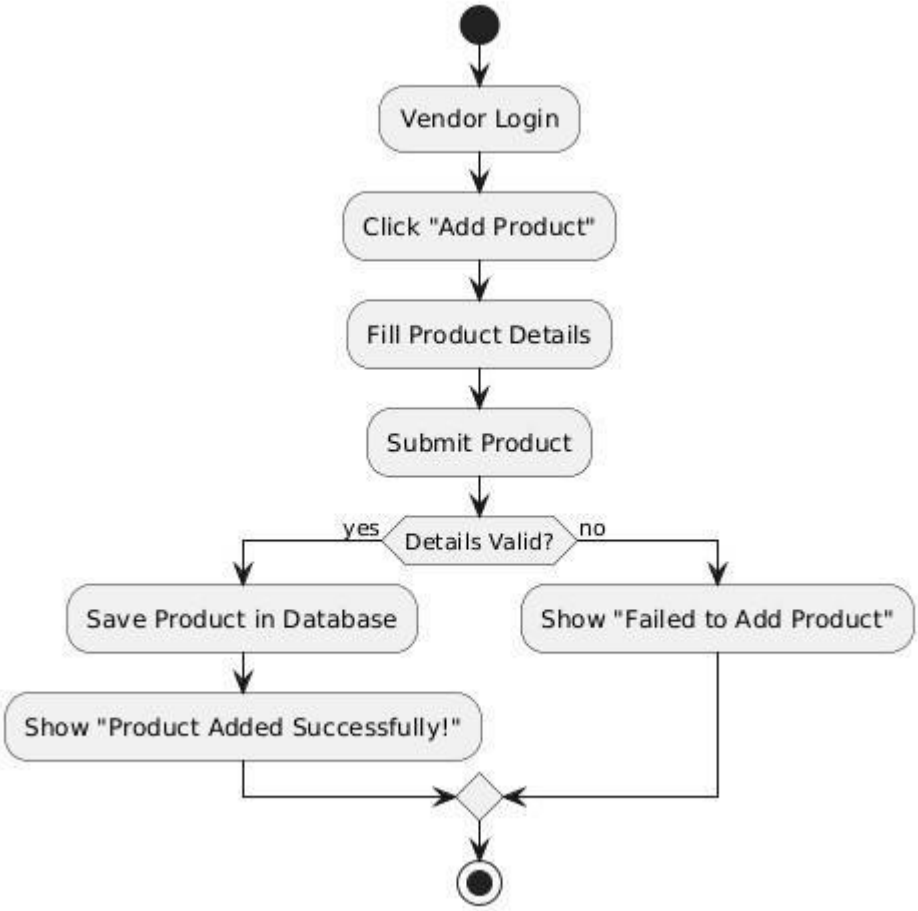
4. Customer Track Order Activity Diagram:



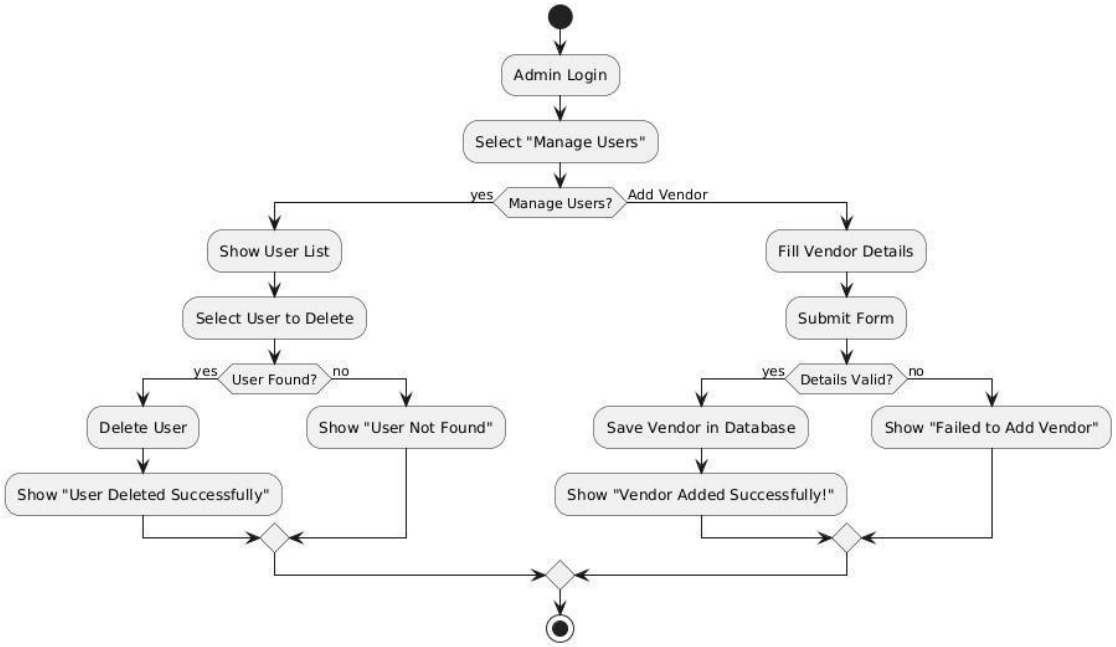
5.Customer – Wishlist Activity Diagram:



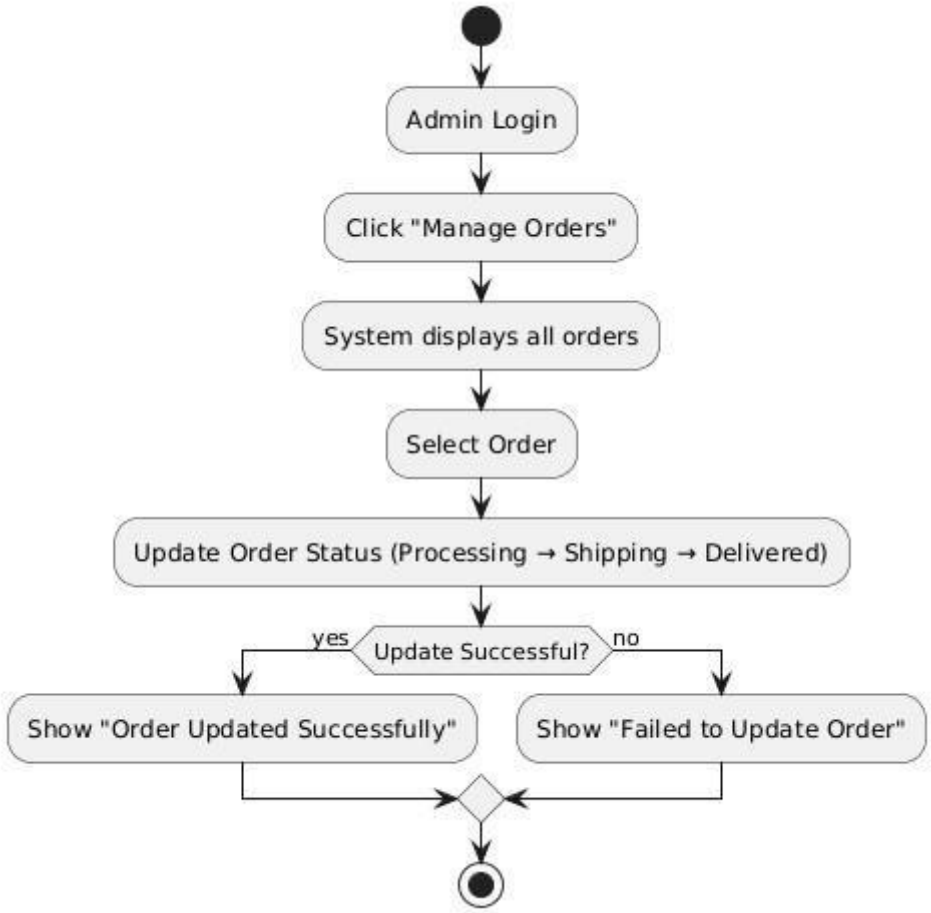
6. Vendor – Add Product Activity Diagram:



7. Manage Users & Vendors Activity Diagram:



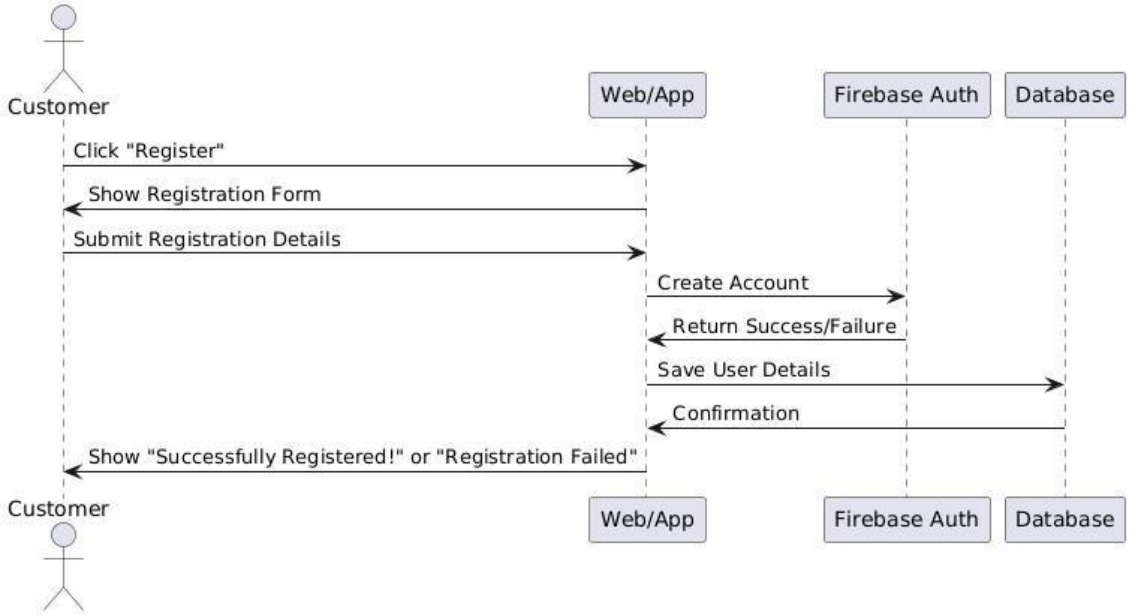
8. Admin – Manage Orders Activity Diagram:



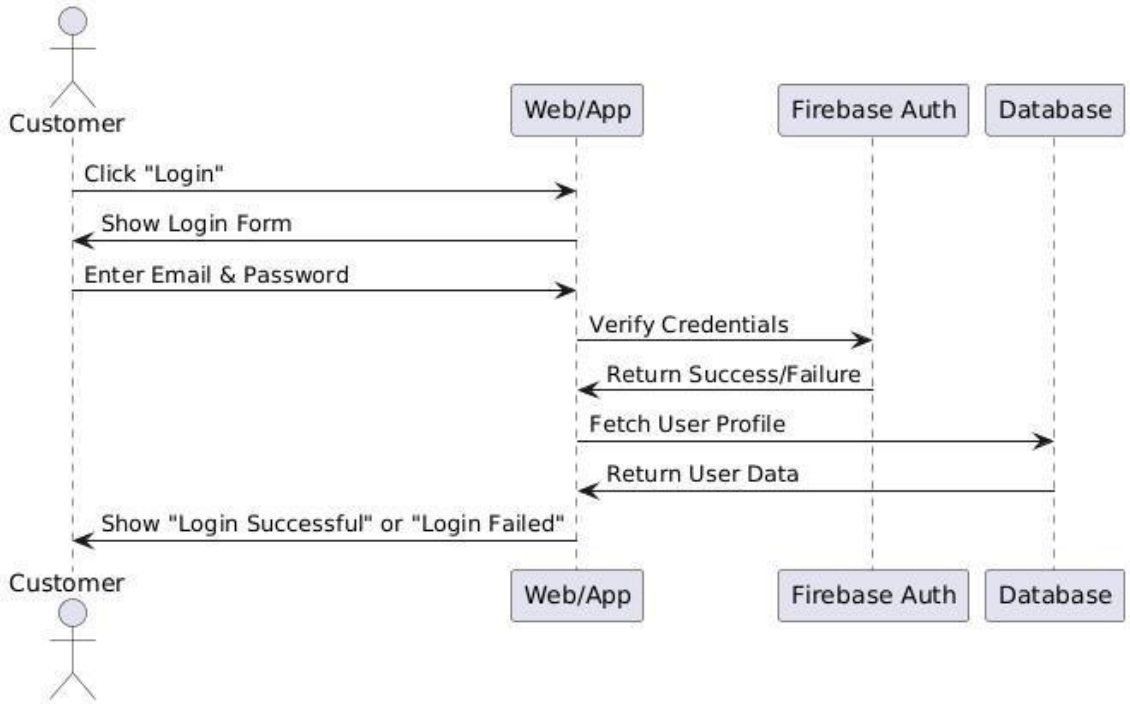
2.4.4 Sequence Diagram

1.Customer – Registration Sequence Diagram:

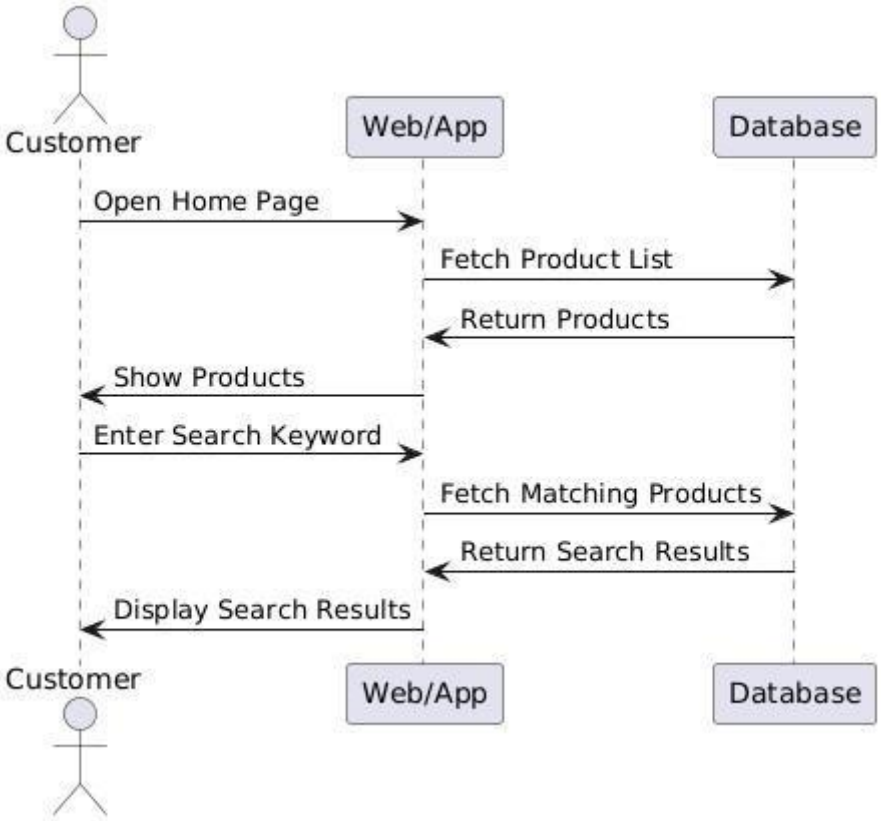
Customer



2. Customer – Login Sequence Diagram:

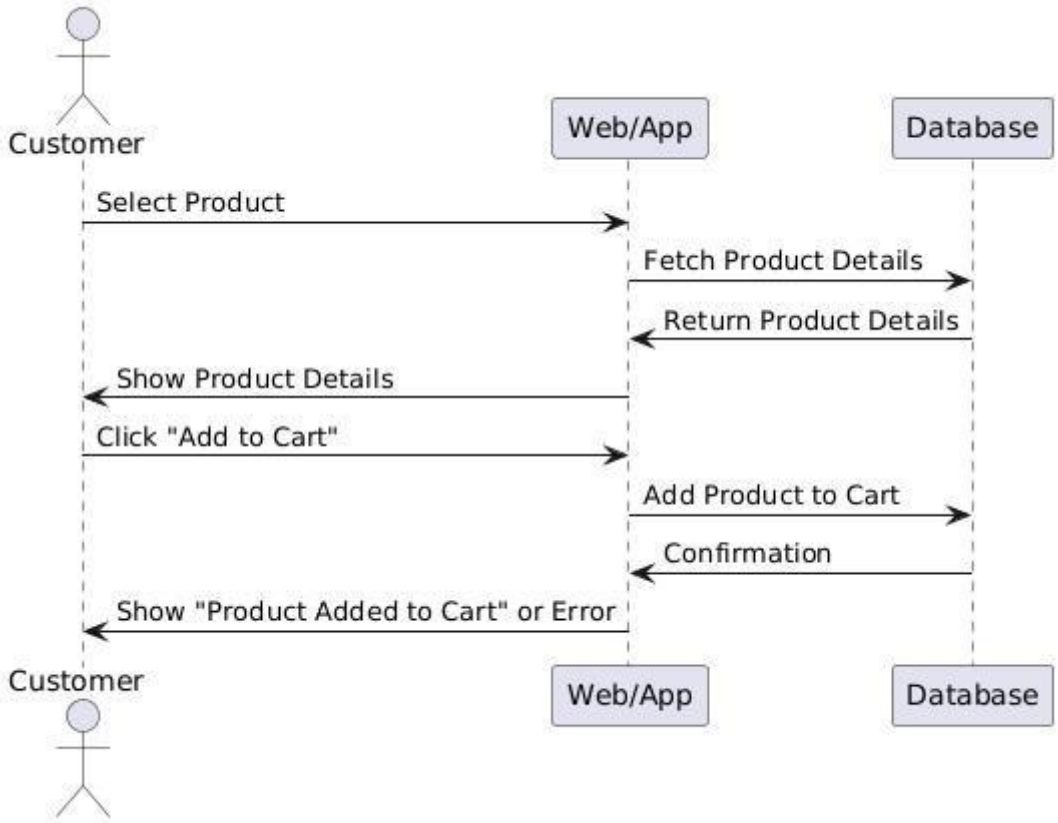


3. Browse & Search Products Sequence Diagram:



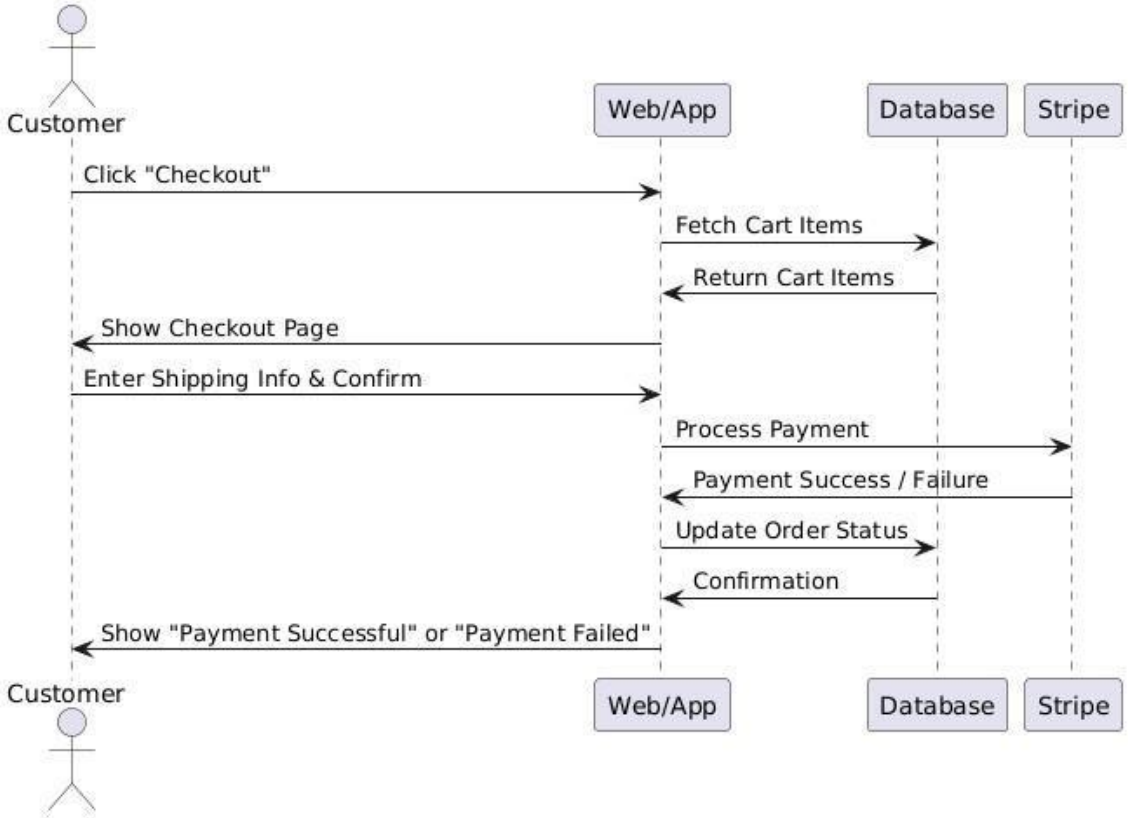
Customer –

4. View Product Details & Add to Cart Sequence Diagram:

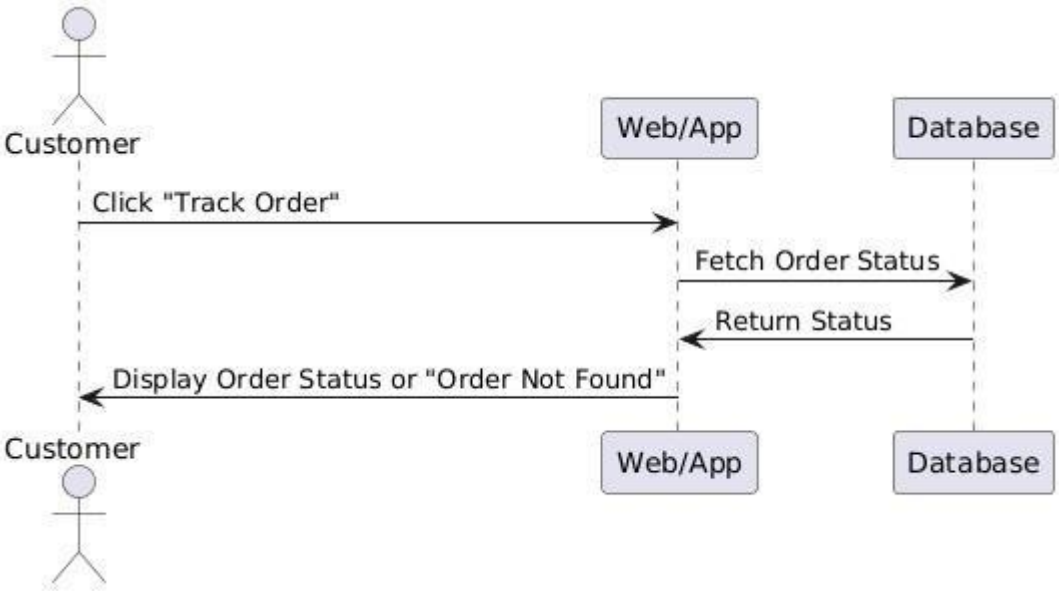


5. Checkout & Online Payment Sequence Diagram:

Customer –

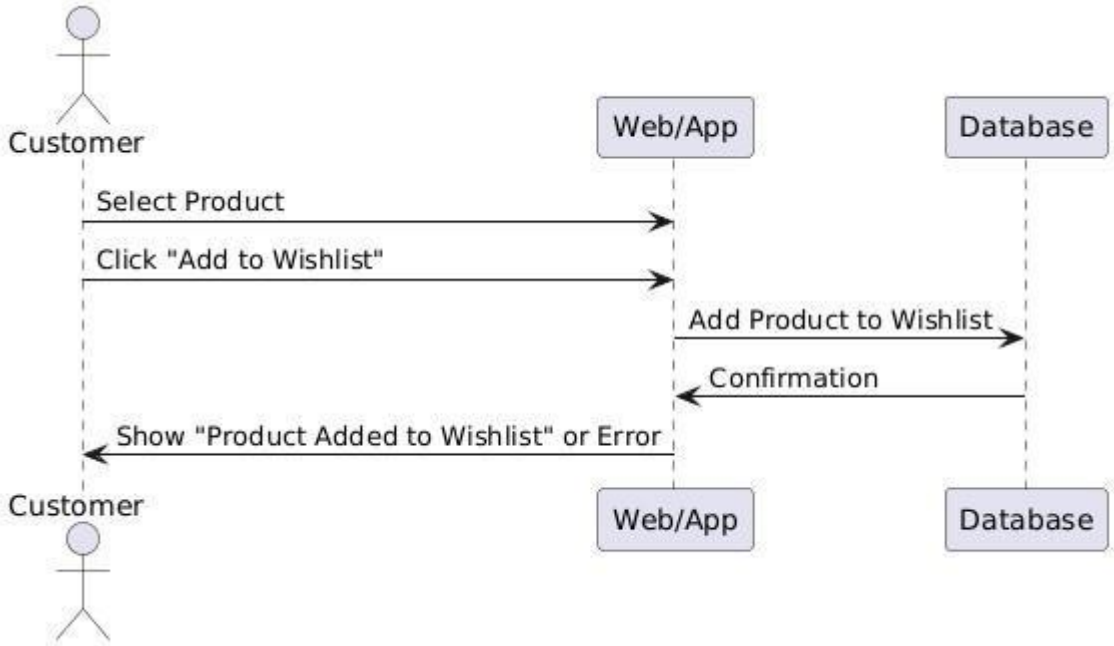


6. Customer – Track Order Sequence Diagram:

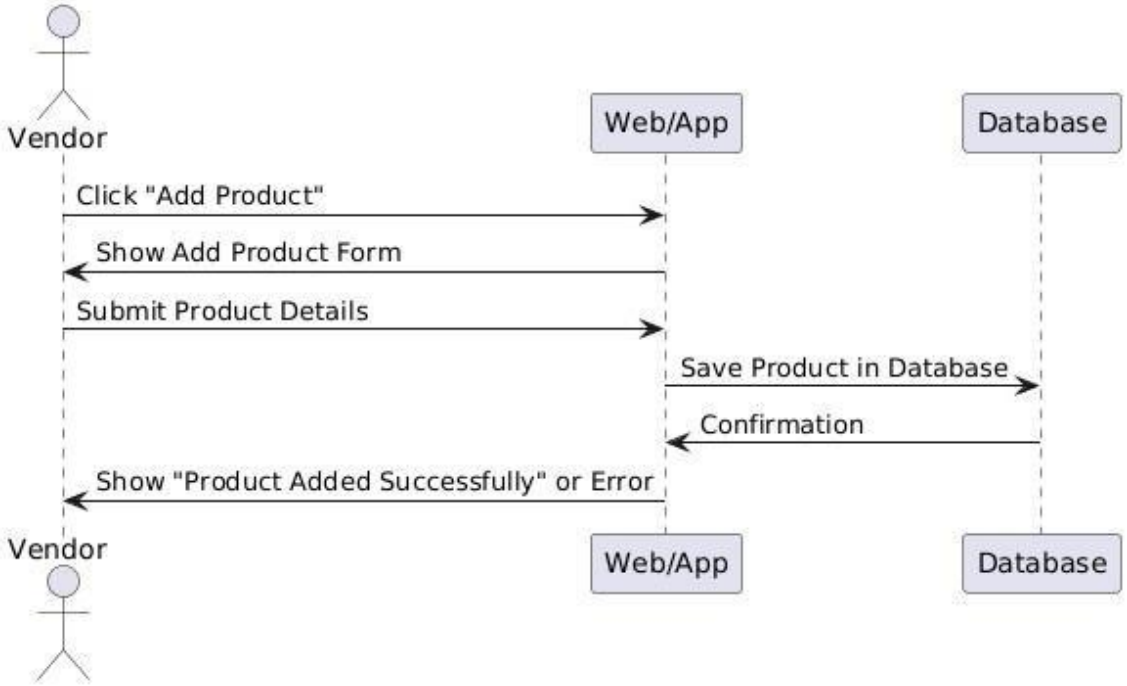


Customer –

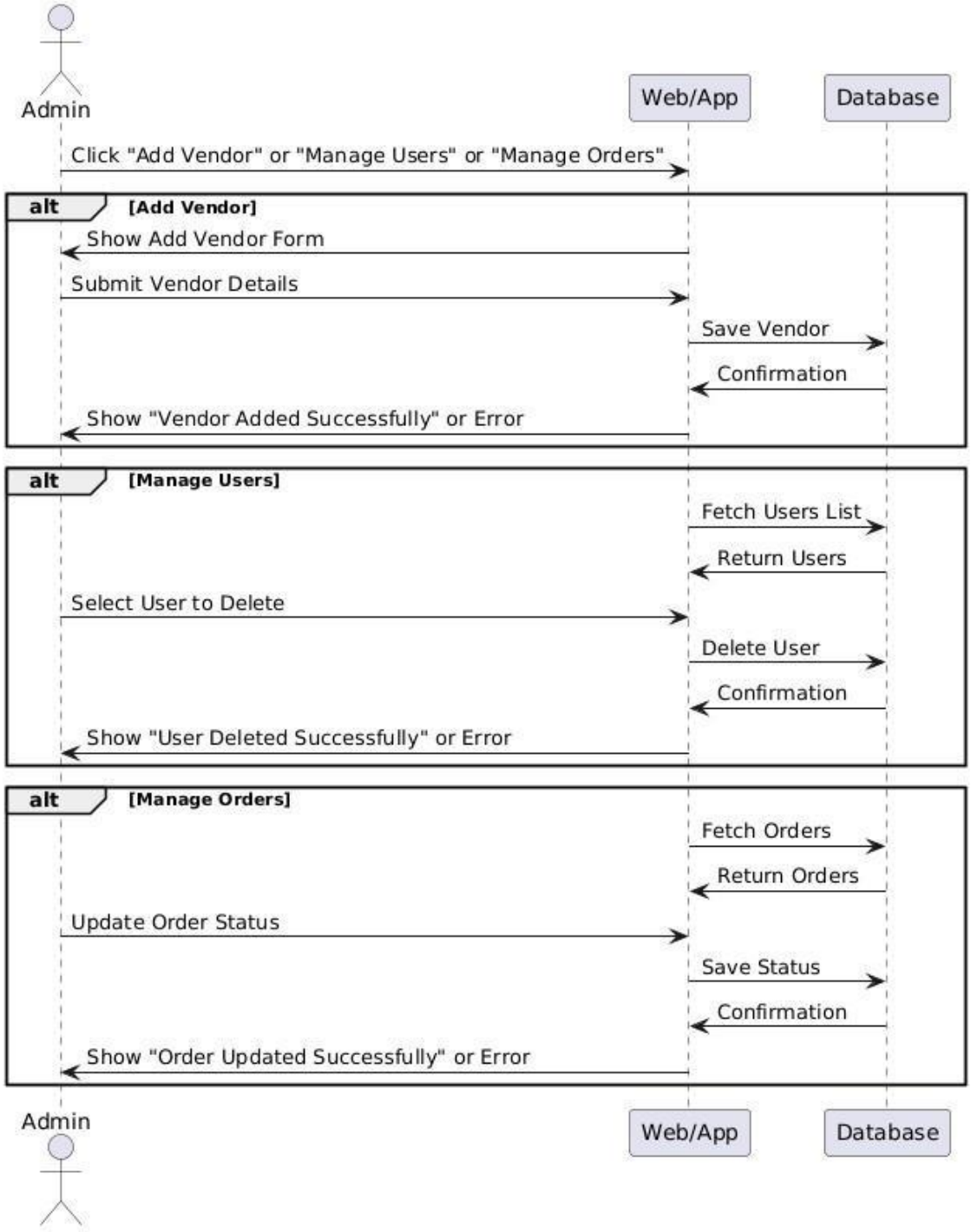
7. Wishlist Sequence Diagram:



8. Vendor – Add Product Sequence Diagram:



9. Admin –Sequence Diagram:



2.4.5 Class Diagram

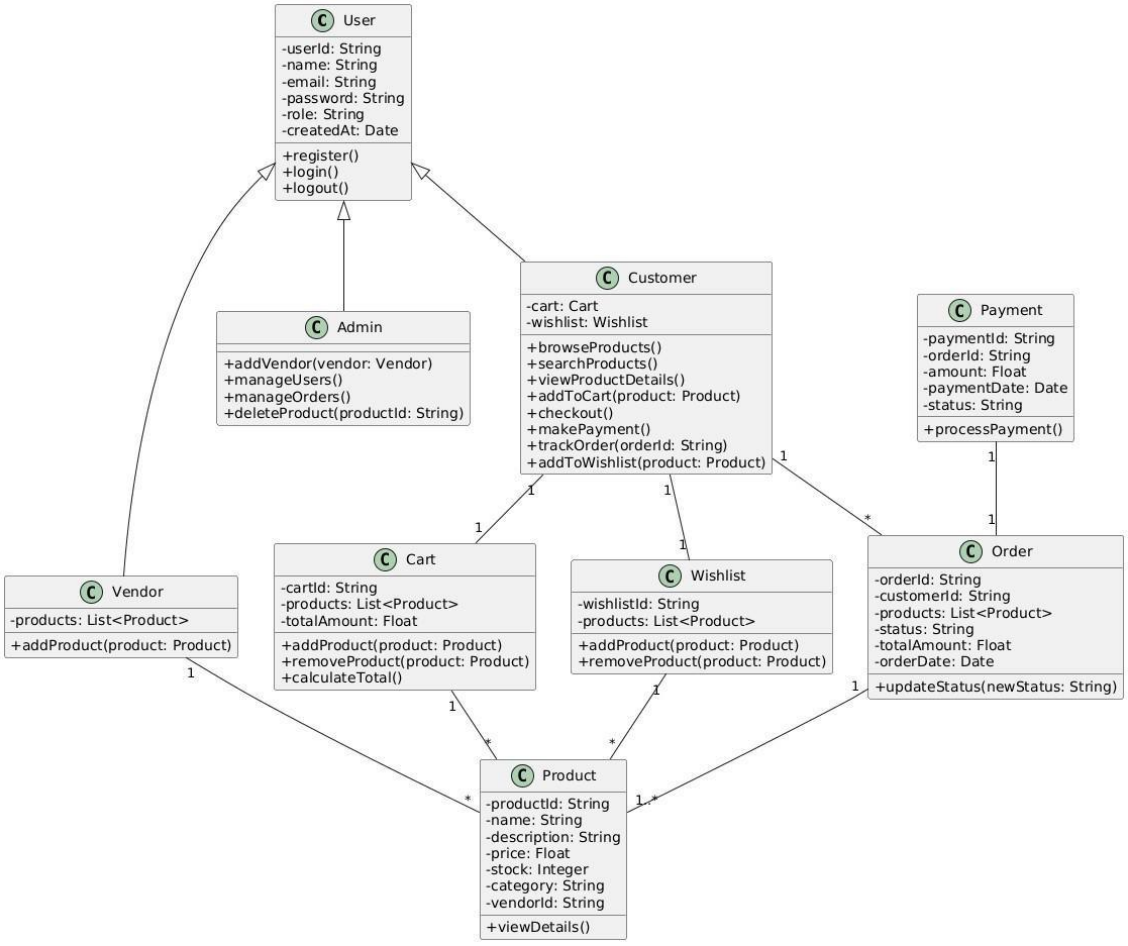
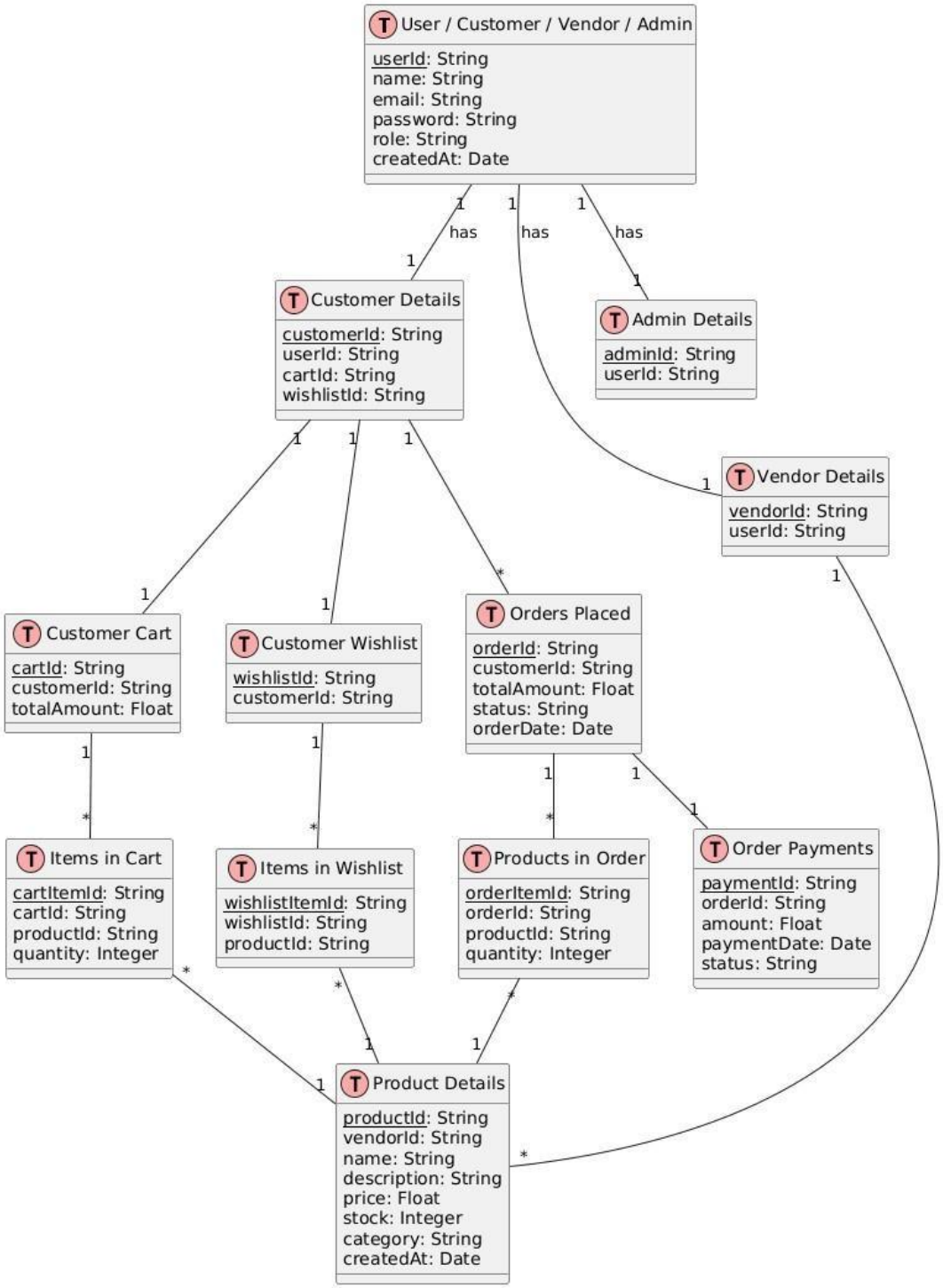


Figure 4: Class Diagram

2.4.6



ER Diagram

Figure 5: ER Diagram

2.5 Coding: Appendix A

2.6 Example 1: Authentaciton

```
import { getAuth, createUserWithEmailAndPassword } from
"firebase/auth";


const auth = getAuth();

createUserWithEmailAndPassword(auth, email, password)

.then((userCredential) => {  const

user  =  userCredential.user;

console.log("User registered:", user.uid);

})  .catch((error)

=> {

console.error("Registration Error:", error.message);

});
```

Example 2: Adding product

```
const express =
require("express"); const router = express.Router();
const Product = require("../models/Product");


router.post("/add", async (req, res) => {
 try {
  const product = new Product(req.body);  await product.save();
res.status(201).json({ message: "Product added successfully" });
 } catch (err) {  res.status(500).json({ error: err.message
});  }

});

module.exports = router;
```

2.6 Summary

This chapter introduces the development of the Multi-Vendor E-Commerce Platform. The report contains the system's architecture, database designing and user interface schematics is presented in such a way that it explains as to how these different components interact with each other to achieve the requirements of this project. The chapter also covers how we can implement a few important functionalities such as user registration, login, product management screen cart and checkout features (and order status track) with technologies like React, Node. js, MongoDB, and Firebase. Through the application of thoughtful design and hands-on code examples, this chapter illustrates how this platform provides secure, customerfriendly, efficient e-commerce for customers, vendors and administrators.

Chapter 3 Software Testing

3.1 Introduction

Software Testing This section is about the testing of the Multi-Vendor Ecommerce Platform to make sure that all of its features work properly, effectively and safely. Testing is the most important step to catch any errors, bugs, and usability problems before deployment. The chapter ranges from the functional testing of core functionalities such as user registration, login, product navigation category management cart handling checkout and order history to non-functional testing for performance reliability security and usability. Through carefully crafted testing, the project wants to make sure that the platform meets customers', vendors' and admins' expectations on a smooth, reliable and secure shopping.

3.2 Testing Features

The Multi-Vendor E-commerce Platform is tested for all its functionality and if that the features are perfect. Key testing features include:

User Registration & Login:

Make sure that registration/login works for both customers, vendors and admins.

Validate email, password and required fields on leave.

Profile Management:

Test create and update customer and vendor profile information.

Product Management:

Make sure that system vendors are able to attach products properly.

Check if the admin is able to handle(create, delete or edit) products.

Cart & Wishlist:

Testing add/remove to cart and wishlist functions.

Verify that cart behaviors need to be signed in.

Checkout & Payment:

Shipping options Integration with gerald/encypted-words "[Test]""Order placement, checkout process, and payment (Stubbed out for Stripe integration)
Order Tracking:

Ensure Customers can track their order status (Processing, Shipping, Delivery).

Admin Functionalities:

Try to manage users, orders and products.

Real-time data on Admin dashboard to be shown accurately.

Conclusion:

3.2.1 Feature to Be Tested

Feature	Description	Expected Outcome

User Registration	Customers and vendors register using Firebase authentication	Users are successfully registered with valid credentials
User Login	Login for Customers, Vendors, and Admin	Users can log in and access their respective dashboards
Profile Management	Add/Update profile information	User profiles are saved and updated correctly
Product Management	Vendors add products; Admin manages products	Products are added by vendors; Admin can delete/manage products
Product Browsing	Customers browse and view product details	Products are displayed correctly with accurate details
Cart Management	Add/remove products from cart	Cart updates correctly; requires login
Wishlist	Add products to wishlist	Wishlist reflects added items accurately
Checkout & Payment	Place orders and make payments (Stripe planned)	Orders are successfully placed and payment processed
Order Tracking	Track order status	Order status updates correctly (Processing, Shipping, Delivery)
Admin Functionalities	Manage users, orders, products	Admin dashboard reflects changes accurately; users and products managed correctly

3.3 Testing Strategies

1. The stratergies that are used to make the Multi-Vendor E-commerce Platform stronger, reliable and working are.testing following.XtraEditors, XtraGrid, WCF, Facebook API Assist in identifying testable aspects of the code.
2. Unit Testing:
3. Isolates all the components, such as user registration, login, adding a product and managing a cart.
4. Assures for the functionality of each module independently.
5. Integration Testing:
6. Provides error checking of module interactions with frontend/ backend, database activity, payment and so on.
7. Ensures that Dorforge as a whole works the way it should.
8. Functional Testing:
9. Validates functional requirements like product navigation, check out, order tracking and admin features.
10. VerificationThe system does what it is supposed to do for all users and admins.
11. Non-Functional Testing:
12. Exercises performance, reliability, security, usability and portability of the platform.

13. Measure: System response, availability, & user experience under different circumstances.
14. User Acceptance Testing (UAT):
15. Played out with sample users (Buyer/Seller) on the assumption to check if system fulfills real-life needs.
16. Indicates the platform is ready for use. ○

3.3.1 Test Approach

The test strategy for the Multi-Vendor E-commerce Platform comprises of manual and automated testing approaches to get a full system verification. Key points include:

Manual Testing:

o Doing the functional testing for core functionalities (e.g. registration, login, product browsing, cart handling, checkout and order-tracking).

o User interface validation for consistency and to ensure the responsiveness crossdevices.

Automated Testing (Planned/Future):

o These can utilize automated scripts to test API and backend database operations to ensure the data integrity and validity.

o Automated regressions tests facilitate the rapid detection of any regression following a new release. Black Box Testing: o Given to the in/out verification, with unknowing of code internal structure.

o Verifies that all requirements are implemented.

White Box Testing (Planned/Future):

o Can be used for confidential backend logic, like payment processing or order status updating, so that we could check out code paths and internal flows.

Regression Testing:

o Makes sure that new updates or features do not destroy the old capabilities. **3.3.2**

Pass/Fail Criteria

Pass Criteria:

The subsystem is working as designed - no problems or errors.

· All User's Inputs are sanitised and validated.

· Responses are also reasonable and not very slow (i.e., pages do not take 2-3 seconds to load).

· Security features, such as log in and transaction processing operate as intended.

· UI components are user-friendly across devices, browsers.

· Fail Criteria:

· The technology is not capable of ensuring appropriate operation or output.

· Core processes, such as registration, login, checkout or my orders result in errors.

· System is running slower than acceptable to user (slow response, time out).

· Errors happen during transactions and data is not valid anymore.

· UI components are either not working, non-interactive, or nonresponsive.

3.4 System Testing (Test Cases with Report)

Test Case 01: Register

Field	Details
Test Case: 5.3.1	**Test Case Name:** Register
System: Multi-Vendor E-commerce System	**Subsystem:** User Authentication
Designed by: Jahid	**Executed by:** Jahid
Description: The user registers by providing valid information.	
Pre-condition: User is on the registration page.	

Steps

STEP	NAME	EMAIL	PASSWORD	RETYPE PASSWORD	RESPONSE	PASS/FAIL	COMMENT
1	Jahid Hasan	jahid@gmail.com	123456	123456	Registration successful	Pass	Valid information provided.
2	*empty*	user@gmail.com	123456	123456	Name field empty	Fail	Name required.
3	Rahim	*empty*	123456	123456	Email field empty	Fail	Email required.
4	Karim	karim@gmail.com	*empty*	123456	Password empty	Fail	Password required.
5	Siam	siam@gmail.com	123456	124473	Password mismatch	Fail	Passwords must match.

Post-condition: User gets registered successfully if all validations are correct.

Test Case 02: Login:

Field	**Details**
Test Case: 5.3.2	**Test Case Name:** Login
Subsystem: User Authentication	

Steps

Step	Email	Password	Response	Pass/Fail	Comment
1	jahid@gmail.com	123456	Login successful	Pass	Valid credentials.
2	*empty*	123456	Email required	Fail	Missing email.

3	jahid@gmail.com	*empty*	Password required	Fail	Missing password.
4	wrong@gmail.com	123456	Account not found	Fail	Invalid email.
5	jahid@gmail.com	wrongpass	Wrong password	Fail	Incorrect password.

3. Test Case 03: Search Products:

Field	**Details**
Test Case: 5.3.3	**Test Case Name:** Search Product
Subsystem: Product Module	
Steps	

Step	**Search Input**	**Response**	**Pass/Fail**	**Comment**
1	"Laptop"	Relevant products shown	Pass	Valid keyword.
2	""	No input error	Fail	Search term required.
3	"@#$%"	No product found	Pass	Handles invalid search safely.

Test Case 04: View Product Details:

Step	**Product ID**	**Response**	**Pass/Fail**	**Comment**
1	101	Product details displayed	Pass	Valid product.
2	9999	Product not found	Fail	Invalid ID handled.

Test Case 05: Add to Cart:

Step	Product	Quantity	Response	Pass/Fail	Comment
1	Laptop	1	Added to cart	Pass	Works correctly.
2	Laptop	0	Quantity error	Fail	Minimum qty = 1.
3	*Invalid Product*	1	Product not found	Fail	Invalid product.

Test Case 06: Checkout:

Step	Cart Items	Address	Response	Pass/Fail	Comment
1	Valid items	Valid address	Proceed to payment	Pass	Normal flow.
2	Empty cart	Valid address	Cannot checkout	Fail	Cart must not be empty.
3	Valid items	*Empty address*	Address required	Fail	Address mandatory.

Test Case 07: Online Payment:

Step	Payment Info	Response	Pass/Fail	Comment
1	Valid card/bKash	Payment successful	Pass	Done.
2	Invalid card	Payment failed	Fail	Invalid card handled.
3	Timeout	Payment failed	Fail	Connection issue handled.

Test Case 08: Track Order:

Step	Order ID	Response	Pass/Fail	Comment
1	Valid order	Status displayed	Pass	Working.
2	Invalid ID	Not found	Fail	Correct error shown.

Test Case 10 (Admin): Add Vendor:

Step	Vendor Info	Response	Pass/Fail	Comment
1	Valid info	Vendor added	Pass	OK
2	Missing fields	Error message	Fail	Validation works

Test Case 11 (Admin): Manage Users:

Step	Action	Response	Pass/Fail	Comment
1	Delete user	User deleted	Pass	Working
2	Wrong ID	Error	Fail	Invalid ID handled

Test Case 12 (Admin): Manage Orders:

Step	Order ID	Action	Response	Pass/Fail	Comment
1	Valid ID	Update status	Status updated	Pass	OK
2	Wrong ID	Update	Order not found	Fail	Error shown

3.5 Summary

Formal testing of software is completed after completion to detect defects such as coding mistakes, errors or omission. This chapter tests the software functions in the Multi-Vendor Ecommerce Platform. It included tests for the functionality of the core modules including registration, login, product browsing, cart checkout and order tracking as well as admin operations testing in addition to non-functional aspect such as performance, reliability security usability and portability. The chapter also touched on testing strategies, test techniques and the pass/fail criteria all of which emphasized the way whose systematic method guides in the identification and repair of any defects. In general, the development & testing as addressed guarantees a stable and secure usage of platform for customerssellers and admins.

Chapter 4 Deployment and Maintenance

4.1 Introduction

This chapter deals with the implementation and management of MultiVendor E-commerce Platform. Here we discuss how to expose the system to our end users and deploy the frontend, backend, and database. Maintenance of monitoring, updates, backups and bug fixing to make the platform reliable, secure and updated is also described in the chapter. There is need for the proper installation and maintenance of a system as much as this is necessary in its setup phase.

4.2 Try and emulate the SRLC (software release life cycle)

In this chapter we focus on implementing and maintaining the MVEP using SRLC. The SRLC model safely delivers, monitors and maintains with systematic approach the software life cycle. Once it is ready, deployment takes place when the system goes from a test environment to its on-site production environment for use by customers, suppliers or employees. Maintenance consists of checking performance, repairing glitches and securing features, as well as making backup copies to

Chapter 5 User Manual

5.1 Introduction

Chapter 1 is devoted to user guide for Multi-Vendor E-Commerce Platform explaining our system usage for customers, vendors and admin. That provides step-by-step solutions to some of the key functionality such as registration, login, product browsing, adding to cart, checkout and more. The goal of the manual is to help all users access, perform functions, and take advantage of what's available on the platform seamlessly. Both clear text directions and screen shots (if any) help users to learn the system to their best advantage. 5. 2 Project Functionalities

Admin Login

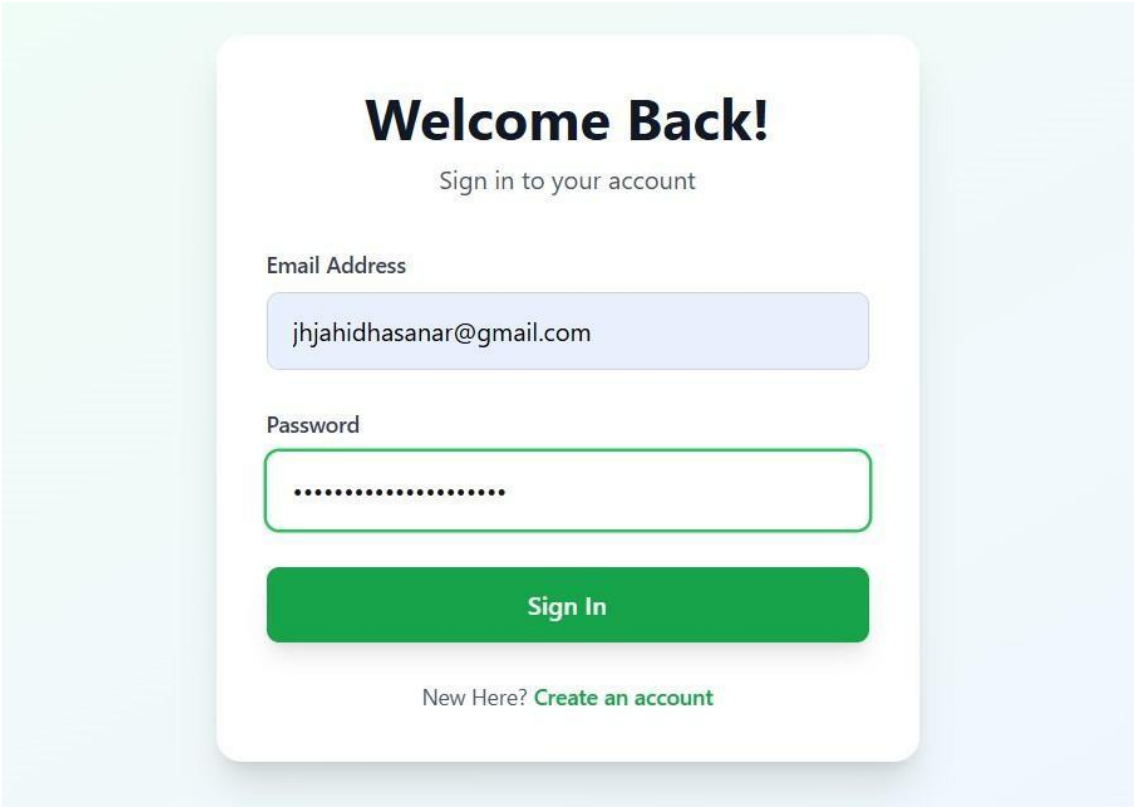
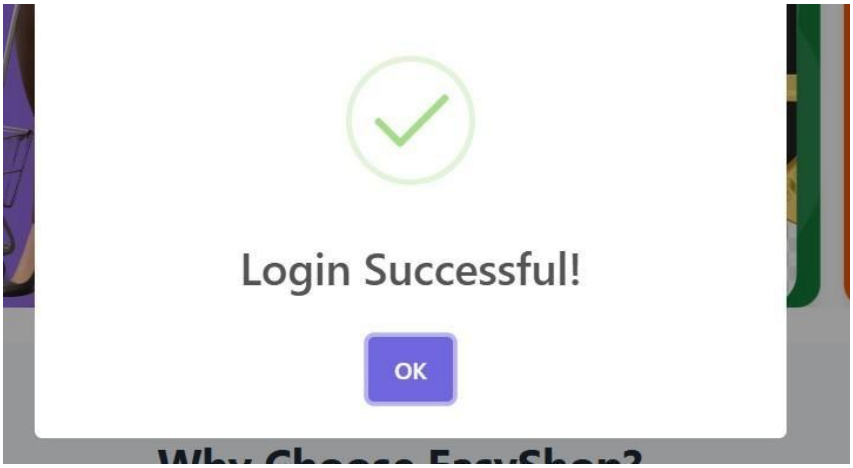
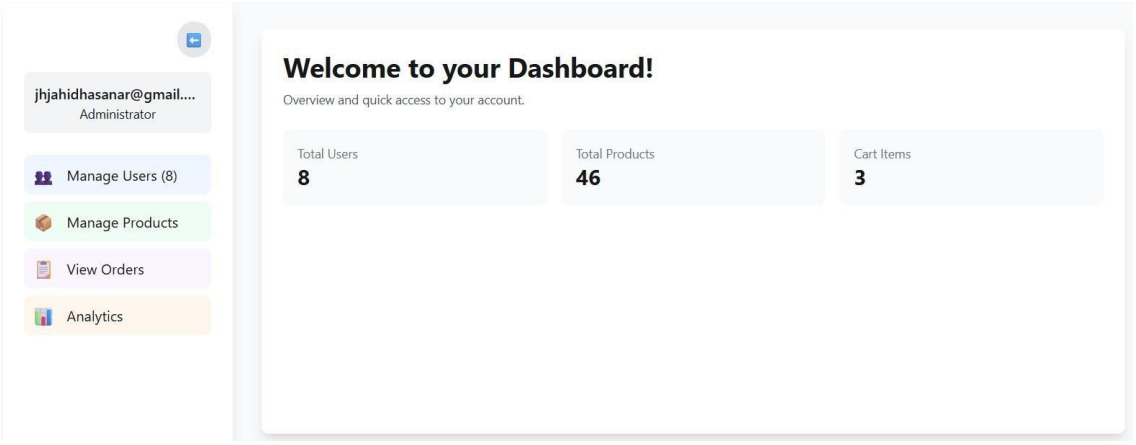


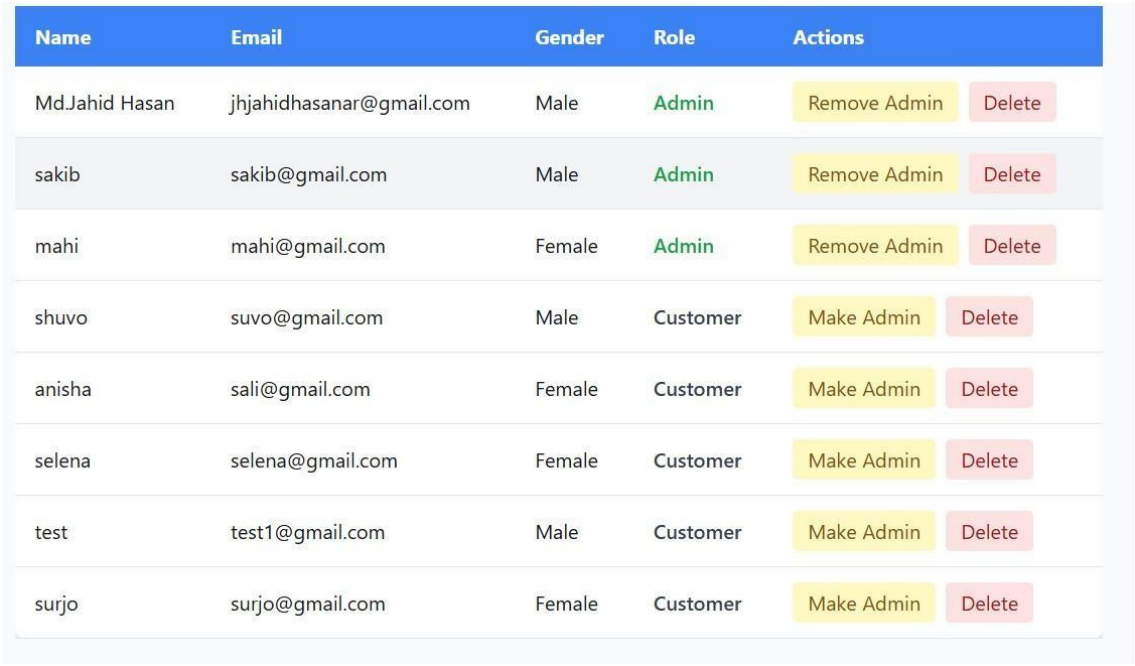
Figure: Admin Login



Admin Dashboard:

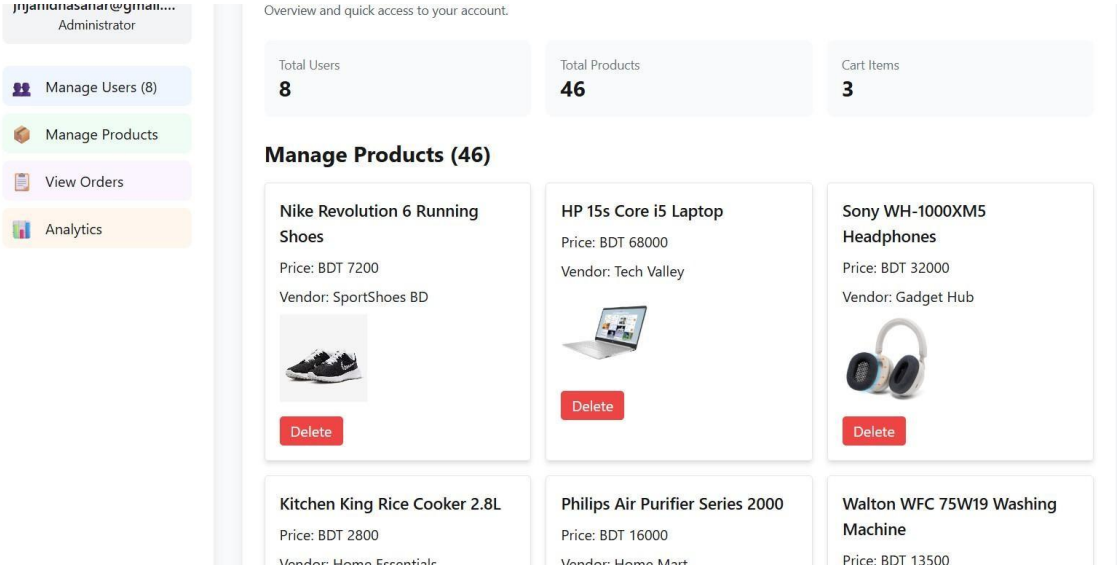


Manage Users:

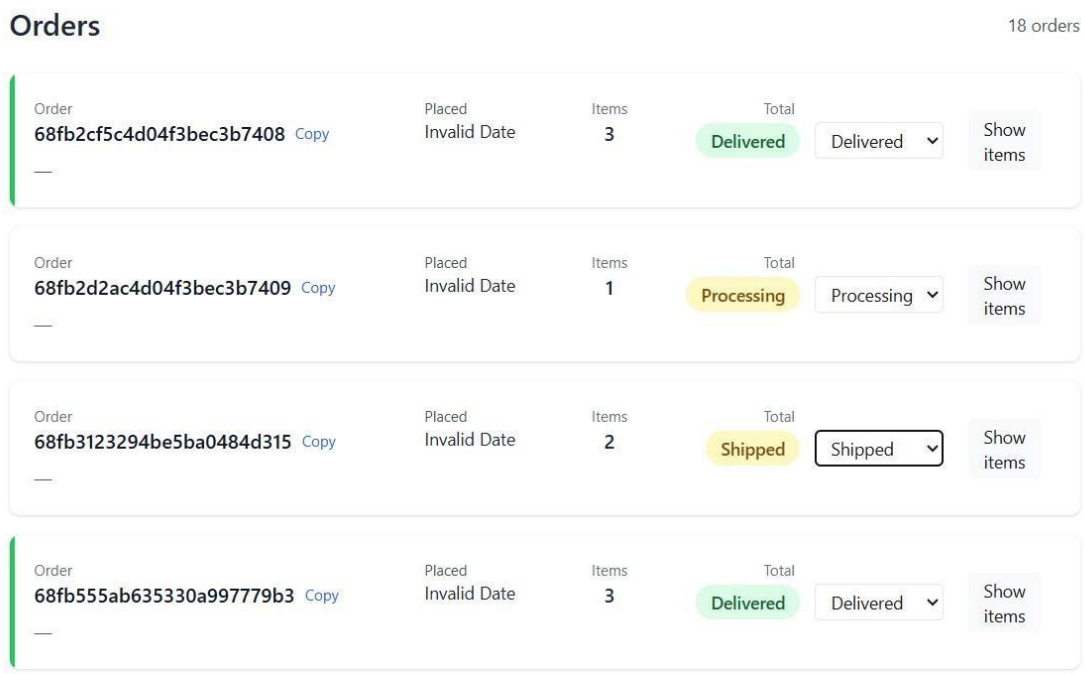


Name	Email	Gender	Role	Actions
Md.Jahid Hasan	jhjahidhasanar@gmail.com	Male	Admin	Remove Admin Delete
sakib	sakib@gmail.com	Male	Admin	Remove Admin Delete
mahi	mahi@gmail.com	Female	Admin	Remove Admin Delete
shuvo	suvo@gmail.com	Male	Customer	Make Admin Delete
anisha	sali@gmail.com	Female	Customer	Make Admin Delete
selena	selena@gmail.com	Female	Customer	Make Admin Delete
test	test1@gmail.com	Male	Customer	Make Admin Delete
surjo	surjo@gmail.com	Female	Customer	Make Admin Delete

Manage Product:

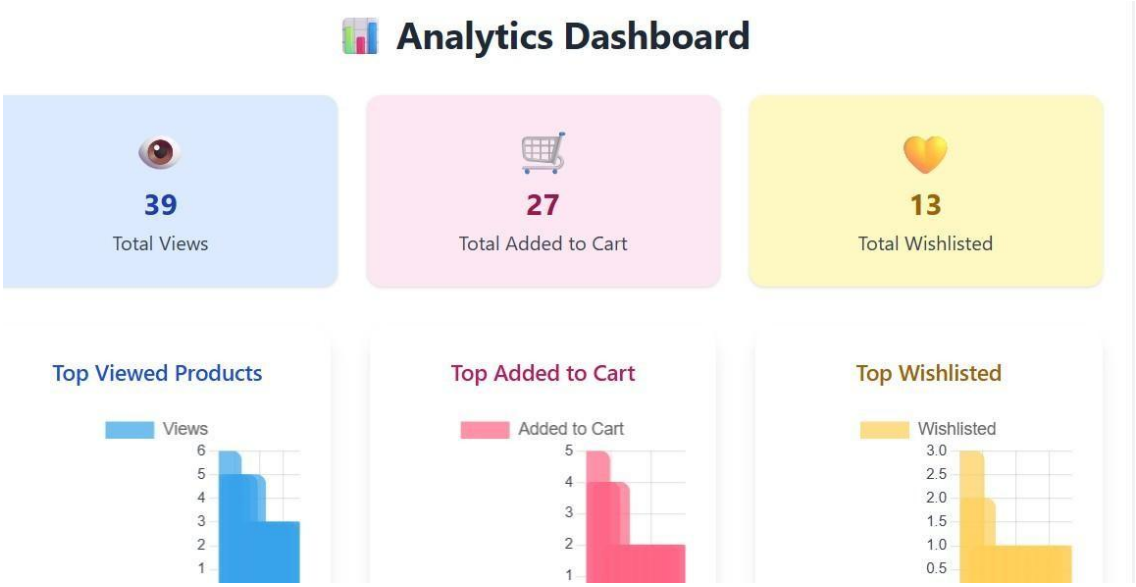


Orders Tracking :

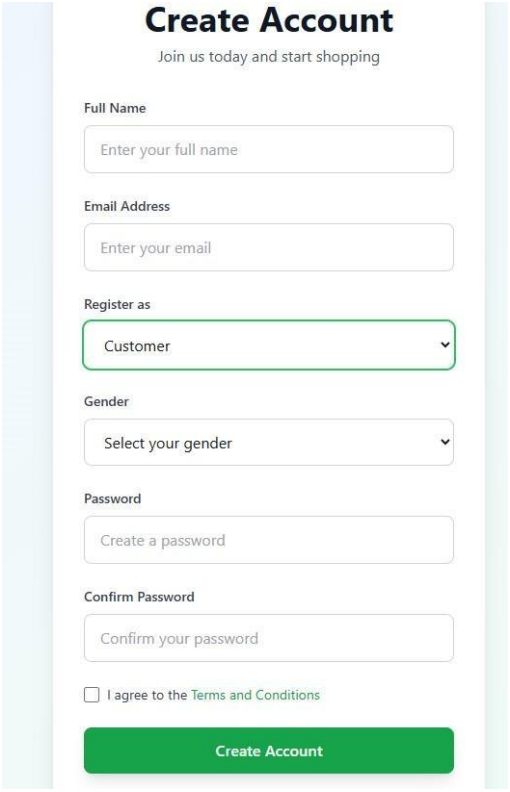


Manage status:

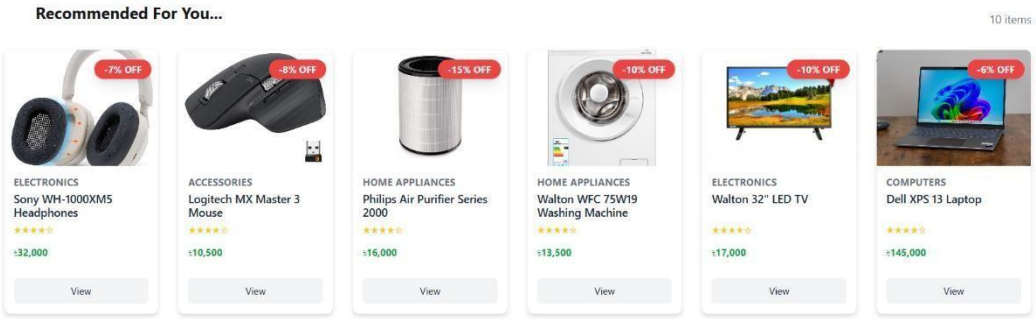
Analytics for admin:



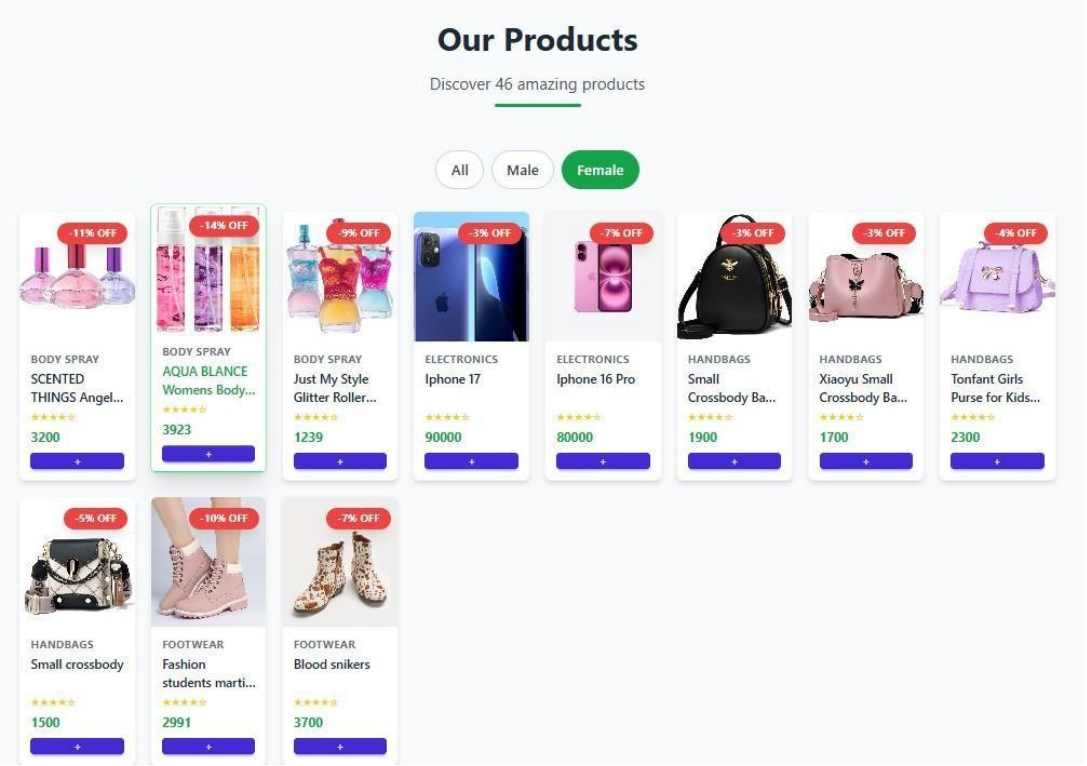
Customer Register:



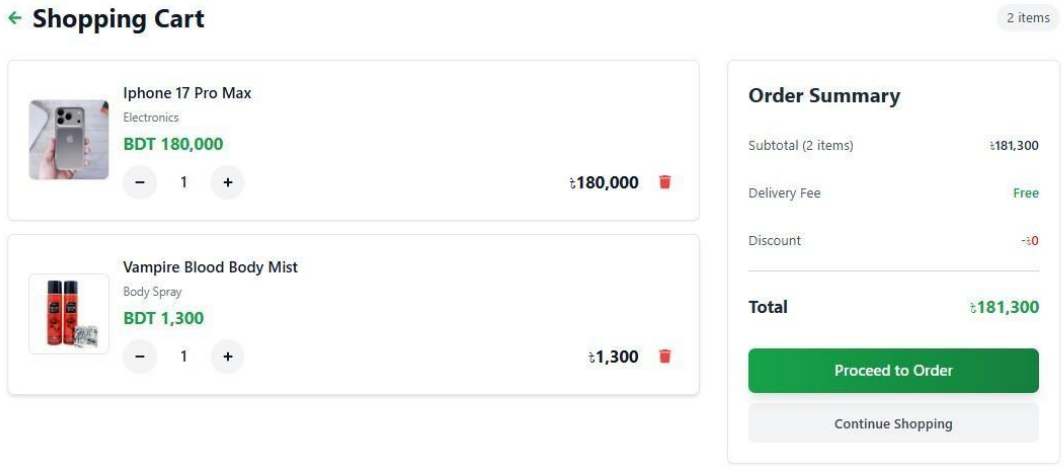
AI Smart Recommendation:



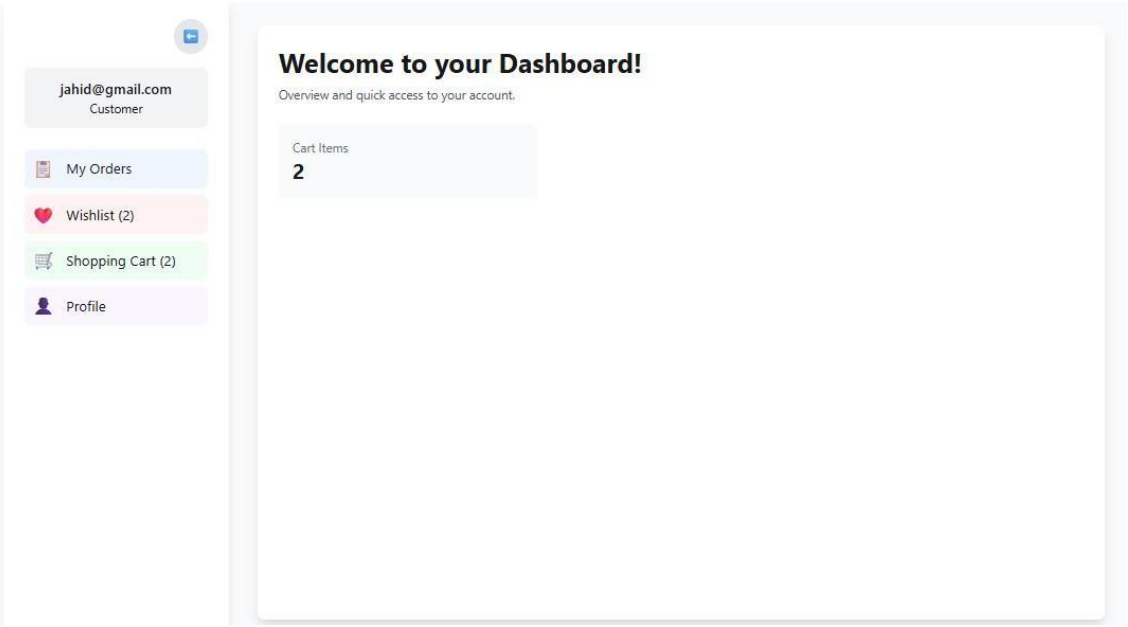
Gender sorting product show:



Shopping Cart:



Customer Dashboard:



Customer can track order:

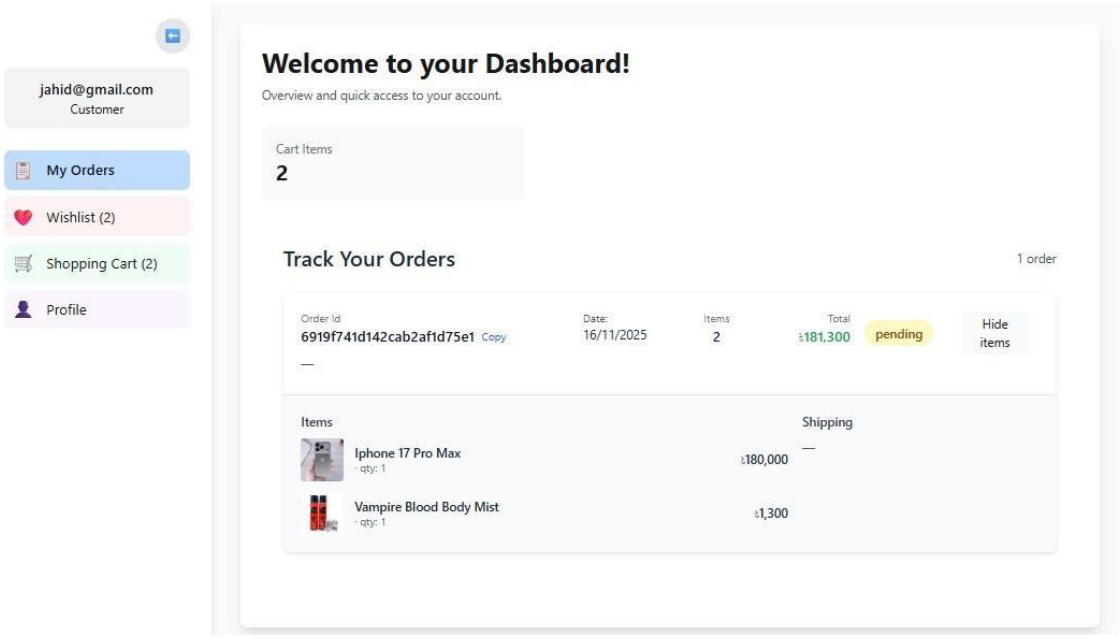


Figure : Add Restaurant

5.3 Summary :

The Multi-Vendor E-commerce Platform The complete guide This chapter is designed to provide the best hands on tutorial of using a program called MULTI-VENDOR. Illustrating the basic functionality for all users, customers, vendors and admin like registration, signing in product listing cart order tracking and checkout using spring rest services. The user guide makes the platform easy to use and users can easily organize what they want, do their job with it efficiently and take full advantage of the system.

Chapter 6 Project Summary

6.1 Introduction

The summary of the Multi-Vendor E-commerce Platform project is given-objective, features, limitations, and scope are explained. It summarizes the work performed, difficulties encountered and, in general terms, the results reached by the project, presenting a comprehensible outline of its importance and accomplishments.

6.2 Project Limitation

Limitations There are several shortcomings in this study:

· Time Constraint: Not enough space to have nice-to-haves things like live chat support, full vendor dashboards, and automatic testing.

· Budget constraints: Limited budget influenced the inclusion of premium tools and services.

· Technology Limitations: Stripe payment integration and JWT-based Authentication were intended but partially implemented.

· Unmet Features: Product reviews, vendor updates and advanced analytics are not part of this edition.

6.3 Scope

The project covers:

· Customer IMP Features: User registration, login, product browse and search, cart management & wishlist management Checkout page and track order.

· Vendor Functionality: Add product.

· Admin Features : Manage users, products and orders.

· Excluded Features: Live chat support, vendor dashboards, product reviews and advanced payment options and AI-based analytics.

6.4 Future Work

Potential future improvements include:

· Stripe payment integration with JWT authentication for more secure payments added.

· Putting in place live chat support for immediate customer help.

· Building rich vendor dashboards for product management, updates and analytics.

· Including customer reviews and ratings, sophisticated search, and AI based recommendations.

· Improving automated testing and CI/CD for improved reliability and maintainability.

6.5 Conclusion

The Multi-Vendor E-commerce Platform adequately develops the safe, functioned and friendly system for the customers, vendors and administrators. Basic goals like product or order management and user administration were covered. This project is an excellent example project for web development, database management, and system deployment and can only serve as a base for anyone who wants to build their new features on top of it or wish to use it as a solid and stable platform for ecommerce websites.

REFERENCES

1. Flanagan, D. *JavaScript: The Definitive Guide*, 7th Edition, O'Reilly Media, 2020.
2. MongoDB Documentation. https://www.mongodb.com/docs/
3. Firebase Documentation. https://firebase.google.com/docs
4. Stripe Documentation. https://stripe.com/docs
5. Mozilla Developer Network (MDN) Web Docs. https://developer.mozilla.org 6. TutorialsPoint. *Node.js, Express.js, React, and Tailwind CSS Tutorials*. https://www.tutorialspoint.com
7. Project-specific articles, online blogs, and tutorials referenced during development.

221-35-959

ORIGINALITY REPORT

16%	10%	2%	14%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	5%
2	Submitted to Midlands State University Student Paper	3%
3	Submitted to NCC Education Student Paper	1%
4	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
5	Submitted to Ain Shams University Student Paper	1%
6	umpir.ump.edu.my Internet Source	<1%
7	Submitted to Majmaah University Student Paper	<1%
8	Submitted to Universidad Nacional Abierta y a Distancia, UNAD,UNAD Student Paper	<1%
9	Submitted to Universiti Malaysia Pahang Student Paper	<1%
10	Submitted to Military Technological College Student Paper	<1%
11	dev.to Internet Source	<1%
12	Submitted to Higher Education Commission Pakistan Student Paper	<1%
13	Submitted to Elizabeth School of London Student Paper	<1%
14	Submitted to Macquarie University Student Paper	<1%
15	www.intrallect.com Internet Source	<1%
16	deeprojectmanager.com Internet Source	<1%
17	Submitted to Academy of Information	<1%