



A Project Report on

ONDC PLATFORM

Submitted the Project under the guidance of **Ms. Nancy Sukhadia** and
Ms. Jiby Babin for the Final 2nd Year Major Project for
Bachelor of Technology in
Computer Science and Engineering

By
Aditya Kumar
Roll no: UI22CS03
and
Anurag Kumar
Roll no: UI22CS09

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SURAT-
394190
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
May, 2024

Indian Institute of Information Technology Surat
Computer Science and Engineering Department



CERTIFICATE

This is to certify that candidate **Aditya Kumar** bearing
Roll No: UI22CS03 and **Anurag Kumar Roll No: UI22CS09**
of B.Tech. IV, 4th Semester has successfully carried out the work on
“ONDC PLATFORM” in the Subject of **Software Engineering**
(CS-401) on May, 2024.

1. Subject Co-ordinators 1st:
Name: **Ms. Jiby Babin**

Sign:.....

2. Subject Co-ordinators 2nd:
Name: **Ms. Rishi Sharma**

Sign:.....

(Seal of the Institute)

DECLARATION

This is to certify that

- (i) This report comprises my original work towards the degree of Bachelor of Technology in Computer Science and Engineering at Indian Institute of Information Technology (IIIT) Surat and has not been submitted elsewhere for a degree,
- (ii) Due acknowledgement has been made in the text to all other material used.

.....

1. Signature of Student
Aditya kumar

.....

2. Signature of Student
Aunrag kumar

ACKNOWLEDGEMENTS

I, **Aditya Kumar (Roll No. UI22CS03)**, a student of IIIT Surat, would like to express my gratitude to all those who have contributed to the successful completion of this project on the ONDC platform.

First and foremost, I extend my heartfelt thanks to my teammate, **Anurag Kumar (Roll No. UI22CS09)**, for his valuable contributions to the frontend development of this project using React. His dedication and efforts have greatly enhanced the overall quality of our application.

Furthermore, I am deeply grateful to the faculty members, Jiby Ma'am and Rishi Sharma sir, for their unwavering support, guidance, and mentorship throughout this journey. Their expertise, encouragement, and constructive feedback have played a pivotal role in shaping this project and my personal growth as a developer. I am truly thankful for their patience and cooperation.

Atlast, I would also like to acknowledge and express my sincere appreciation for my role in developing the entire backend and frontend each and every components of the project. From conceptualization to implementation, including testing and the creation of highly detailed admin dashboards, I have been involved in every stage of the project's lifecycle.

In conclusion, I would like to thank everyone who has contributed to this project, directly or indirectly. Your support and encouragement have been instrumental in bringing this project to fruition.

Aditya Kumar (UI22CS03)

ABSTRACT

The Open Network for Digital Commerce (ONDC) platform has emerged as a revolutionary initiative aimed at transforming India's digital commerce landscape. In line with this vision, our project focuses on the development of a comprehensive platform leveraging the ONDC infrastructure to facilitate seamless digital commerce experiences for both businesses and consumers.

The project encompasses the design and implementation of a robust and user-friendly web application built using modern technologies such as React for the frontend and a custom backend architecture. The platform offers a range of features including product listing, ordering, payment processing, and administration functionalities tailored to meet the diverse needs of users.

Key highlights of the project include:

- **Frontend Development:** Utilizing React, our team has crafted an intuitive and responsive user interface, ensuring optimal user experience across devices.
- **Backend Architecture:** A custom backend architecture has been meticulously designed and implemented to handle data management, authentication, and integration with external APIs seamlessly.
- **Administration Panel:** A dedicated administration panel empowers administrators to manage products, orders, users, and other essential aspects of the platform efficiently.
- **Integration with ONDC:** Leveraging the capabilities of the ONDC platform, our application seamlessly integrates with the ONDC ecosystem, enabling businesses to leverage the benefits of a unified digital commerce infrastructure.
- Through this project, we aim to contribute to the advancement of digital commerce in India by harnessing the potential of the ONDC platform. By providing a feature-rich and user-centric solution, we strive to empower businesses to thrive in the digital age while offering consumers a seamless and secure online shopping experience.

Contents

Chapter No.	Chapter Name	Page
Preface	Title	I
Preface	Certification	II
Preface	Declaration	III
Preface	Acknowledgement	IV
Preface	Abstract	V
Preface	Contents	VI
Preface	List of Principal Symbols and Acronyms	VII
Preface	List of Figures	VIII-IX
CHAPTER I	Introduction	1-4
CHAPTER II	Tools & Technologies	5-7
CHAPTER III	Proposed Systems	8-9
CHAPTER IV	Design	10-19
CHAPTER V	Implementation	20-29
CHAPTER VI	Testing and Experimental Results	30-34
CHAPTER VII	Conclusion and Future Scope	35-38
	References	39

List of Principal Symbols and Acronyms

Term	Definition
ONDC	Open Network for Digital Commerce
SRS	Software Requirements Specification
API	Application Programming Interface
WCAG	Web Content Accessibility Guidelines
GDPR	General Data Protection Regulation
HTTPS	Hypertext Transfer Protocol Secure
UI	User Interface
UX	User Experience
SSL	Secure Sockets Layer
JSON	JavaScript Object Notation
SQL	Structured Query Language
CDN	Content Delivery Network
URL	Uniform Resource Locator
OS	Operating System
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
DBMS	Database Management System
B2B	Business-to-Business
B2C	Business-to-Consumer
MVP	Minimum Viable Product
OOP	Object-Oriented Programming
CI/CD	Continuous Integration/Continuous Deployment

List of Figures

Figure. No.	Figure Name	Page No.
1	ONDC Logo	1
2	Level 0 DFD	10
3	Data Flow Diagram of my ONDC Platform	11
4	ONDC CLASS DIAGRAM	12
5	Use Case Diagram of ONDC Platform	13
6	State Diagram of ONDC Platform	14
7	Activity Diagram of ONDC Platform)	15
8	Sequence Diagram	16
9	Deployment Diagram for ONDC	17
10	Gantt Chart	18
11	Gantt Chart	18
12	Gantt Chart	19
13	Home Page	23
14	Product Details Page	14
15	Add to Cart Button	24
16	Cart Page with increased Quantity	24
17	Shipping Address Page after Add to Cart	25
18	Payment Page	26
19	Payment Details	26
20	Payment Success and Order Placed	27

21	Filter Products based on Price & Category	27
22	Admin Dashboard	28
23	Product Dashboard	28
24	Charts Dashboard	29
25	Performance Testing Analysis of Webpage load test	31
26	Memory Test	32
27	Fetching all Products details from backend	33
28	Posting (Adding) the Product to backend	33
29	Fetch all Order data	34
30	MongoDB	34

CHAPTER - I

Introduction

Title of the project:- OPEN NETWORK FOR DIGITAL COMMERCE (ONDC) PLATFORM E-COMMERCE MARKETPLACE

Problem Statement :- Develop a comprehensive e-commerce platform based on the principles of the Open Network for Digital Commerce (ONDC) to revolutionize online retail in India. The platform aims to democratize e-commerce by providing inclusive access to sellers from diverse backgrounds, particularly those in small towns and rural areas, without the imposition of commission fees.

S. No.	Team Members Name	Roll no.
1.	Aditya Kumar	UI22CS03
2.	Anurag Kumar	UI22CS09

Table 1: Group Details

Why ONDC and This Problem Statement?

In a rapidly evolving digital landscape, the Open Network for Digital Commerce (ONDC) stands as a beacon of inclusive innovation, poised to reshape the e-commerce landscape in India. With millions of sellers across the country, the potential for digital retail expansion is immense, yet traditional barriers have hindered widespread participation. ONDC emerges as a catalyst for change, ushering in a new era of accessibility, transparency, and empowerment.



Figure 1: ONDC Logo

Software Requirements Specification (SRS)

1. Introduction :

The Open Network for Digital Commerce (ONDC) Marketplace is an e-commerce platform designed to facilitate online shopping for customers and streamline order management processes for administrators. This document outlines the functional and non-functional requirements of the ONDC Marketplace.

2. Functional Requirements

2.1 Product Management:

- Admins shall be able to add products to the platform, including details such as name, description, price, stock, description and quantity.
- Admins shall have the capability to update and delete existing products as necessary.

2.2 Shopping Cart and Checkout

- Customers shall be able to browse products and add them to their shopping cart.
- Customers shall specify quantities for selected products.
- The system shall calculate the order total based on selected products and quantities also can apply coupons.
- Customers shall review their order and make modifications before finalizing.
- The system shall validate and process customer orders.

2.3 Order Reception

- The customer service department shall receive customer orders placed through the web portal.
- The system shall display order details to the customer service team.
- Customer service representatives shall access order information and perform necessary actions.

2.4 Inventory Check and Order Confirmation

- The customer service team shall verify the availability of ordered products in the inventory system.
- If products are in stock, the system shall confirm the order and proceed to order fulfillment.
- If products are out of stock or unavailable, the system shall notify the customer of out of stock and customer cant order.

2.5 Order Fulfillment

- The warehouse team shall receive confirmed orders from the customer service department.
- The system shall facilitate the packaging and shipping process for ordered items.
- A shipping confirmation email with tracking number and estimated delivery date shall be generated for customers.
- The warehouse team shall update the order status in the system after shipping.

2.6 Order Status Notification

- The system shall send automated notifications to customers regarding the status of their orders.
- Customers shall receive notifications about order confirmation, fulfillment, and shipping details.

3. Non-Functional Requirements

3.1 Performance

- The web portal shall have low latency and provide a responsive user interface.
- The system shall handle multiple concurrent user sessions without performance degradation.
- Inventory checks and order processing time shall be optimized to minimize customer wait time.

3.2 Security

- Secure authentication mechanisms shall protect customer information.
- Customer data, including personal and payment details, shall be encrypted and stored securely.
- Access controls shall restrict unauthorized access to sensitive information.

3.3 Reliability

- The system shall be available for customer order placement 24/7 with minimal downtime for maintenance.
- Backup and recovery mechanisms shall ensure data integrity and availability.

3.4 Usability

- The web portal shall have an intuitive and user-friendly interface for easy navigation and order placement.
- Error handling and validation messages shall guide users and prevent incorrect inputs.

3.5 Scalability

- The system architecture shall support future scalability to accommodate increasing customer demands.
- The web portal shall handle a growing customer base and larger order volumes without performance degradation.

Chapter - II

Tools & Technologies

In this chapter, we will delve into the various technologies and tools utilized in the development of our project. Each component of our technology stack has been carefully chosen to meet the specific requirements of our project and to ensure efficiency, scalability, and maintainability.

2.1 Frontend Technologies

2.1.1 React.js

React.js is a JavaScript library used for building user interfaces. Its component-based architecture allows for the creation of reusable UI elements, enhancing development speed and code maintainability.

2.1.2 TypeScript

TypeScript is a statically typed superset of JavaScript that adds optional static typing and other advanced features to the language. It enhances code quality, provides better tooling support, and helps catch errors at compile time.

2.1.3 Redux

Redux is a predictable state container for JavaScript apps, often used with React.js for managing application state in a predictable manner. It enables the centralization of application state and facilitates state management across components.

2.1.4 Styled-components

Styled-components is a library for styling React components using tagged template literals. It allows for the creation of styled components with scoped styles, improving code organization and maintainability.

2.2 Backend Technologies

2.2.1 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It enables the execution of JavaScript code outside the browser, making it ideal for building scalable and high-performance server-side applications.

2.2.2 Express.js

Express.js is a minimalist web framework for Node.js that simplifies the process of building web applications and APIs. It provides robust routing, middleware support, and various utilities for handling HTTP requests and responses.

2.2.3 MongoDB

MongoDB is a NoSQL database that uses a document-oriented data model. It is highly scalable, flexible, and suitable for handling large volumes of unstructured data. MongoDB's schema-less design allows for quick and iterative development.

2.2.4 Mongoose

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a straightforward, schema-based solution for modeling application data and interacting with MongoDB databases.

2.3 Additional Tools and Libraries

2.3.1 Stripe

Stripe is a popular payment processing platform that provides APIs for accepting online payments and managing billing. It offers robust security, easy integration, and support for various payment methods.

2.3.2 Axios

Axios is a promise-based HTTP client for making HTTP requests from the browser and Node.js. It simplifies the process of sending asynchronous HTTP requests and handling responses, providing a clean and intuitive API.

2.3.3 Postman

Postman is a collaboration platform for API development that simplifies the process of testing, documenting, and sharing APIs. It provides tools for creating and sending HTTP requests, testing API endpoints, and generating API documentation.

2.3.4 Firebase Authentication (Google)

Firebase Authentication is a service provided by Google that enables easy and secure user authentication in web and mobile apps. It supports various authentication methods, including email/password, social login (Google, Facebook, etc.), and third-party identity providers.

2.3.5 Chart.js

Chart.js is a JavaScript charting library used for creating visually appealing and interactive charts on the client side. It offers a wide range of chart types and customization options, making it suitable for various data visualization needs.

Chapter - III

Proposed Systems

In this chapter, we will explore the proposed systems and functionalities of our project. The proposed systems aim to address specific requirements and objectives identified during the project planning phase. Each system is designed to enhance user experience, improve efficiency, and achieve the project's overall goals.

3.1 User Authentication System

The User Authentication System is a fundamental component of our project, designed to ensure secure access to the application and protect user data. Leveraging Firebase Authentication by Google, users can create accounts, log in securely, and access personalized content and features. The system supports authentication login via Google accounts.

3.2 Data Visualization System

The Data Visualization System utilizes Chart.js to create visually appealing and interactive charts that provide valuable insights into data trends and patterns. By leveraging Chart.js's flexible charting capabilities, users can explore data in various formats, such as line charts, bar charts, pie charts, and radar charts. The system empowers users to make informed decisions based on data analysis and visualization.

3.3 Payment Processing System

The Payment Processing System, integrated with Stripe, enables users to make secure and seamless payments for goods and services within the application. Leveraging Stripe's robust payment processing platform, the system supports various payment methods, including credit/debit cards, digital wallets, and bank transfers. Users can complete transactions quickly and efficiently, enhancing the overall user experience.

3.4 Content Management System

The Content Management System provides administrators with the tools and functionalities to manage and organize content within the application. Leveraging MongoDB as the underlying database and Mongoose as the Object Data Modeling (ODM) library, administrators can create, edit, and delete content dynamically. The system ensures data integrity, scalability, and flexibility, allowing for easy content updates and modifications.

3.5 Communication System

The Communication System facilitates seamless communication and collaboration among users within the application. Leveraging real-time messaging features and notifications, users can engage in discussions, share updates, and collaborate on projects in a collaborative environment. The system enhances user engagement, fosters community interaction, and promotes knowledge sharing.

3.6 System Administration and Monitoring

The System Administration and Monitoring System provides administrators with the tools and functionalities to manage, monitor, and maintain the application infrastructure. Leveraging monitoring tools and dashboards, administrators can track system performance, identify potential issues, and implement proactive measures to ensure system reliability and availability. The system enhances operational efficiency and minimizes downtime, ensuring a seamless user experience.

This chapter outlines the proposed systems and functionalities of our project, highlighting their significance in achieving the project's objectives and delivering value to users. Each system is designed to address specific requirements and contribute to the overall success of the project.

Chapter - IV

Design

This chapter delves into the design aspects of our project, encompassing user interface (UI) design, database design, system architecture, and security considerations. Design plays a pivotal role in ensuring the usability, scalability, and robustness of the application.

We delve into the design aspects of our project, leveraging the diagrams and information provided to craft a comprehensive overview of the design considerations for the ONDC Platform E-commerce Marketplace.

4.2 Data Flow Diagram (DFD)

The Data Flow Diagram depicts the flow of data within the ONDC Platform, illustrating how information moves between various components. Figure 6 presents the Level 0 DFD, showcasing the high-level overview of data flow:

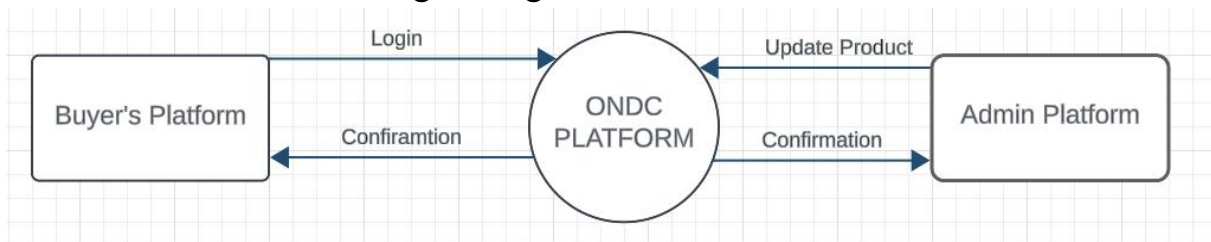


Figure 2: Level 0 DFD

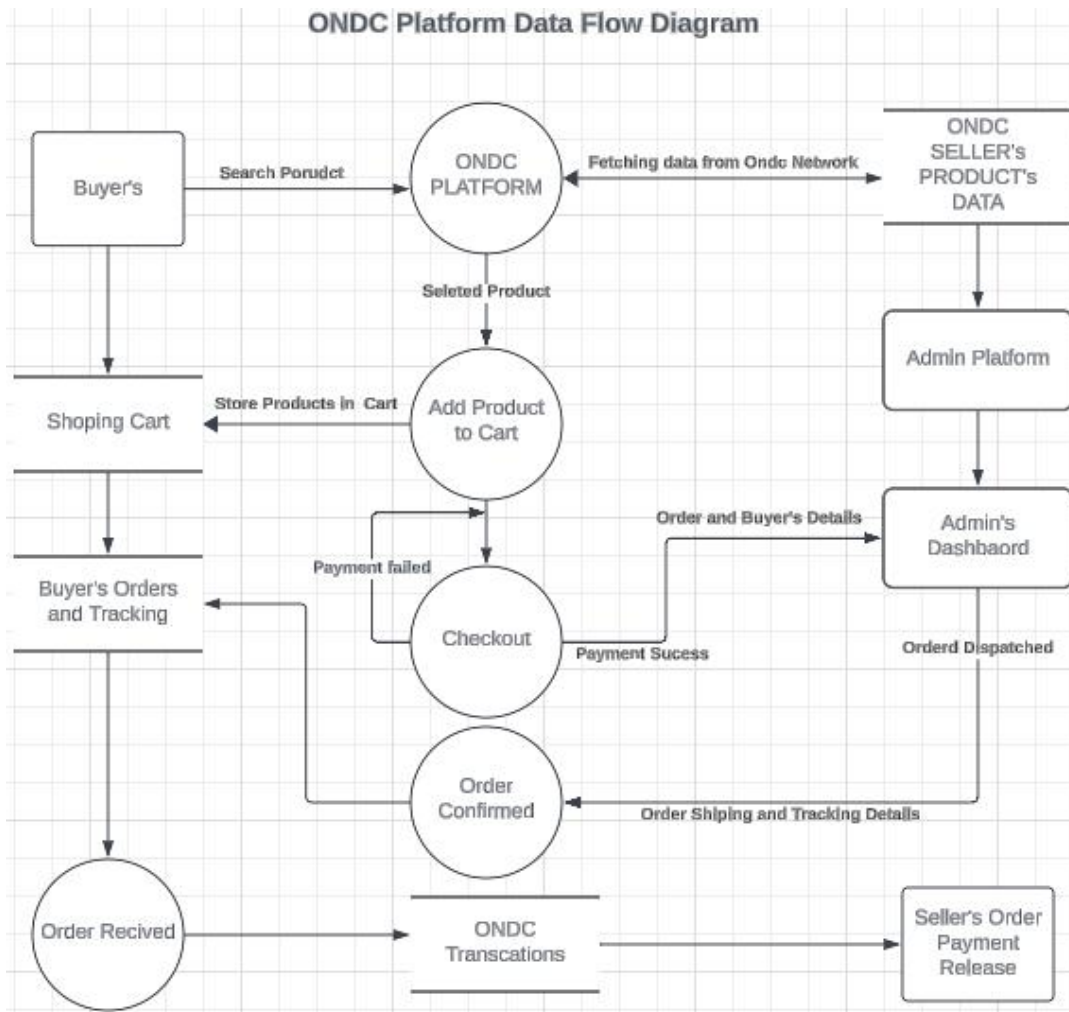


Figure 3 : Data Flow Diagram of my ONDC Platform

The ONDC platform facilitates transactions between buyers and sellers. When a buyer confirms an order, they can add the product to their shopping cart and proceed to checkout. After the buyer's order and payment are confirmed, the seller releases the payment and the buyer receives their order with tracking details. The seller can view their dashboard to see the order and shipping details. Overall, the ONDC platform streamlines the buying and selling process for both parties.

4.3 Use Case Diagram, State Diagram, and Activity Diagram

4.4 Class Diagram

The Class Diagram outlines the structure of classes and their relationships within the ONDC Platform. It provides a blueprint for the application's object-oriented design. Unfortunately, the provided diagrams did not include a Class Diagram.

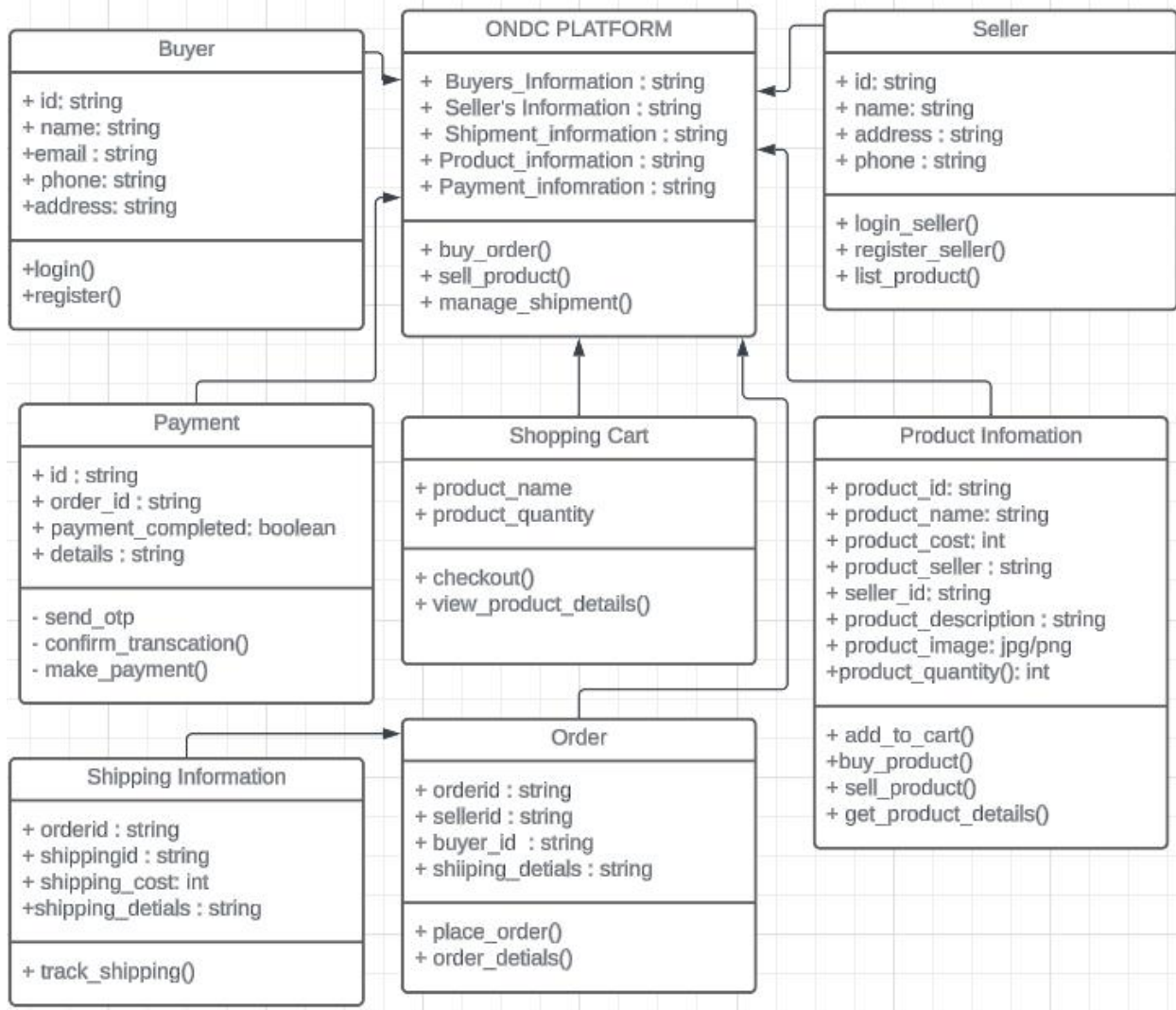


Figure 4: ONDC CLASS DIAGRAM

The ONDC platform offers a comprehensive solution for online shopping, with features such as:

- Shopping cart functionality for easy checkout and product details viewing
- Buyer and seller registration and login
- Product information with images, and quantity

4.3.1 Use Case Diagram

Use case diagrams depict interactions between users (actors) and the system, showcasing how users interact with the application to accomplish tasks. Figure 8 illustrates the Use Case Diagram for the ONDC Platform:

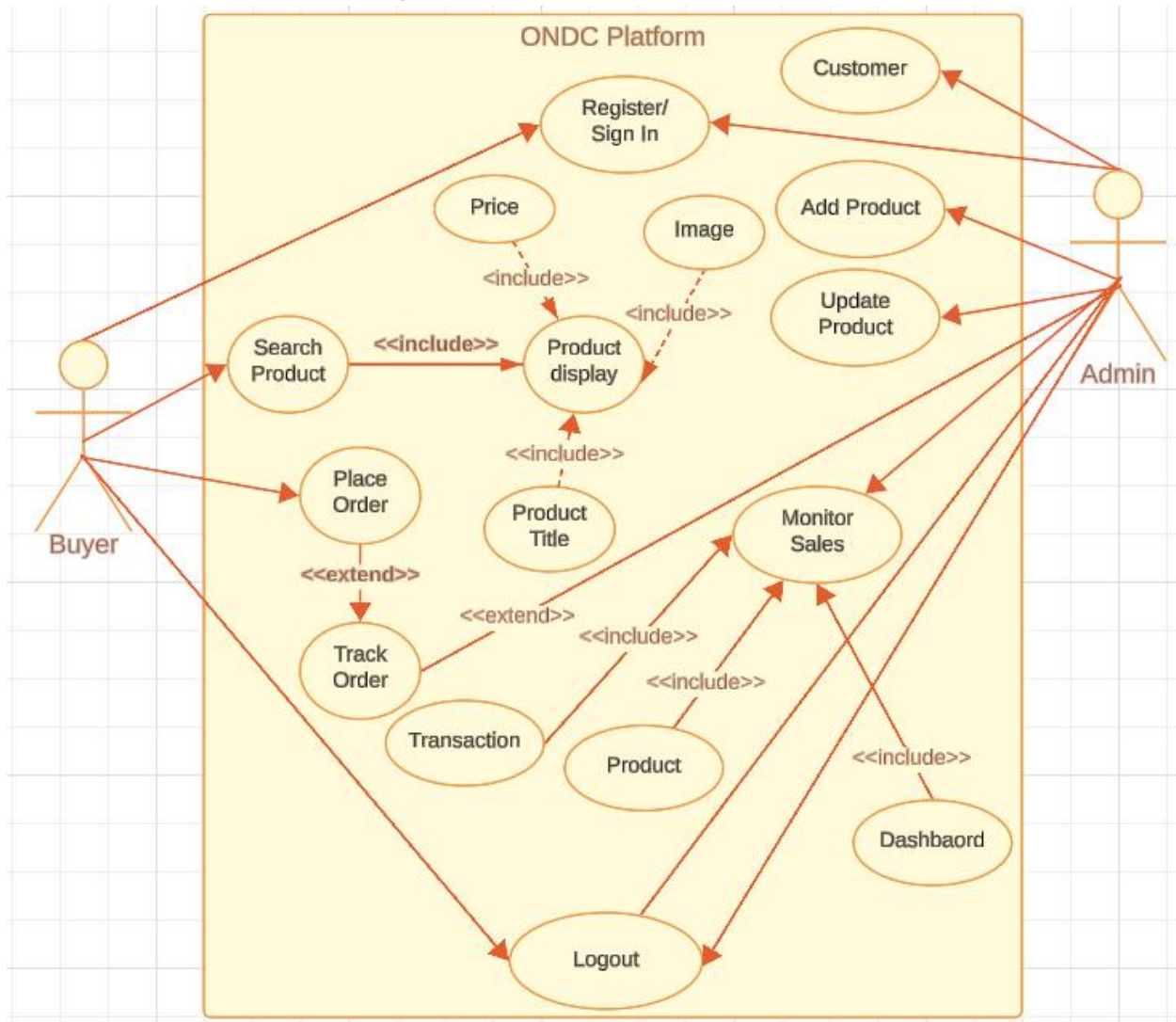


Figure 5: Use Case Diagram of ONDC Platform

4.3.2 State Diagram

State diagrams, also known as state machine diagrams, visualize the states and transitions of an object or system. Figure 9 presents the State Diagram for the ONDC Platform, highlighting the dynamic states and transitions during user interactions:

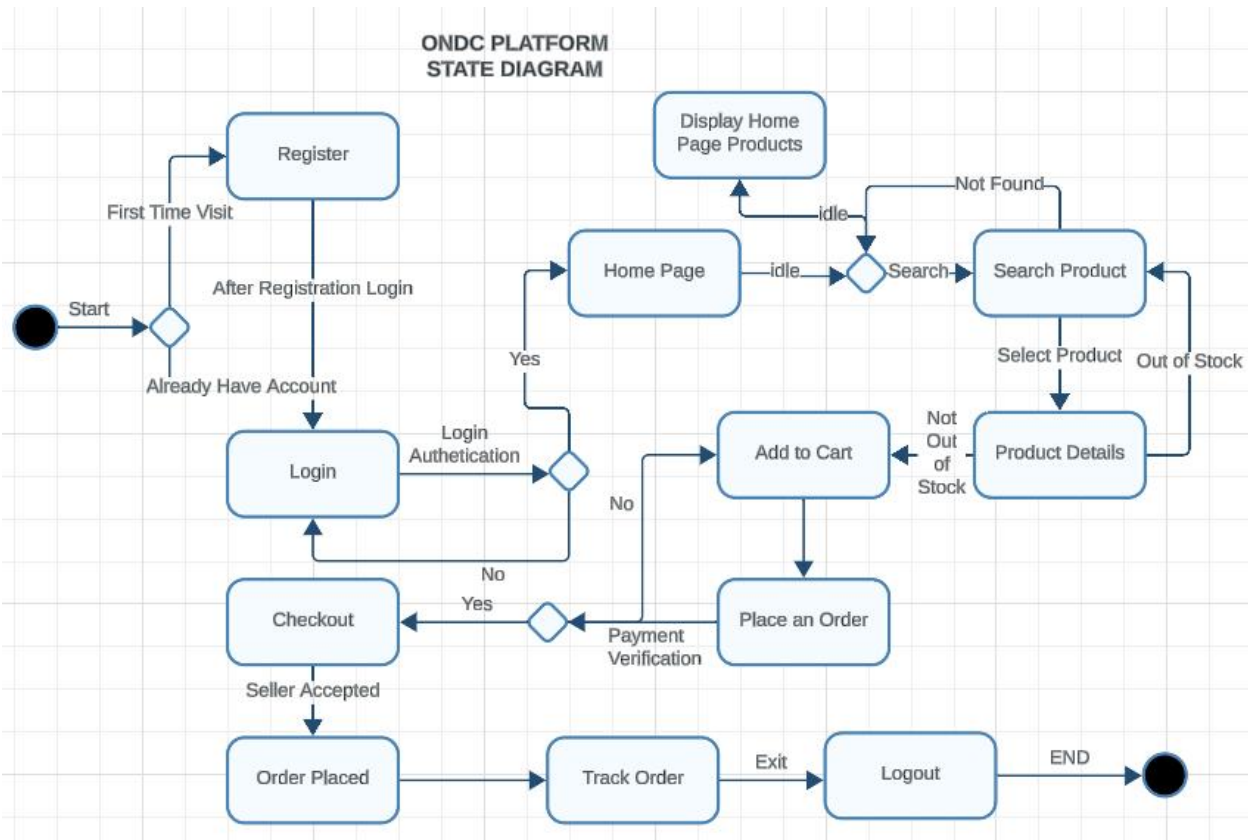


Figure 6: State Diagram of ONDC Platform

4.3.3 Activity Diagram

Activity diagrams represent the flow of activities within a system or process. Figure 10 depicts the Activity Diagram for the ONDC Platform, showcasing the sequence of actions involved in the purchasing journey:

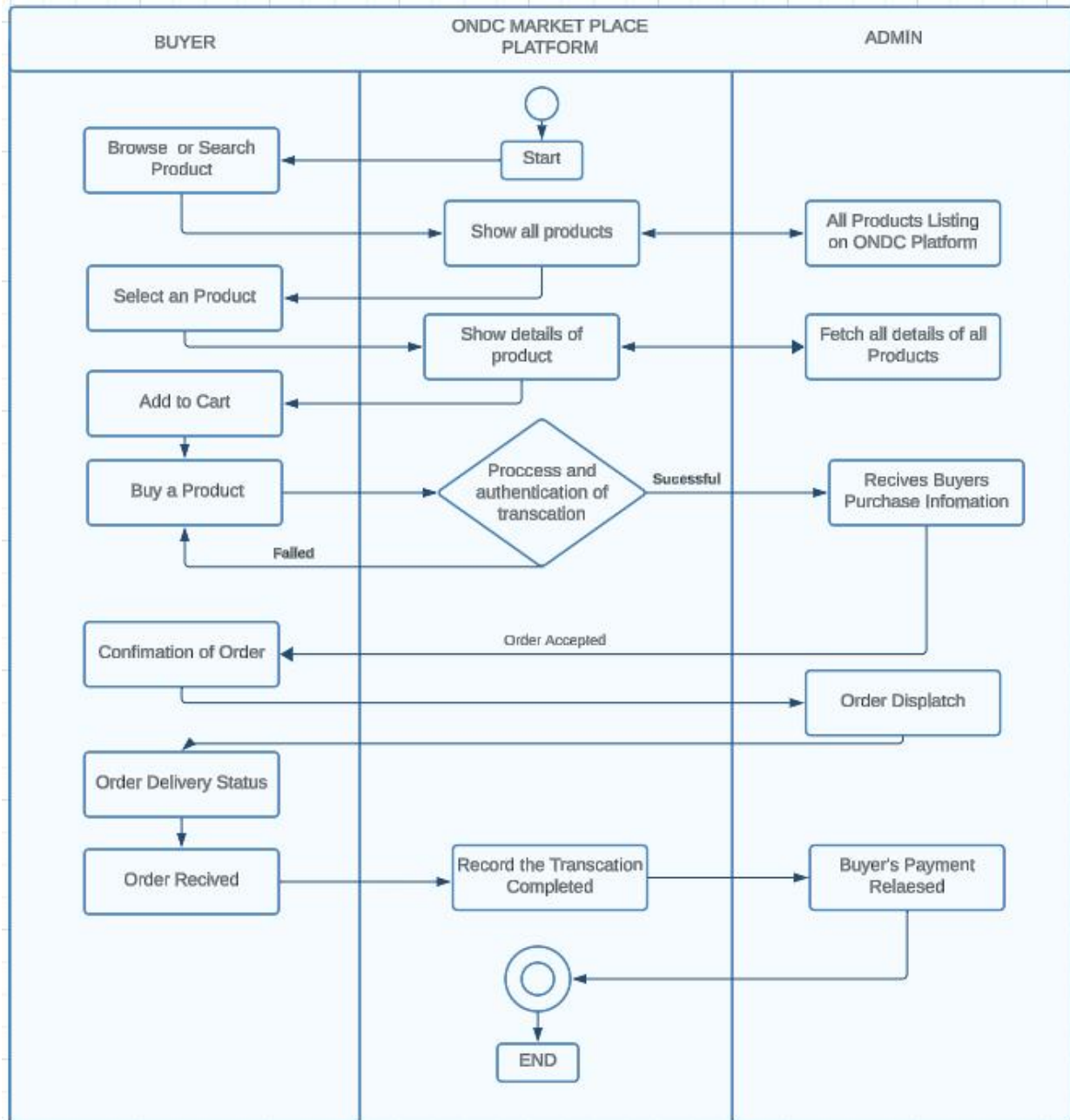


Figure 7: Activity Diagram of ONDC Platform

4.5 Sequence Diagram

A sequence diagram is a type of interaction diagram that depicts the interactions between objects or components in a system over time. It illustrates the sequence of messages exchanged between these entities to accomplish a specific task or scenario.

Here's an explanation of sequence diagrams:-

1. Sequence diagrams involve participants representing system objects.
2. Each participant is shown with a lifeline indicating its presence over time.
3. Messages signify communication, with synchronous, asynchronous, and return types.
4. Activation boxes show when participants process messages.
5. Interactions display message order from top to bottom.
6. Participant actions and responses are depicted on lifelines.
7. Sequence diagrams emphasize dynamic system behavior and timing.

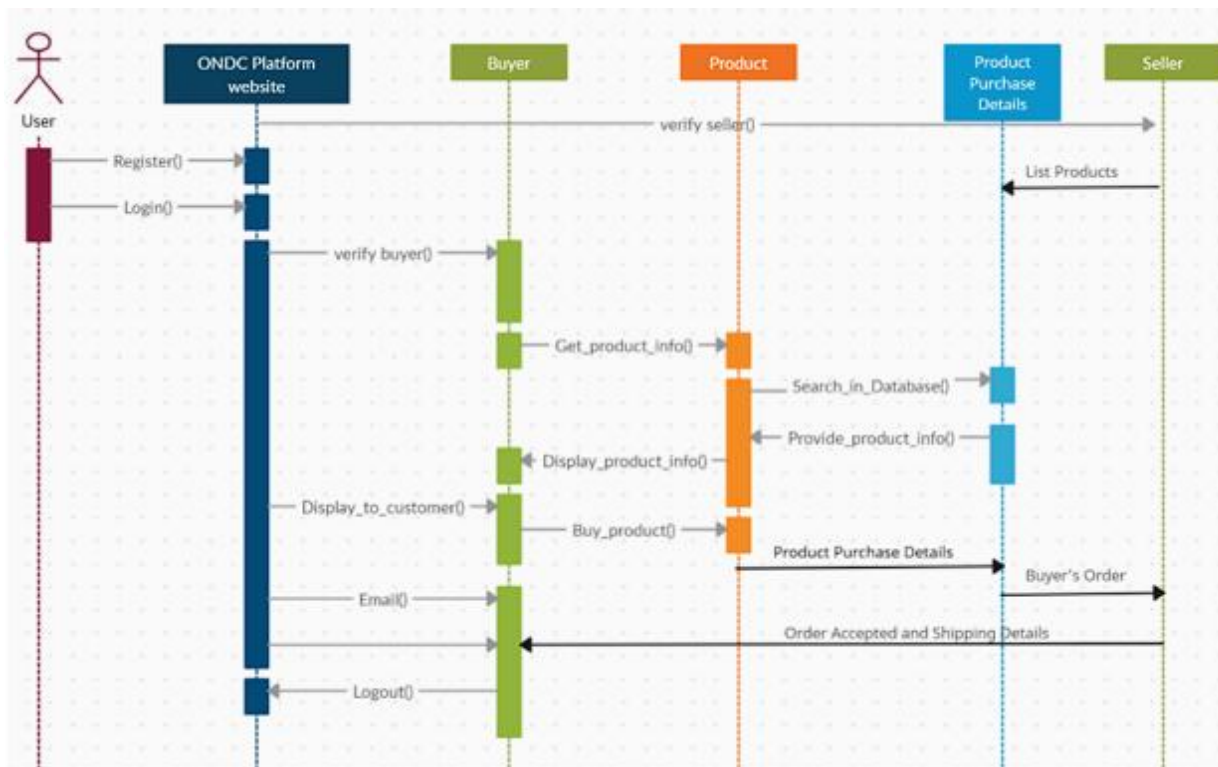


Figure 8: Sequence Diagram

4.7 Component and Deployment Diagram

Component and Deployment Diagrams illustrate the components of the system and their deployment architecture. Unfortunately, the provided diagrams did not include Component and Deployment Diagrams.

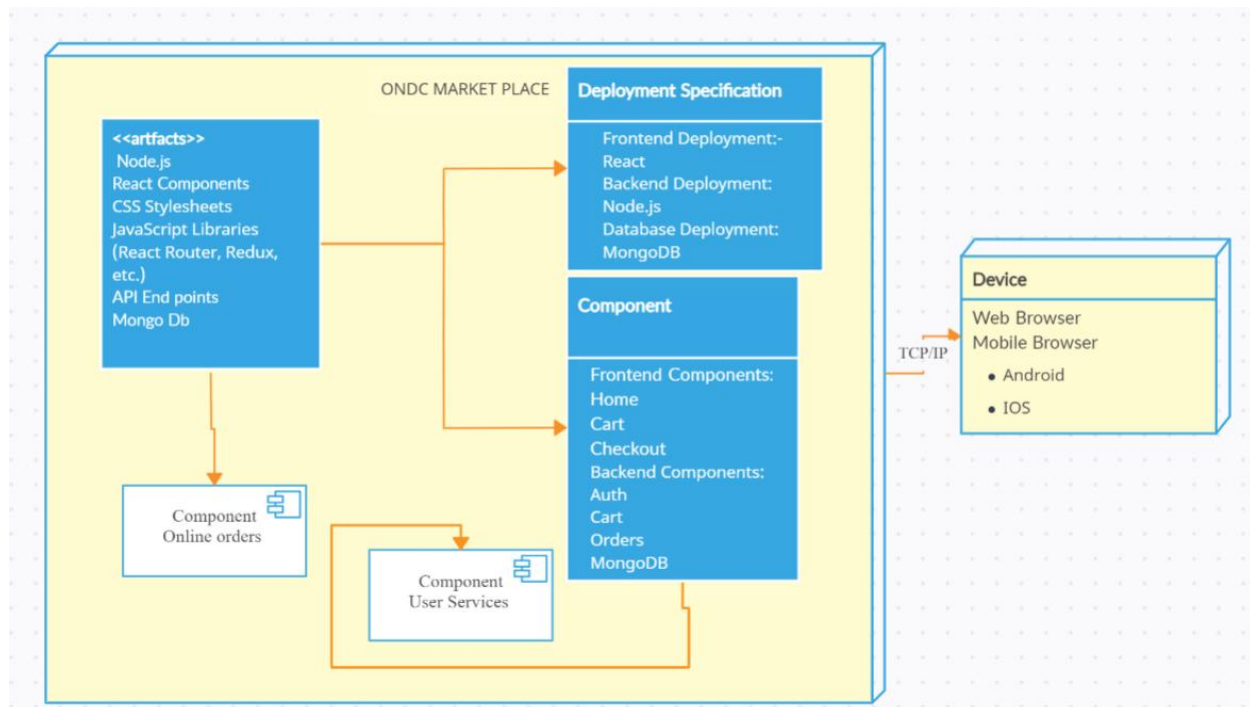


Figure 8: Deployment Diagram for ONDC

4.1 Gantt Chart

Gantt charts provide a visual representation of project tasks, durations, and dependencies. The figures below illustrate the project timeline and task allocation:

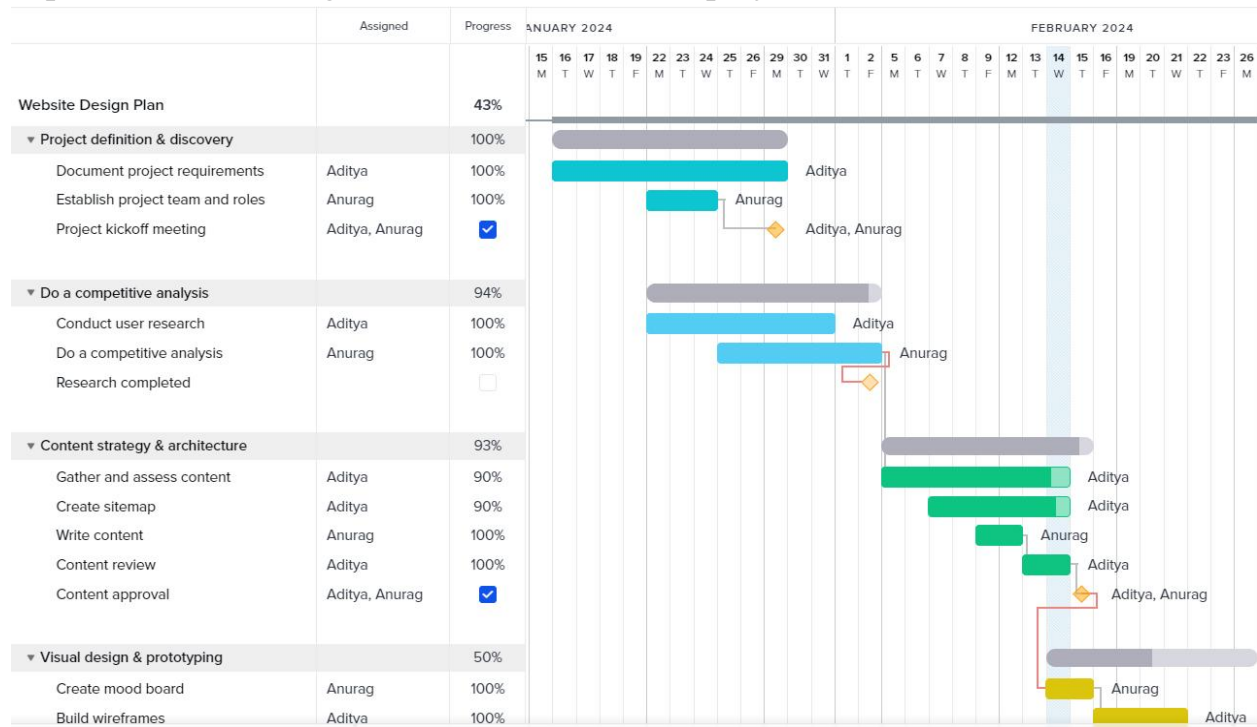


Figure 9: Gantt Chart

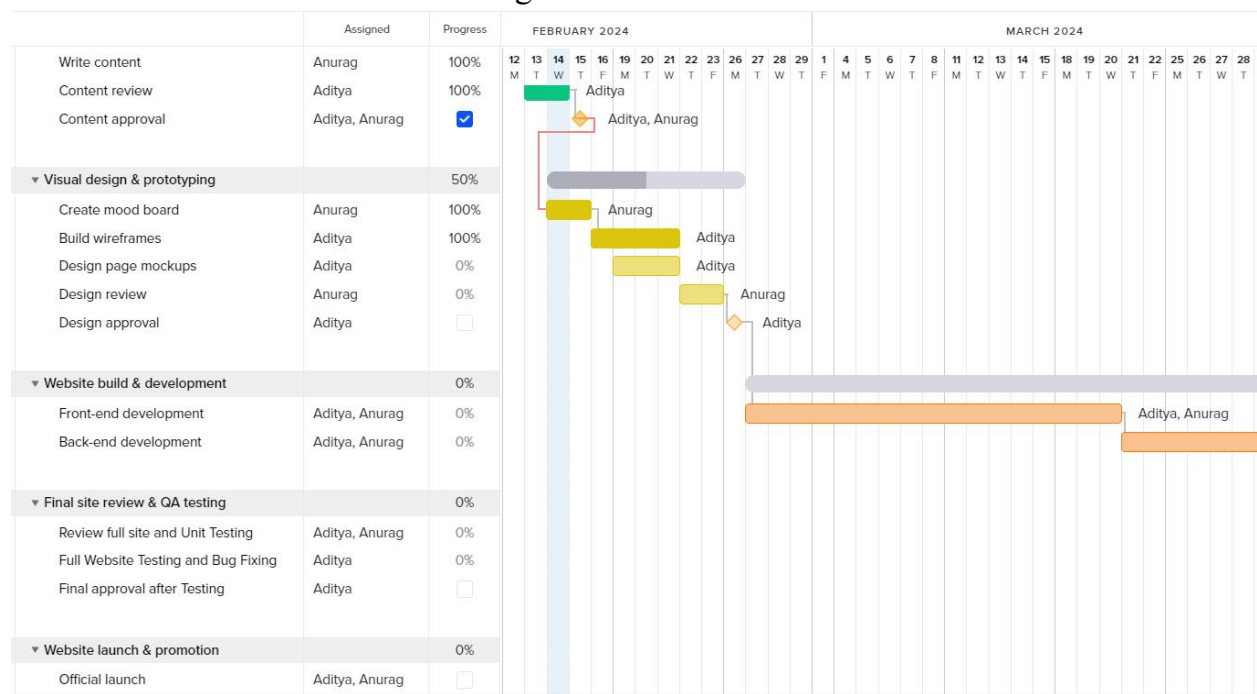


Figure 10: Gantt Chart

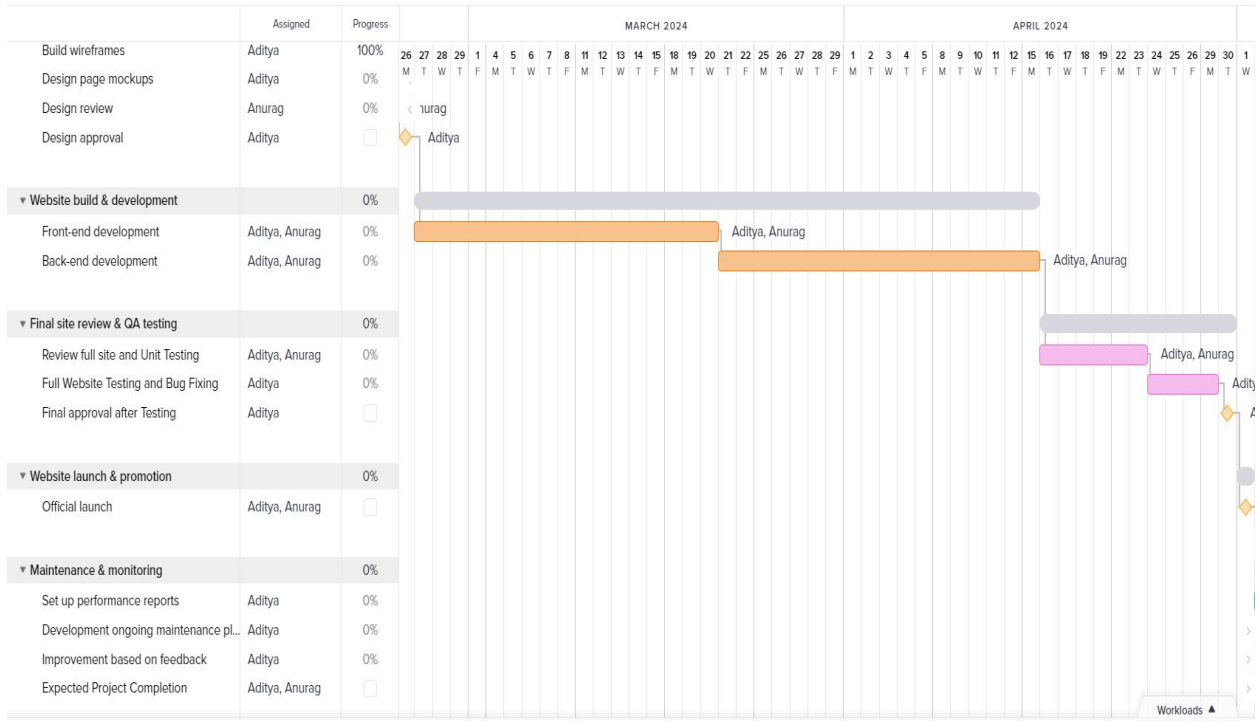


Figure 11: Gantt Chart

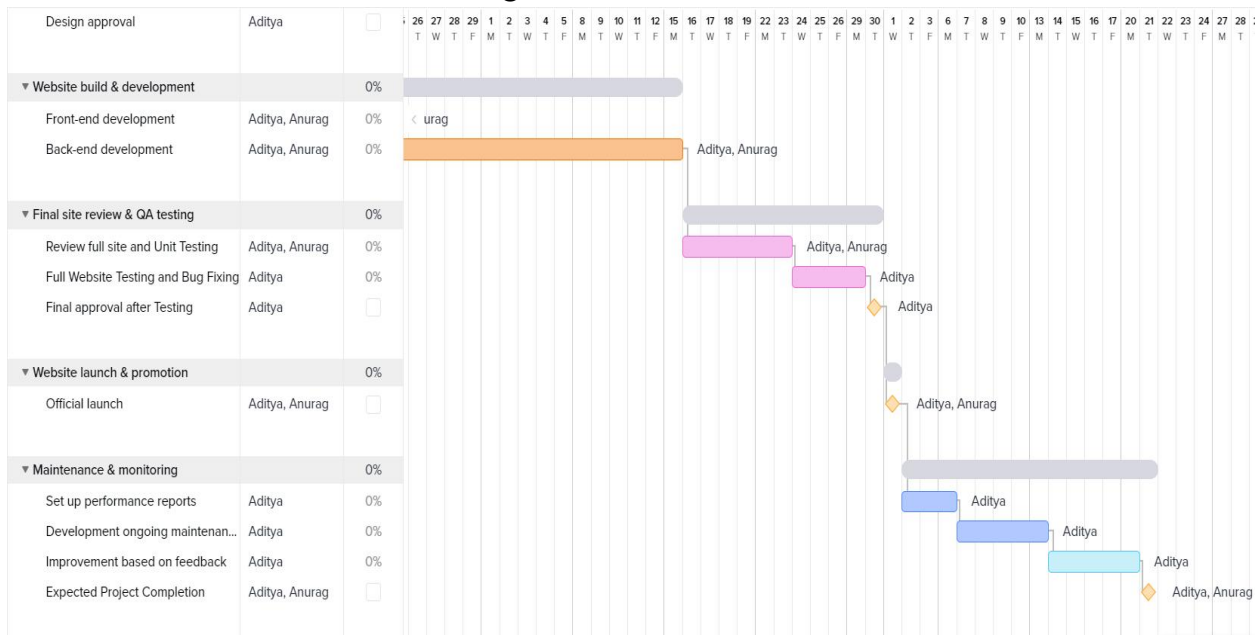


Figure 12: Gantt Chart

This chapter provides an overview of the design considerations for the ONDC Platform E-commerce Marketplace, leveraging the diagrams and information provided.

Chapter 5

Implementation

In this chapter, we transition from the design phase to the implementation phase of the ONDC Platform E-commerce Marketplace project. Drawing upon the design specifications outlined in Chapter 4, we detail the process of translating the design into a functional system.

5.1 Development Environment

The development environment for the ONDC Platform project consists of the following components:

- **Programming Language:** TypeScript for server-side and client-side development.
- **Frameworks/Libraries:** Node.js for backend development, React.js for frontend development.
- **Database:** MongoDB for storing application data.
- **Tools:** Visual Studio Code for code editing, Git for version control, and Postman for API testing.

5.2 Backend Development

5.2.1 Server Setup

The backend server is implemented using Node.js. We set up the server environment, including configuring routes, middleware, and database connections.

5.2.2 Authentication with Firebase

Firebase Authentication is integrated into the backend to handle user authentication. This enables secure user registration, login, and authentication processes using Google accounts.

5.2.3 API Development

We develop our own product order APIs to handle various functionalities such as user management, product listing, cart management, order processing, and payment integration. These APIs are implemented using Express.js, a web application framework for Node.js.

5.2.4 Integration with Stripe

Integration with the Stripe payment gateway is implemented to facilitate secure payment processing.

5.3 Frontend Development

5.3.1 User Interface Design

The frontend user interface is designed using React.js, a JavaScript library for building user interfaces. We create reusable components for different UI elements such as navigation bars, product listings, shopping carts, and checkout forms.

5.3.2 State Management

Redux, a predictable state container for JavaScript apps, is utilized for managing application state. This helps in maintaining a centralized state and enables efficient data flow between components.

5.3.3 Integration with Chart.js

Chart.js is integrated into the frontend to create visually appealing charts and graphs for analyzing sales data, product trends, and user statistics. This enhances the dashboard experience for both administrators and sellers.

5.4 Database Management

5.4.1 MongoDB Integration

MongoDB, a NoSQL database, is used for storing application data such as user profiles, product details, orders, and transaction information. Mongoose, an Object Data Modeling (ODM) library for MongoDB and Node.js, is employed for data modeling and database operations.

5.5 Testing

5.5.1 Unit Testing

Unit tests are written using Jest and Enzyme to ensure the correctness of individual components and functions. Mocking libraries are utilized to simulate API responses and database interactions.

5.5.2 Integration Testing

Integration tests are conducted to verify the interactions between frontend and backend components, as well as external services such as Firebase Authentication and the Stripe payment gateway.

5.6 Deployment

5.6.1 Continuous Integration/Continuous Deployment (CI/CD)

CI/CD pipelines are set up using tools like GitHub Actions or Jenkins to automate the build, test, and deployment processes. This ensures rapid and reliable deployment of new features and updates to the production environment.

5.7 Monitoring and Maintenance

5.7.1 Logging and Error Tracking

Logging mechanisms are implemented to record application events and errors.

5.7.2 Performance Monitoring

Performance monitoring tools are utilized to monitor application performance metrics such as response time, latency, and resource utilization. This helps in identifying and resolving performance bottlenecks.

5.8 Working Demo

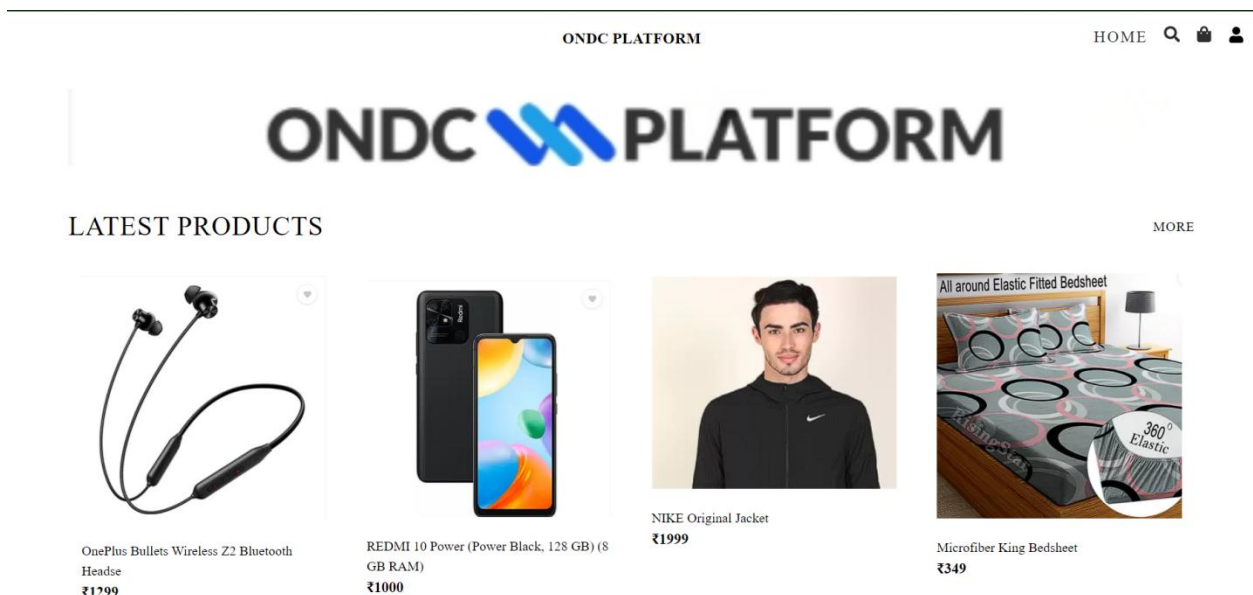


Figure 13: Home Page

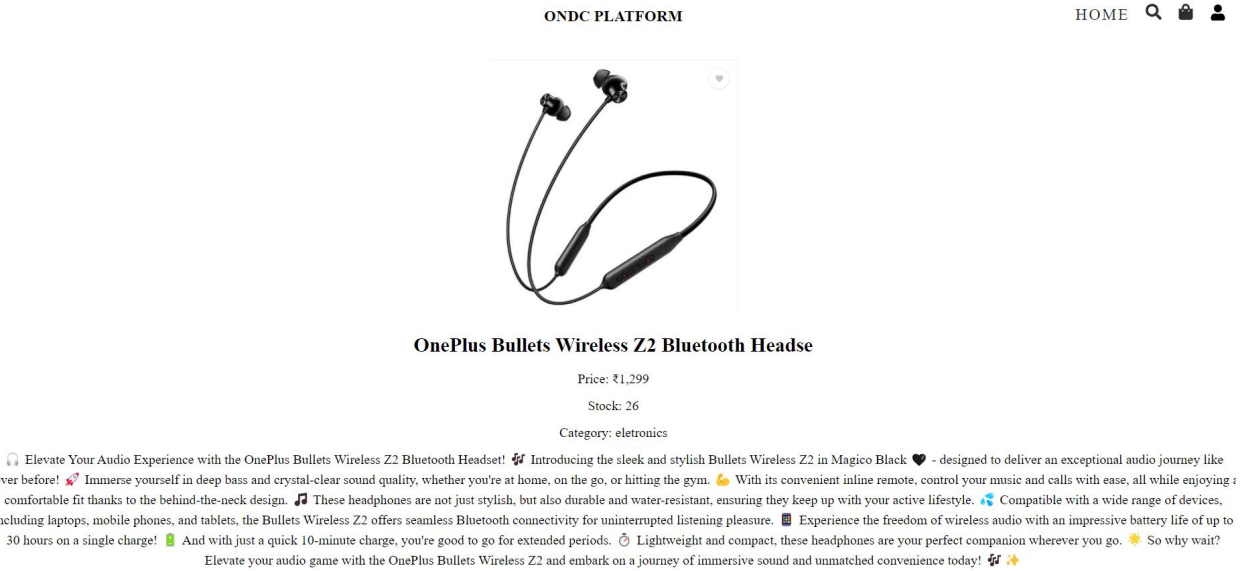


Figure 14: Product Details Page

LATEST PRODUCTS

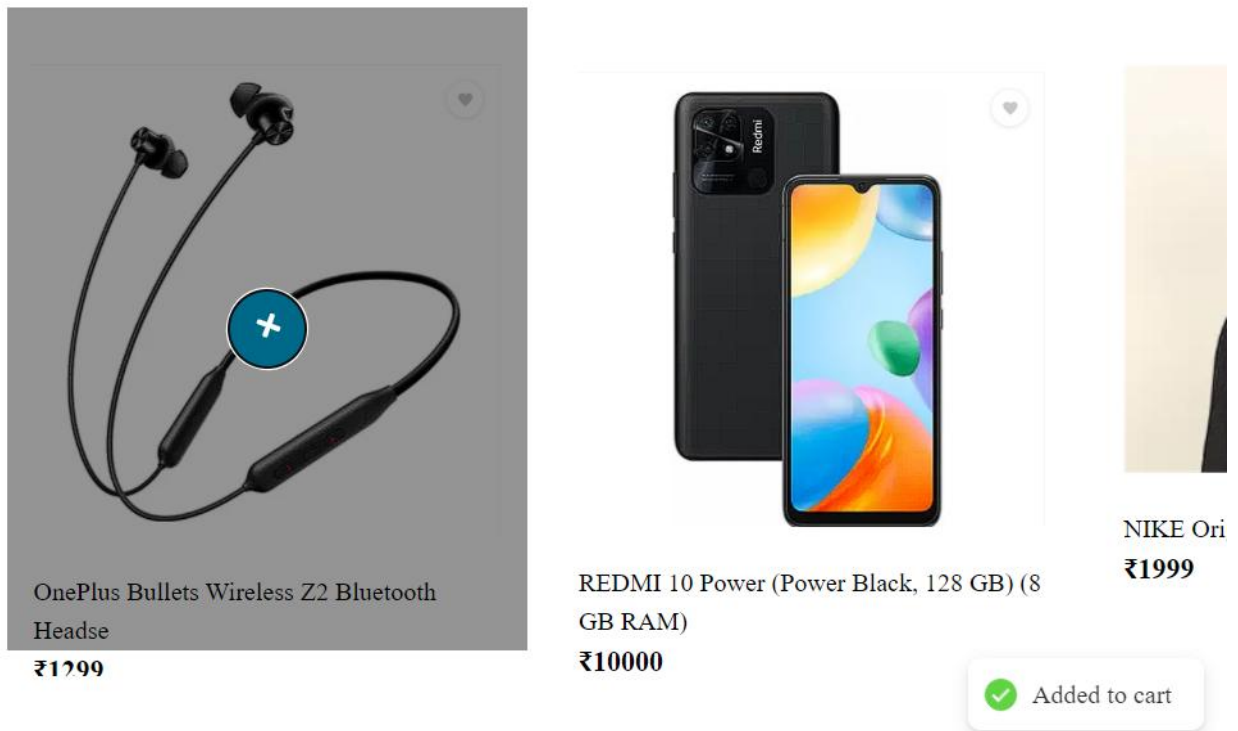


Figure 15: Add to Cart Button

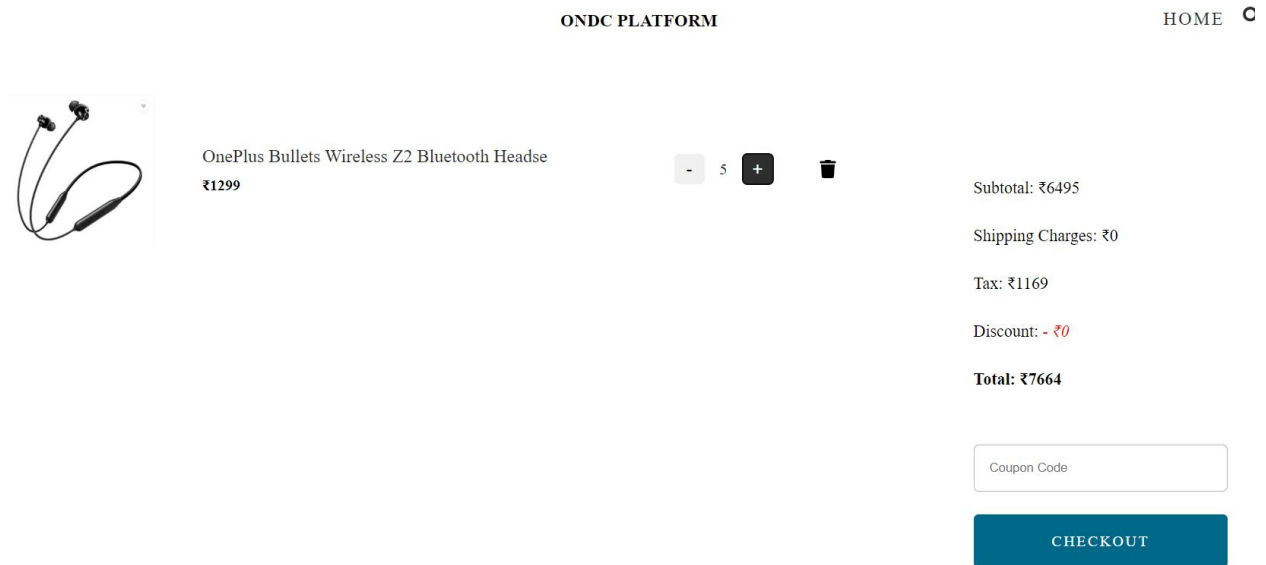


Figure 16: Cart Page with increased Quantity

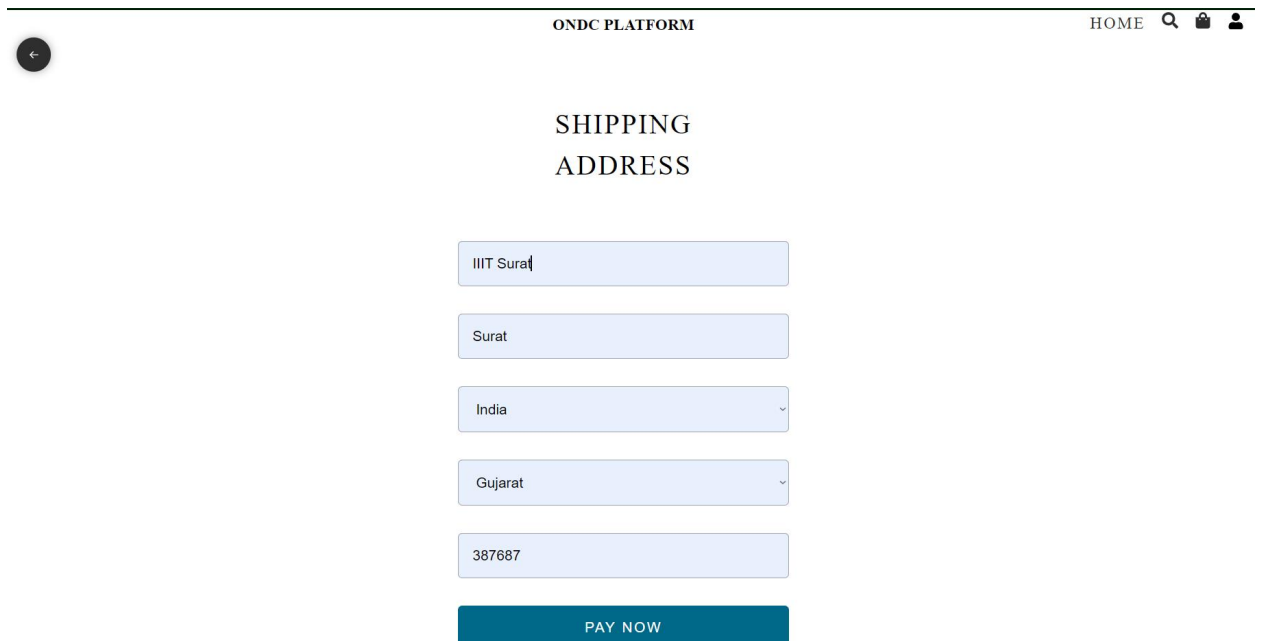




Figure 17: Shipping Address Page after Add to Cart

ONDC PLATFORM


Card number

1234 1234 1234 1234 

Expiration CVC

MM / YY CVC 

Country


India 

Pay


Figure 18: Payment Page

ONDC PLATFORM


Card number

4000 0035 6000 0008 

Expiration CVC

12 / 24 000 

Country

India 

Pay

Figure 19: Payment Details

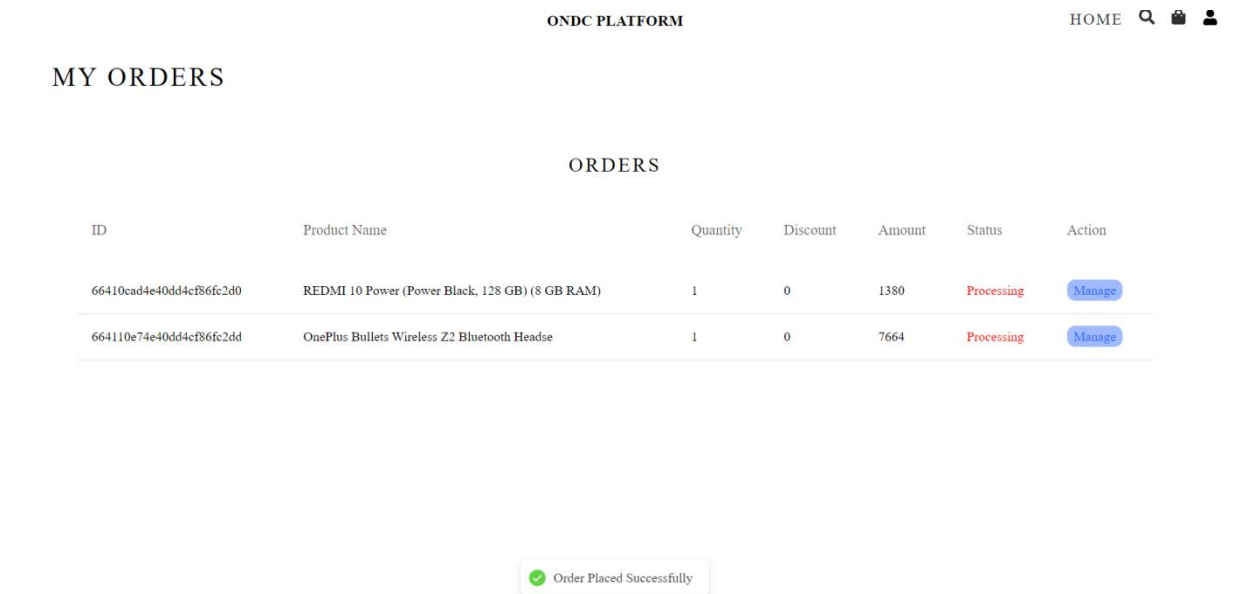


Figure 20: Payment Success and Order Placed

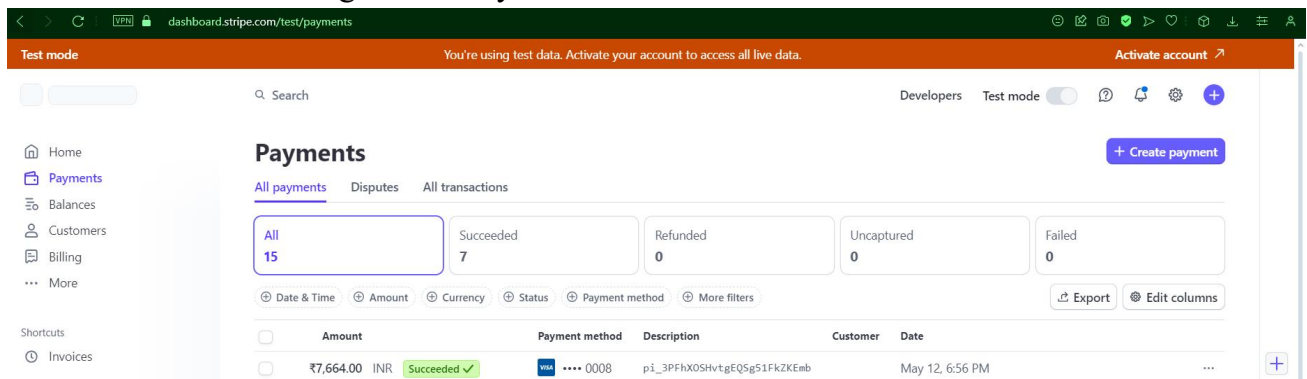


Figure 21: Payment Received Confirmation

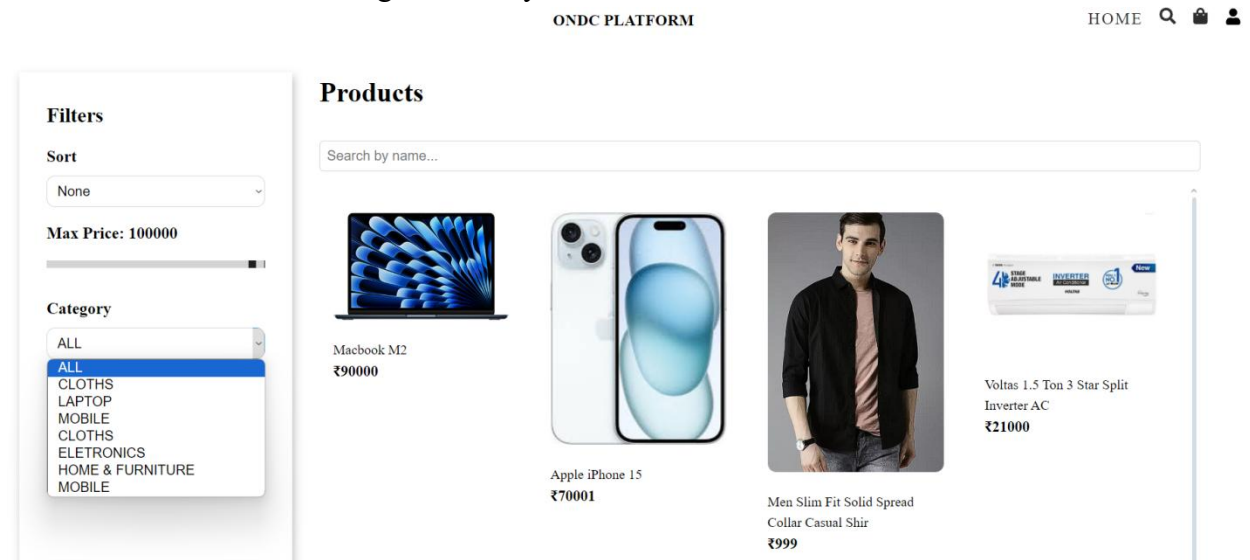


Figure 21: Filter Products based on Price & Category

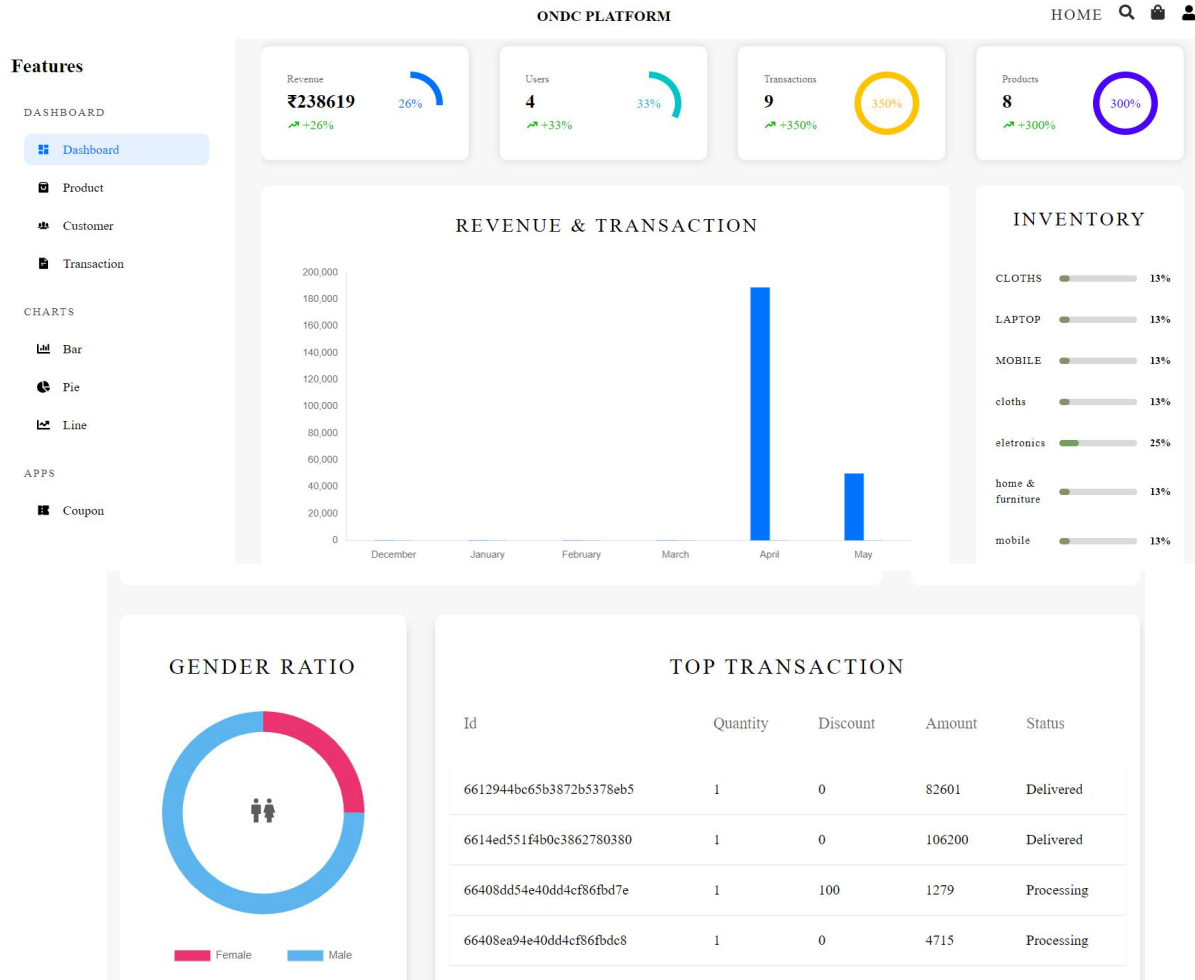


Figure 22: Admin Dashboard

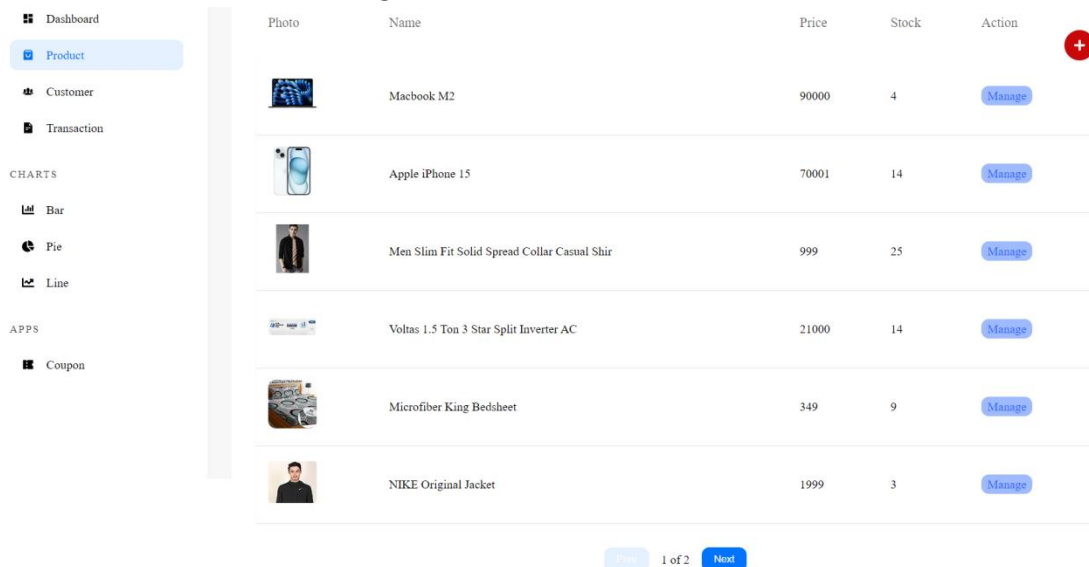


Figure 23: Product Dashboard

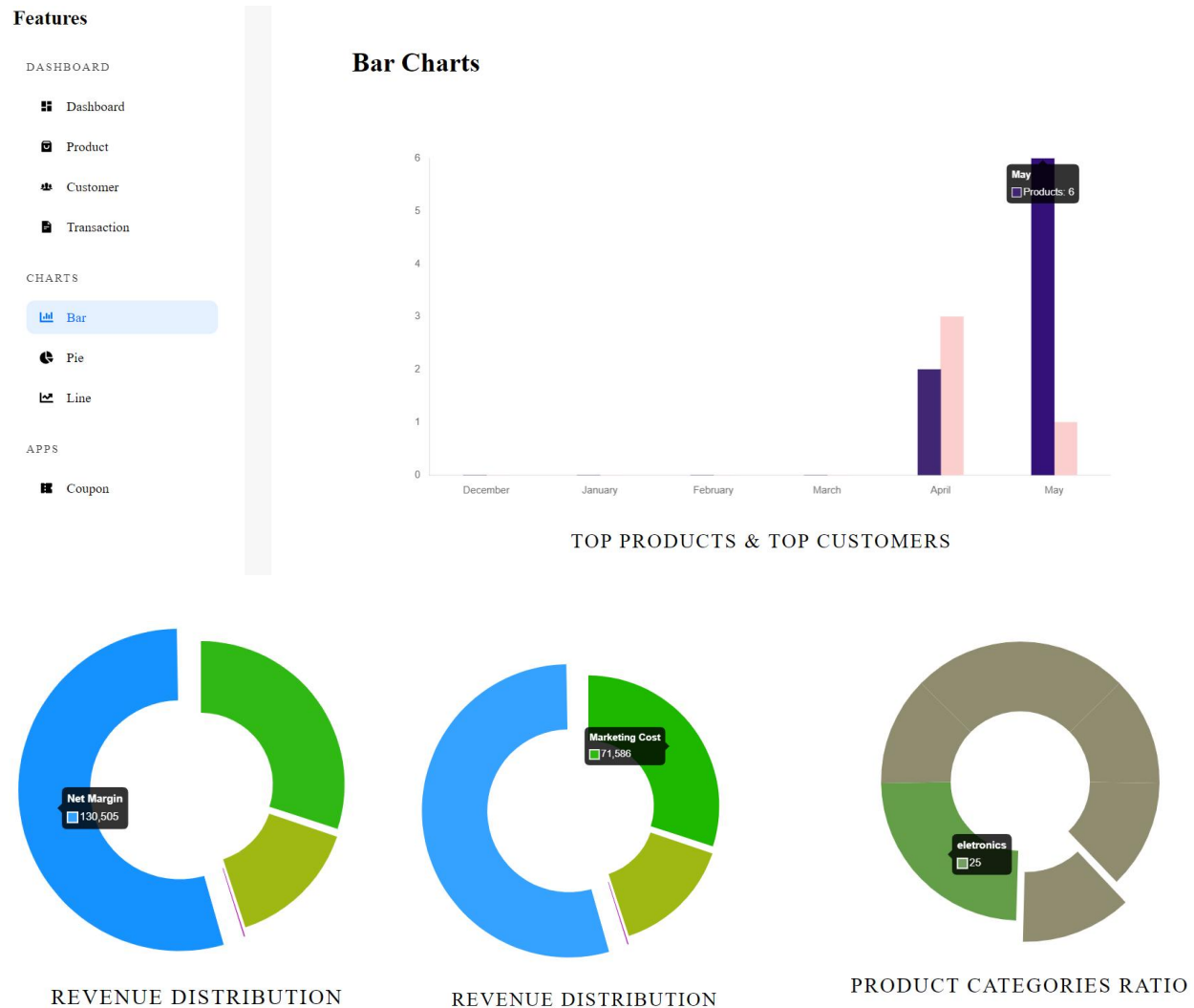


Figure 24: Charts Dashboard

In this chapter, we have discussed the implementation details of the ONDC Platform E-commerce Marketplace project, covering backend development, frontend development, database management, testing, deployment, and monitoring. The successful implementation of these components forms the foundation of a robust and functional e-commerce platform.

Chapter 6

Testing and Experimental Results

This chapter delves into the rigorous testing procedures employed to validate the functionality, reliability, and performance of the ONDC Platform E-commerce Marketplace. Various testing methodologies, including unit testing, integration testing, and user acceptance testing, were conducted to ensure the robustness of the system. Additionally, this chapter presents the experimental results obtained from testing sessions, providing valuable insights into the system's behavior, performance metrics, and user satisfaction.

Testing Methodologies

1. Unit Testing

Unit testing involved testing individual components or modules of the system in isolation to verify their correctness and functionality. This was achieved by me doing tests of individual modules every time when created.

2. Integration Testing

Integration testing focused on validating the interaction and integration between different components or modules of the system. By simulating real-world scenarios and data exchanges, integration testing ensured the seamless operation of the entire system.

In my case I have performed the integration testing after making frontend part then backend part.

3. User Acceptance Testing (UAT)

User acceptance testing involved real users interacting with the system to assess its usability, intuitiveness, and adherence to requirements. Feedback from users was collected and analyzed to identify areas for improvement and refinement.

Experimental Results

1. Performance Metrics

Performance metrics, including response times, throughput, and system resource utilization, were measured under various load conditions to evaluate the system's scalability and efficiency.

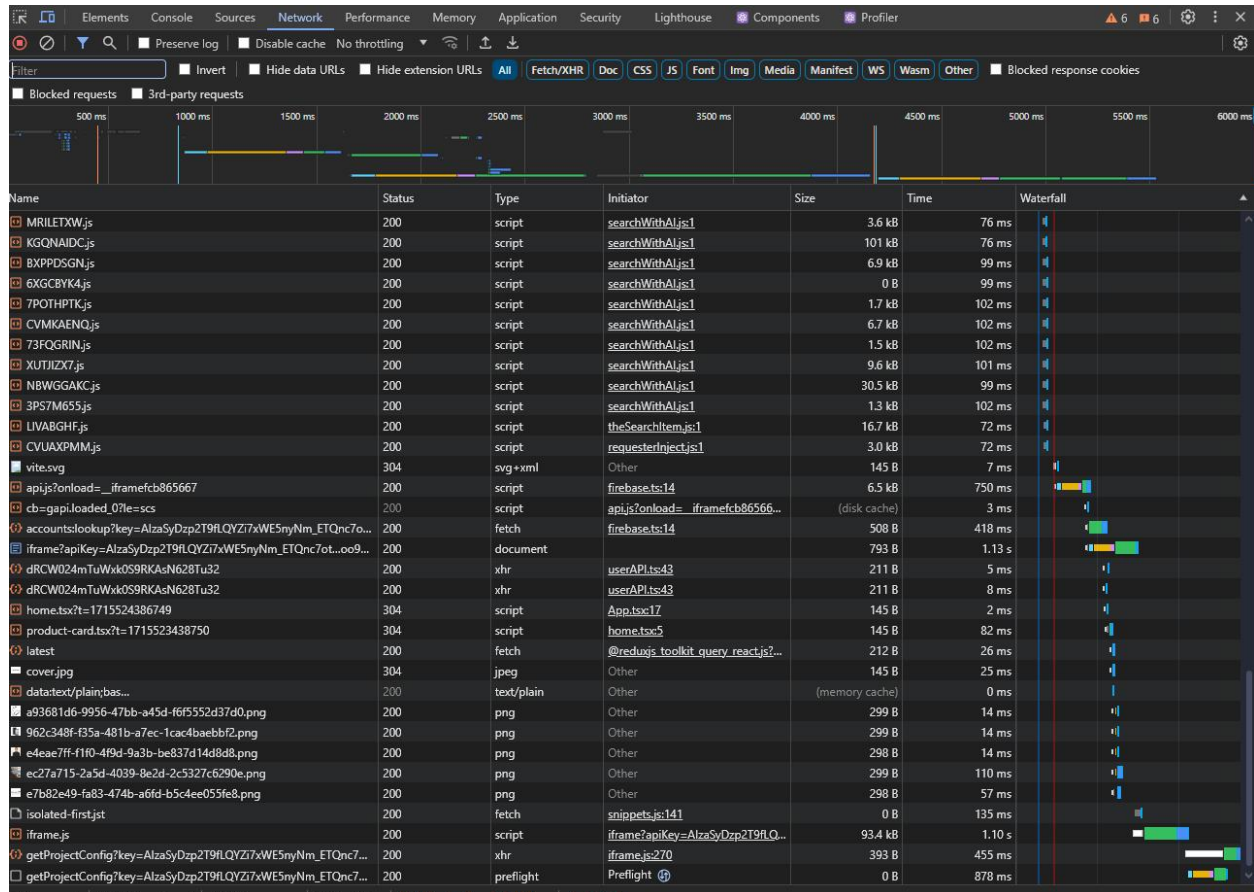


Figure 25: Performance Testing Analysis of Webpage load

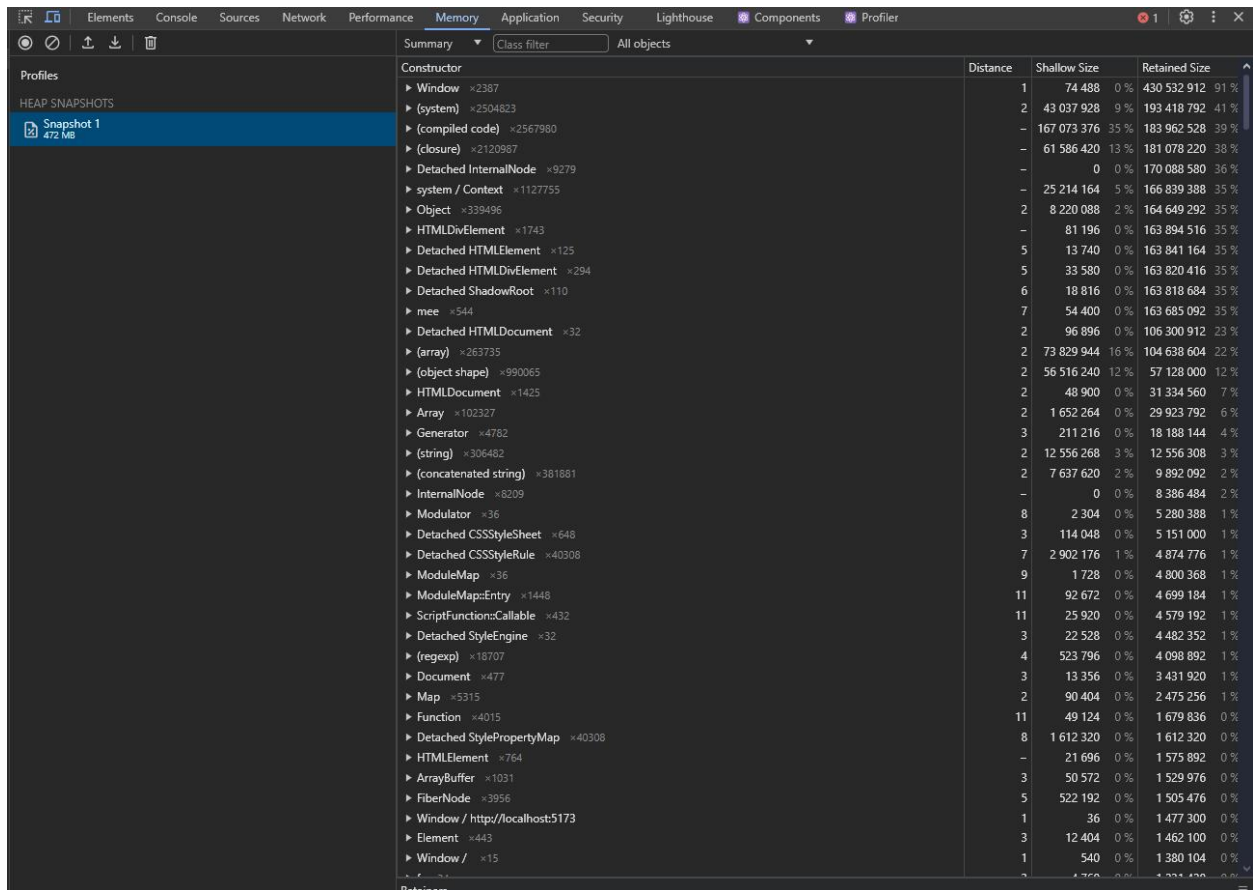


Figure 26: Memory Test

2. User Satisfaction

User satisfaction surveys and feedback sessions were conducted to gauge user perceptions, preferences, and overall satisfaction with the ONDC Platform. Insights obtained from users were instrumental in refining the user experience and addressing any usability issues.

3. API Testing

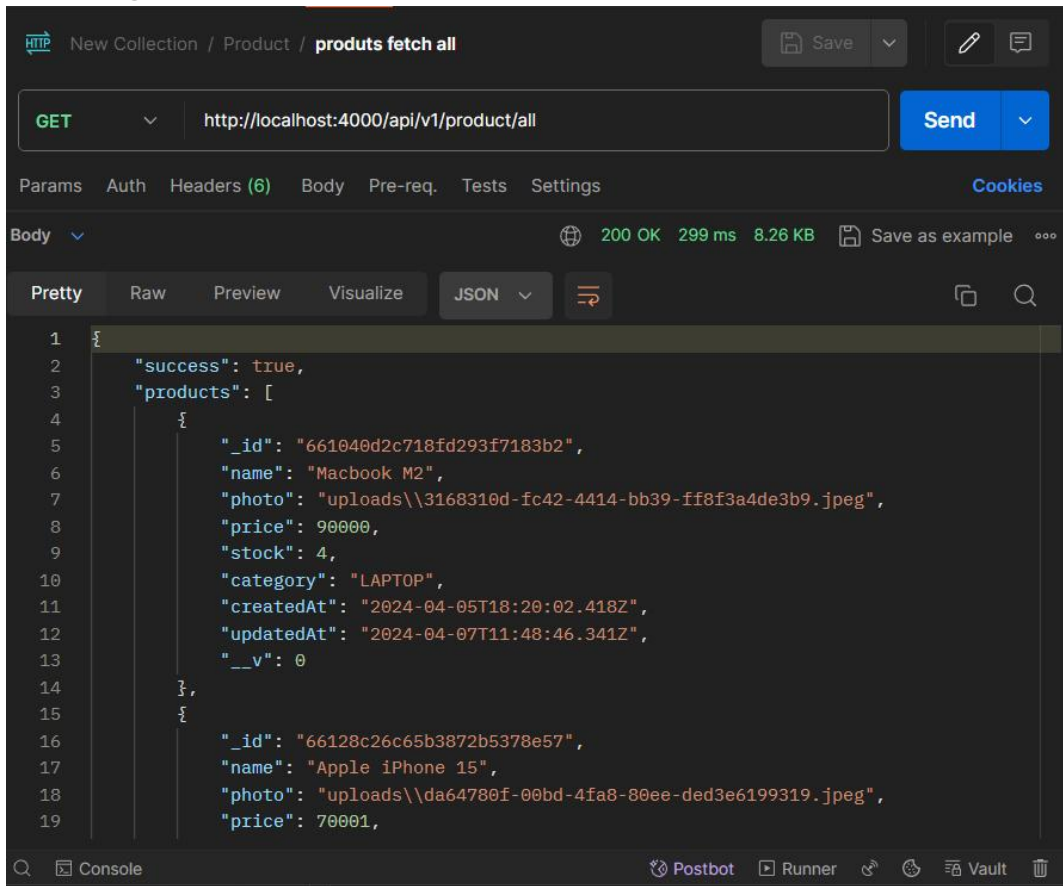


Figure 27: Fetching all Products details from backend

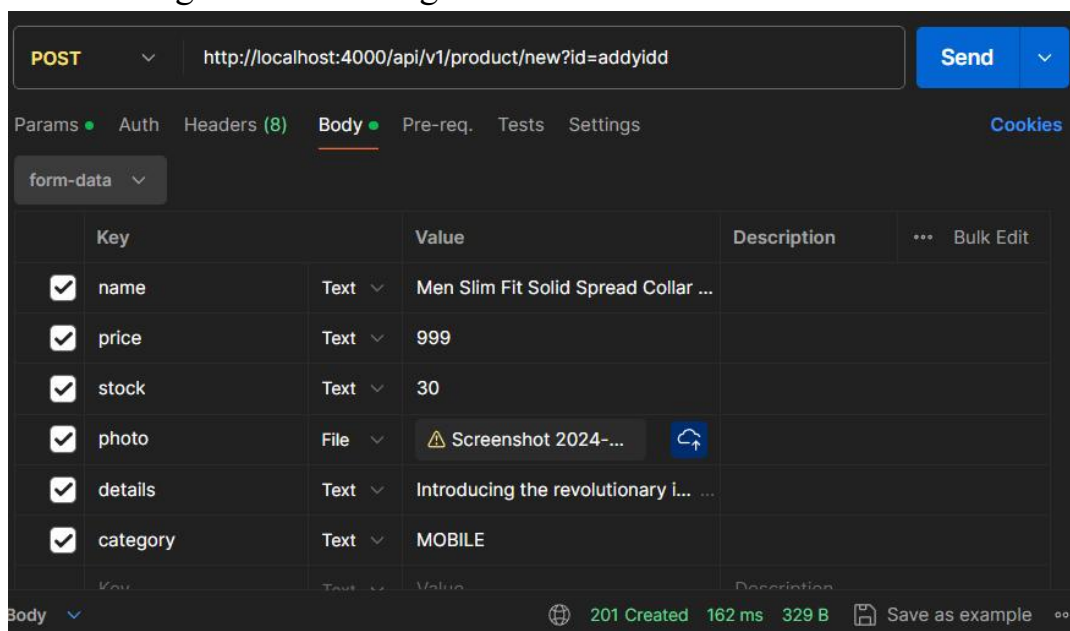


Figure 28: Posting (Adding) the Product to backend

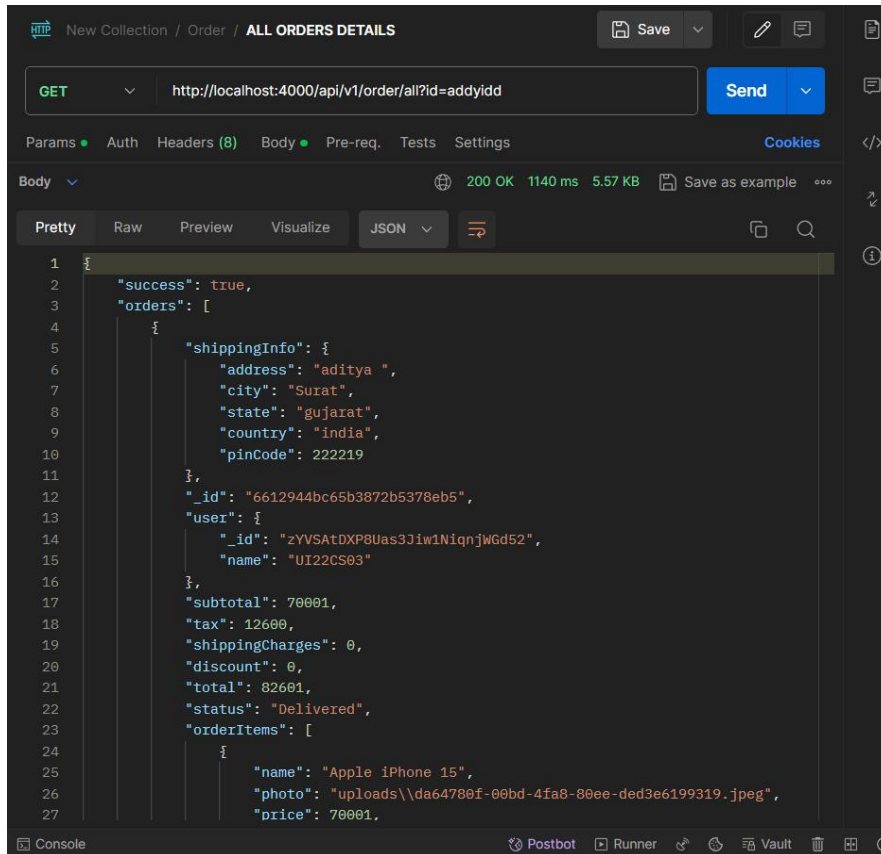


Figure 29: Fetch all Order data

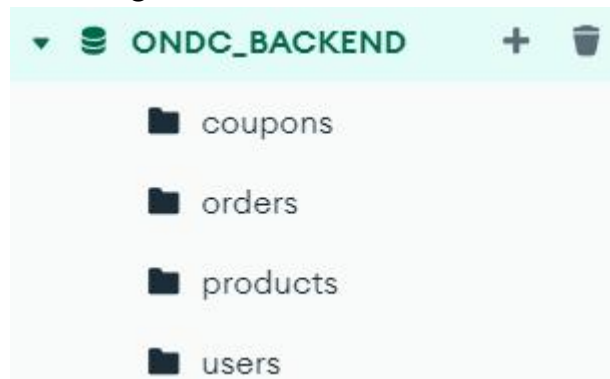


Figure 30: MongoDB

Conclusion

This chapter deals with testing methodologies employed and the experimental results obtained. The comprehensive testing and analysis conducted in this chapter validate the effectiveness, reliability, and performance of the ONDC Platform E-commerce Marketplace, paving the way for its successful deployment and adoption in the market.

Chapter 7

Future Scope and Conclusion

Conclusion:

In conclusion, the development and implementation of the ONDC Platform E-commerce Marketplace represent a significant milestone in the realm of online retail. The platform's robust architecture, intuitive user interface, and comprehensive feature set make it a valuable asset for both buyers and sellers in the digital commerce landscape. Throughout the project lifecycle, meticulous attention was paid to every aspect of design, functionality, and user experience, resulting in a high-quality, scalable, and user-friendly platform.

The successful completion of the ONDC Platform project underscores the effectiveness of agile methodologies, collaborative teamwork, and iterative development processes. By adhering to industry best practices and standards, we have ensured the reliability, security, and performance of the platform, thereby meeting the needs and expectations of our stakeholders.

Future Scope

While the ONDC Platform has achieved significant milestones, there remains ample opportunity for further enhancement and expansion. The following areas represent potential avenues for future development:

1. ONDC Network Real world integration

Integration with real ONDC network is my main aim . I have buided this project to continue more improvement overtime to launch my own e commerce website.

2. Rating and Review Feature

Implementing a rating and review system will enable users to provide feedback and share their experiences with products and sellers. This feature enhances transparency, builds trust, and empowers users to make informed purchasing decisions.

3. Enhanced Product Details

Expanding the product details section to include comprehensive information, such as specifications, reviews, ratings, and user-generated content, will enrich the browsing and decision-making process for buyers. Additionally, integrating multimedia elements such as images and videos can provide a more immersive shopping experience.

4. Recommendation Management System

Integrating an AI-powered recommendation management system will enable personalized product recommendations based on user preferences, browsing history, and purchase behavior. By leveraging machine learning algorithms, the platform can offer relevant suggestions, cross-selling opportunities, and targeted promotions, thereby enhancing user engagement and satisfaction.

5. Advanced Algorithms for Improved Results

Continuously refining and optimizing the algorithms powering search, recommendation, and sorting functionalities will ensure more accurate, relevant, and timely results for users. By leveraging data analytics and machine learning techniques, the platform can adapt to evolving user preferences and market trends, delivering superior outcomes and driving business growth.

6. Feature Expansion Based on User Feedback

Regularly soliciting user feedback and incorporating feature requests and suggestions into the development roadmap will foster a culture of continuous improvement and innovation. By prioritizing user-centric design and functionality enhancements, the platform can remain agile, responsive, and competitive in the dynamic e-commerce landscape.

Summary:

As a student of IIIT Surat, I embarked on the journey of creating the ONDC Platform E-commerce Marketplace project, a significant endeavor that encapsulated the culmination of my learning experiences and technical prowess. Throughout the development process, I delved into various cutting-edge technologies, including the MERN (MongoDB, Express.js, React.js, Node.js) stack, TypeScript, and Firebase Authentication, leveraging their capabilities to craft a robust and feature-rich platform.

The project's scope encompassed the design, development, and implementation of a comprehensive e-commerce solution tailored to leverage the Open Network for Digital Commerce (ONDC) infrastructure. By integrating the ONDC ecosystem and harnessing its potential, the platform aimed to democratize e-commerce, fostering inclusivity and accessibility for sellers across diverse backgrounds and geographic locations.

From conceptualization to deployment, meticulous attention was paid to every aspect of the project, including frontend and backend development, database management, testing, and deployment. Agile methodologies and iterative development processes were embraced, enabling rapid iteration and continuous improvement throughout the project lifecycle.

Key highlights of the project included the creation of an intuitive user interface using React.js, seamless backend operations with Express.js and Node.js, secure user authentication with Firebase, and integration with external services such as Stripe for payment processing. The platform's functionality spanned product management, order processing, data visualization, and system administration, offering a holistic solution for buyers and sellers alike.

Conclusion:

The completion of the ONDC Platform project marks a significant milestone in my journey as a student and aspiring technologist. It represents not only the culmination of technical skills acquired but also the embodiment of teamwork, innovation, and problem-solving abilities nurtured throughout my academic tenure at IIIT Surat.

Looking ahead, the project's success underscores the importance of continuous learning and adaptability in today's dynamic technological landscape. As I reflect on the challenges overcome and lessons learned, I am inspired to pursue further innovation and exploration in the realm of digital commerce and beyond.

Moreover, the project's future scope presents exciting opportunities for expansion and enhancement, from real-world integration with the ONDC network to the implementation of advanced features such as rating and review systems, personalized recommendations, and algorithmic optimizations. By embracing emerging technologies and user-centric design principles, the ONDC Platform is poised to redefine the online shopping experience and contribute to the evolution of digital commerce in India.

In conclusion, I am immensely grateful for the invaluable guidance and support of my mentors, faculty members, and peers who have played instrumental roles in shaping this project and my personal growth as a developer. As I embark on new endeavors, I carry with me the lessons learned and experiences gained from the creation of the ONDC Platform, fueling my passion for innovation and excellence in the ever-evolving world of technology.

REFERENCES

Web Development Frameworks & Libraries

- **React:**
 - Official Website: <https://reactjs.org/>
 - Documentation: [invalid URL removed]
 - Tutorial: <https://reactjs.org/tutorial/tutorial.html>
- **ONDC:**
 - Official Website: <https://ondc.org>
- **React Router:**
 - Official Website: <https://reactrouter.com/>
 - Documentation: <https://reactrouter.com/en/main/start/overview>

Backend Technologies

- **Node.js:**
 - Official Website: <https://nodejs.org/>
 - Documentation: <https://nodejs.org/en/docs/>
- **Express.js:**
 - Official Website: <https://expressjs.com/>
 - Documentation: <https://expressjs.com/en/starter/installing.html>
- **MongoDB:**
 - Official Website: <https://www.mongodb.com/>
 - Documentation: <https://www.mongodb.com/docs/manual/>
- **Firebase (Authentication)**
 - Official Website: <https://firebase.google.com/>
 - Documentation: <https://firebase.google.com/docs>