E-COMMERCE WEBSITE USING MERN STACK

A PROJECT REPORT

Submitted By

AYUSH JHA
(2100290140045)
AYUSH SHARMA
(2100290140047)
AMAN RAGHAVA
(2100290140020)

Submitted in partial fulfilment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Vipin Kumar
Associate Professor

Submitted To

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(JUNE – 2023)

DECLARATION

I hereby declare that the work presented in this report entitled **"E-Commerce Website Using MERN Stack"**, was carried outby me. I have not submitted the matter embodied in this report for the award of any other degreeor diploma from any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, and results, that are not my original contribution. I have usedquotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name – Ayush Jha (2100290140045)

Ayush Sharma (2100290140047)

Aman Raghava (2100290140020)

CERTIFICATE

Certified that **Ayush Jha (21002901400045), Ayush Sharma (2100290140047), Aman Raghava (2100290140020),** have carried out the project work having **"E-Commerce Website Using MERN Stack"** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carriedout by the student himself and the contents of the project report do not form the basis for the awardof any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

Ayush Jha (210029040045)

Ayush Sharma (2100290140047)

Aman Raghava (2100290140020)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Tripathi

Head, Department of Computer Applications

KIET Group of Institutions, Ghaziabad

ABSTRACT

The Business to Consumer (B2C) aspect of electronic commerce (e-commerce) is the most visible business use of the Word Wide Web. The primary goal of an e-commerce site is to sell goods or services online. The drivers for electronic commerce (e-commerce) are both technological (under the tremendous pressure of innovation) and business oriented. E-Commerce is now seen as a reality for many businesses and a normal part of a business plan. The immediate benefits, in terms of cost savings, efficiencies and enhanced profitability are clear at every stage in the supply chain.

Adopting e-business is no longer a competitive advantage, but a normal business process, without which an enterprise is unlikely to survive in the new age. Year 2000 saw many Dot-com companies built up and many companies going into E-commerce however now it is a different story, more and more companies are failing, and investors are becoming cautious to invest money into Internet ventures. There is more cash needed then was expected. Some of them had to get on the bandwagon as everybody else were and didn't want to be left behind, and now that the bubble has burst they are facing the consequences.

In order to make a website that can acquire the needs of both customers and retailers, MERN (MongoDB, Express.js framework, Reacts library, NodeJS platform) is one of the powerful stacks that can help us to develop an e-commerce web application.

This is a project with the objective to develop a basic website where a consumer is provided with a shopping cart application and also to know about the technologies used to develop such an application.

Keywords-: B2C (Business to Consumer), e-Commerce, Innovation, MERN, Web Application

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help, and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Arun Tripathi**, Professor, and Head of, the Department of Computer Applications, for his insightful comments and administration help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot in many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kinds of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

AYUSH JHA (2100290140045)
AYUSH SHARMA (2100290140047)
AMAN RAGHAVA (2100290140020)

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Project Description

E-commerce and online shopping in India is getting a noticeable growth as more usage of internet facilities, high educational standards, changing life style and economic growth of the country reasons in the demand of ecommerce techniques and tools. Versatile shopping experience and rapid development of transaction facilities is further boosting opportunities for the remaining market segments. The biggest advantage of e-commerce is the ability to provide secure shopping transactions via the internet and coupled with almost instant verification and validation of credit card transactions. One of the most important issues to be addressed in electronic commerce is the area of services.

Purchasing and selling products and services over the internet without the need of going physically to the market was a time taking process. Online shopping is just like retail store shopping that we do by going to the market, but it is done through the internet. Online shopping has made shopping painless and added more fun. Online stores offer product descriptions, pictures, comparisons, prices, and much more. A few examples of these are Amazon.com, ebay.com, and framt.com and the benefits of online shopping is that by having direct access to consumer, the online stores can offer products that cater to the needs of the consumer, and cookies can be used for tracking the customer selection over the internet or what is of their interest when they visit the site again. Online shopping makes use of digital technology for managing the flow of information, products, and

payment between consumers, site owners, and suppliers. Online shopping can be either B2B (business to business) or B2C (business to consumer).

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The over-project "**E-Commerce Website Using MERN Stack"** provides users to login into the project by entering their name, email, and other details and password link. The account verification link gets in the email, after clicking the link the account will get activated and the account and user can log in to their account and sees the products and order the items. The selected items are stored in the card where he /she sees all the products that he/ she ordered in the project. For ordering the project the user registered their Address and select the payment method to pay the price of that product. If the user selects by Debit card/credit card then the customer must enter their card details for doing the online payment functionality we are using the PayPal payment gateway.

PayPal is a popular online payment gateway that provides a secure and convenient way for individuals and businesses to send and receive payments over the internet. It serves as a bridge between buyers and sellers, facilitating transactions without the need to share sensitive financial information.

Users can create a PayPal account and link it to their bank accounts, credit cards, or store funds directly in their PayPal balance. This allows them to make payments to merchants or send money to other PayPal users with just a few clicks. PayPal supports transactions in multiple currencies, making it suitable for international payments.

For merchants, integrating PayPal as a payment option on their website or online store offers several benefits. It enables them to accept payments from customers around the world, provides a trusted and recognizable payment method, and offers built-in security features to protect against fraud and unauthorized transactions. PayPal also provides tools for managing invoices, generating reports, and tracking sales data.

In addition to online payments, PayPal offers additional features such as PayPal One Touch, which allows customers to make purchases with a single touch or click on enabled websites and apps, and PayPal Checkout, which streamlines the payment process by allowing customers to complete transactions without leaving the merchant's website.

Overall, PayPal is a widely adopted payment gateway that simplifies online transactions, enhances security, and provides a seamless payment experience for both buyers and sellers.

With the advancement in technology and science, people can now do various things in the comfort of their homes and one such thing is online shopping. It has gained a lot of spotlights due to it ever- increasing demand and craze among people. Online shopping refers to the way of purchasing things online without actually going to the physical stores. People nowadays are busy earning their livelihood and they hardly get any time to go shopping, however, with the advent of online shopping, they can now order anything be it clothes, footwear, gadgets, appliances, and much more. There are numerous advantages of online shopping, let’s take an insight into it. Online shopping is the best option for people who do not have much time and are busy in their office and business work. It is a convenient way for the people who cannot withstand crowded places and malls for shopping so they can sit at their home or office and can order anything anytime. Online shopping offers a vast variety of options which is not possible with physical shopping. You can browse through different websites and can choose the product according to your requirements.

Online shopping does not require physical cash and you can make payments through your debit or credit cards although you have an option of cash on delivery. Well, everything comes with some disadvantages as well. Although online shopping is easy and convenient, however sometimes it disappoints you as things ordered online may not seem the same when they arrive at your door, the color, the size, or something else can be different from the actual item. Also, there are some websites that are fake and provide you with great offers to tempt you and befool you in the end. So it is important that we should do online shopping wisely and with much care in order to avoid.

1.2 Literature Review

Electronic Commerce (e-commerce) applications support the interaction between different parties participating in a commerce transaction via the network, as well as the management of the data involved in the process.

The increasing importance of e-commerce is apparent in the study conducted by researchers at the GVU (Graphics, Visualization, and Usability) centre at the Georgia Institute of Technology. In their summary of the findings from the eighth survey, the researchers report that "e-commerce is taking off both in terms of the number of users shopping as well as the total amount people are spending via Internet based transactions".

Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65%), followed by availability of vendor information (60%), no pressure form sales person (55%) and saving time (53%).

Although the issue of security remains the primary reason why more people do not purchase items online, the GVA survey also indicates that faith in the security of ecommerce is increasing. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online

A good e-commerce site should present the following factors to the customers for better usability:

1. Knowing when an item was saved or not saved in the shopping cart.
2. Returning to different parts of the site after adding an item to the shopping cart.
3. Easy scanning and selecting items in a list.
4. Effective categorical organization of products.
5. Simple navigation from home page to information and order links for specific products.
6. Obvious shopping links or buttons.
7. Minimal and effective security notifications or messages.
8. Consistent layout of product information.

1.3 Project Scope

The project scope of an e-commerce MERN (MongoDB, Express, React, Node.js) project encompasses the overall objectives, features, and functionalities that will be included in the development of the e-commerce platform. In this project, the primary goal is to create a robust and user-friendly online marketplace where customers can browse, purchase, and manage products, while administrators can manage inventory, orders, and user accounts. The following paragraphs provide a detailed overview of the project scope.

1. User Management: The e-commerce platform will allow users to register and create accounts. Registered users will have access to features such as profile management, order history, and wish lists. Users will also be able to log in using social media accounts for a seamless experience.

2. Product Catalogue: The platform will offer a comprehensive product catalogue with various categories and subcategories. Users will be able to search, filter, and sort products based on different criteria. Each product will have a detailed product description, pricing information, and images.

3. Shopping Cart and Checkout: Users will be able to add products to their shopping cart, view cart contents, update quantities, and proceed to checkout. The checkout process will include various payment options, such as credit/debit card payments, PayPal, or other popular payment gateways.

4. Order Management: Administrators will have access to an order management system where they can view, process, and track orders. This includes generating invoices, updating order statuses, and managing returns and refunds.

5. Inventory Management: The platform will provide a back-end inventory management system for administrators to track product stock levels, update product information, and manage product variants, such as sizes or colours.

6. User Reviews and Ratings: Users will have the ability to leave reviews and ratings for products they have purchased. This will help other users make informed decisions and provide valuable feedback to the platform administrators.

7. Admin Dashboard: Administrators will have a secure and intuitive dashboard where they can manage various aspects of the e-commerce platform. This includes managing user accounts, adding and editing products, monitoring sales and revenue, and generating reports.

8. Responsive Design: The e-commerce platform will be designed to be responsive and compatible with different devices, including desktops, tablets, and mobile phones. This will ensure a seamless user experience across multiple platforms.

9. Security and Performance: The project will prioritize implementing robust security measures to protect user data, including encryption and secure authentication mechanisms. Performance optimization techniques, such as caching and database indexing, will be employed to ensure fast and efficient loading times.

10. Scalability and Extensibility: The project will be developed with scalability and extensibility in mind, allowing for future enhancements and integration with third-party services, such as analytics tools, marketing platforms, or shipping providers.

In summary, the scope of the e-commerce MERN project encompasses all the essential features required to create a fully functional online marketplace. By implementing user management, a comprehensive product catalogue, shopping cart and checkout functionality, order and inventory management systems, user reviews and ratings, an admin dashboard, responsive design, security measures, and scalability options, the project aims to deliver a robust and user-friendly e-commerce platform that meets the needs of both customers and administrators.

They can analyze customer buying patterns and preferences and offer tailor-made offers,discounts, and services.

Figure 1.1 To Show how many people shop online

1.4 Project Objective

The project objective is to deliver the online shopping application. The objective of this project is to develop a general purpose e-commerce store where books from any field can be bought from the comfort of home through the Internet.

An online store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as credit card number. An e- mail notification is sent to the customer as soon as the order is placed.

This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products from anywhere through internet by using a web site. Thus the customer will get the service of online shopping and home delivery from this shop.

Since the application is available online through its website so it is easily accessible and always available.

Sellers can increase and widen their reach to way beyond their cities – they can get customers from literally anywhere in the world, provided they are willing to ship.

• Even small businesses can increase their sales and grow by selling online

• They can enjoy massive savings in infrastructure, as they need not rent or purchase space in pricey locations or spend on interiors, displays.

• As online stores can be operated with minimal staff, there are huge savings in salaries; sellers can also save on overheads like electricity and other utility bills.

• Online storefronts are open 24/7 to serve customers – no more worrying about missing out because of holidays, strikes, or even lockdowns.

• They can respond quickly to market demands

• Sellers can deal in a wide range of products

1.5 Hardware / Software Requirements

Following are some of the hardware and software requirements which are necessary for the project to run -:

- Windows XP, Windows 7 (32/64 bit) or higher
- Minimum 4 GB RAM and higher
- 10 GB available space on the hard disk
- At least one Internet Browser e.g. Chrome, Firefox, Microsoft Edge etc.
- Node.js to be installed
- Active internet connection minimum speed 512kbps and above.
- At least one installed code Editor to test and debug your code e.g.
- Visual studio code

1.6 Methodology

Methodology is a body of methods, rules, and postulates employed by a discipline or a particular procedure or set of procedures.

Sharing your methodology gives legitimacy to your research. An unreliable or erroneous methodology produces unreliable or erroneous results. The reader of your research expects you to have followed accepted practices so that the conclusions you reach are valid. The methodology you report needs to be repeatable, meaning anyone who uses the methods you write about should reach the same conclusions you reached.

The project provides users to login into the website by entering their name, email, and other details and password links. The account verification link will be sent to that email address, after clicking the link the account will get activated and the user can log in to their account and sees the products and order the items. The selected items are stored in the shopping cart where he /she sees all the products that he/ she selects to order.

For ordering the product the user's address is then given and user selects the payment method to pay the price of that product. If the user selects by debit card/credit card then the customer must enter their card details for doing the online payment functionality with the help of payment gateway.

A shopping cart is one of the important facilities provided in online shopping, this lets customers browse different goods and services, and once they select an item to purchase they can place the item in the shopping cart, and continue browsing till the final selection. Customers can even remove the items from the shopping cart that were selected earlier before they place the final order. It reminds us of the shopping basket that we carry in a departmental store.

1.7 Module Description

In this project, we give many modules that's make the website user friendly and easy to use the project following are key modules -:

1. Create Account or Sign Up Module: In this module, the user enters his /her name, email id, and password and other details necessary for account creation such as address, contact number and pin code and then clicks the continue button then the customer gets an account verification link in his email account and after confirmation the user can login with their credentials.

2. Login Module: In this module, after the creation of the account the user login with the help of Email id and password, if a customer forgot the password then he/she can reset the password with the forgot the password button provided in the login module.

3. Home Page: After successful login the home page is opened and on the home page user can see different products, search the products and if he wants to buy a product he/she can adds the product to the cart.

4. Add to Cart: After selecting and adding the products which the user needs to buy. He or she can open add to Cart page where the customer see all products which he/she wants to buy. In this page, users can also remove the products

5. Buy Module: On the Buy Page user can buy the product and select the payment method such as cash or card or UPI, if the user chooses by card method then the customer needs to enter his or her card details

CHAPTER 2

FEASIBILITY STUDY

2.1 Introduction

A feasibility study in a project is an assessment conducted to determine the practicality and viability of a proposed project. It involves analyzing various factors to determine whether the project is feasible in terms of technical, economic, operational, legal, and scheduling aspects.

During a feasibility study, the project team evaluates the project's objectives, requirements, and constraints. They assess the technical feasibility by considering if the necessary technology and resources are available or can be developed within the project's constraints. Economic feasibility involves analyzing the project's financial viability, including costs, benefits, return on investment, and potential risks.

Operational feasibility assesses whether the project can be implemented smoothly within the existing operational environment and if it aligns with the organization's goals and strategies. Legal feasibility involves analyzing any legal or regulatory requirements, permits, licenses, or intellectual property considerations that may impact the project.

Additionally, scheduling feasibility examines whether the project can be completed within the desired timeframe, considering available resources and potential constraints. The feasibility study helps stakeholders make informed decisions about whether to

proceed with the project, modify certain aspects, or abandon it altogether based on the identified risks and potential returns.

Ultimately, a feasibility study provides valuable insights into the viability and potential successes of a project, helping stakeholders make informed decisions and allocate resources effectively.

2.2 Technical Feasibility

The technical feasibility of an e-commerce MERN (MongoDB, Express, React, Node.js) project refers to the assessment of whether the required technology, infrastructure, and resources are available or can be developed to support the successful implementation of the project. It involves evaluating the technical aspects of building and maintaining the e-commerce platform.

To determine the technical feasibility of the project, several factors need to be considered:

1. Technology Stack: The MERN stack, consisting of MongoDB, Express, React, and Node.js, is a popular choice for developing web applications. The availability of skilled developers and community support for these technologies ensures that the required technical expertise is readily available.

2. Scalability: The e-commerce platform should be designed to handle a large number of users, products, and transactions. The technical infrastructure, such as the hosting environment, database systems, and caching mechanisms, should be capable of scaling up to meet increasing demands.

3. Integration Capabilities: The project may require integrating with third-party services, such as payment gateways, shipping providers, or analytics tools. Assessing the availability of APIs, documentation, and compatibility with the chosen technology stack is crucial to ensure seamless integration.

4. Security Measures: E-commerce platforms handle sensitive customer data, including personal information and payment details. The technical feasibility assessment should include evaluating the implementation of robust security measures, such as encryption, secure authentication, and protection against common vulnerabilities.

5. Performance Optimization: The platform should be optimized for performance to provide a smooth and responsive user experience. This includes techniques like

caching, database indexing, code optimization, and server-side rendering to minimize page load times and ensure efficient resource utilization.

6. Infrastructure and Hosting: The technical feasibility analysis should consider the infrastructure requirements, such as server capacity, bandwidth, and storage. Assessing the availability of hosting options, cloud services, and infrastructure management tools is crucial for reliable and scalable hosting.

By conducting a thorough technical feasibility study, the project team can identify any potential technical challenges, assess the availability of required resources and expertise, and ensure that the chosen technology stack and infrastructure can support the development and operation of the e-commerce platform effectively.

2.3 Operational Feasibility

The operational feasibility of an e-commerce MERN (MongoDB, Express, React, Node.js) project refers to the assessment of whether the proposed project can be implemented and integrated smoothly within the existing operational environment. It involves analyzing various factors related to the organization, its processes, and resources to determine if the project aligns with the operational goals and strategies.

To evaluate the operational feasibility of the e-commerce MERN project, several considerations need to be addressed:

1. Organizational Alignment: The project team needs to ensure that the e-commerce platform aligns with the organizations overall goals, strategies, and business model. It should complement existing operations and contribute to the organization's growth and profitability.

2. Process Integration: The project team must assess how the e-commerce platform will integrate with existing business processes. This includes inventory management, order fulfillment, customer support, and other operational workflows. The platform should enhance efficiency and streamline processes rather than disrupt them.

3. Resource Availability: Adequate resources, including human resources and infrastructure, should be available to support the successful implementation and ongoing operation of the e-commerce platform. This involves assessing staffing requirements, skill sets, training needs, and potential resource constraints.

4. Change Management: Implementing an e-commerce platform may require changes to the organization's structure, roles, and responsibilities. The operational feasibility analysis should consider the organization's ability to manage and adapt to these changes, ensuring proper change management strategies are in place.

5. User Adoption: The success of an e-commerce platform depends on user adoption. The project team should assess the readiness of users, such as employees,

customers, and other stakeholders, to embrace the new platform. Training, user support, and user experience design should be considered to facilitate user acceptance.

6. Legal and Regulatory Compliance: The e-commerce platform must adhere to legal and regulatory requirements, including data privacy, consumer protection, and payment regulations. The project team should assess the feasibility of meeting these compliance requirements and ensure necessary measures are implemented.

By conducting an operational feasibility study, the project team can identify any potential operational challenges, evaluate resource availability, assess the impact on existing processes, and ensure that the proposed e-commerce platform can be seamlessly integrated into the organization's operations. This analysis helps in making informed decisions and planning for a successful implementation of the project.

2.4 Economical Feasibility

The economic feasibility of an e-commerce MERN (MongoDB, Express, React, Node.js) project involves assessing the financial viability and potential economic benefits of implementing the project. It aims to determine whether the project is financially feasible and whether the anticipated benefits outweigh the costs associated with development, operation, and maintenance.

To evaluate the economic feasibility of the e-commerce MERN project, the following aspects need to be considered:

1. Cost-Benefit Analysis: A comprehensive cost-benefit analysis is conducted to determine the financial implications of the project. This includes assessing development costs, infrastructure costs, ongoing maintenance expenses, and potential revenue generation. The benefits derived from increased sales, expanded customer reach, and cost savings are weighed against the investments required.

2. Return on Investment (ROI): The project team calculates the expected ROI by comparing the projected financial gains with the initial investment. This helps stakeholders understand the profitability and financial viability of the e-commerce platform. Factors such as sales projections, profit margins, and cost reduction opportunities are considered to determine the ROI.

3. Revenue Generation: The e-commerce platform presents opportunities for revenue generation through increased sales and customer acquisition. The economic feasibility analysis evaluates the potential for generating revenue through product sales, advertising, commissions, and subscription models. The projected revenue should be realistic and based on market research and industry trends.

4. Cost Reduction and Efficiency: An e-commerce platform can offer cost reduction opportunities by streamlining processes, reducing manual interventions, and automating tasks. The analysis considers potential cost savings in areas such as inventory management, order processing, customer support, and marketing.

5. Market Analysis: The economic feasibility assessment includes analyzing the market demand, competition, and growth potential. Understanding the target market's size, purchasing behavior, and trends helps determine the revenue potential and market penetration opportunities for the e-commerce platform.

6. Risk Assessment: The economic feasibility analysis identifies potential risks and uncertainties associated with the project's financial aspects. This includes market risks, technological risks, regulatory risks, and financial risks. Strategies to mitigate these risks and alternative scenarios are considered in the evaluation.

By conducting an economical feasibility study, the project team can assess the financial viability of the e-commerce MERN project. This analysis enables stakeholders to make informed decisions regarding resource allocation, investment commitments, pricing strategies, and revenue projections. It helps ensure that the project aligns with the organization's financial goals and contributes to its long-term sustainability and profitability.

2.5 Behavioral Feasibility

The behavioral feasibility of an e-commerce project developed using the MERN (MongoDB, Express.js, React.js, and Node.js) stack is highly viable and has proven to be successful in the industry. The MERN stack provides a robust and scalable framework for building modern web applications, including e-commerce platforms.

One of the key advantages of using the MERN stack for an e-commerce project is its ability to handle complex user interactions and dynamic content. The React.js library, in combination with Node.js and Express.js, allows for the creation of highly responsive and interactive user interfaces. This is crucial for e-commerce platforms, as they require seamless navigation, product browsing, shopping cart management, and checkout processes. The MERN stack's component-based architecture promotes code reusability and makes it easier to develop and maintain complex user interfaces.

Additionally, the MERN stack is known for its scalability, which is essential for e-commerce projects that anticipate high traffic and the need to handle large volumes of data. MongoDB, a NoSQL database used in the MERN stack, offers flexibility in data modeling and can easily accommodate changing business requirements. This allows e-commerce platforms to store and manage product catalogs, user profiles, order histories, and other relevant data efficiently.

Furthermore, the MERN stack supports the development of real-time features, such as inventory management, live chat support, and order tracking. These functionalities enhance the user experience and provide essential tools for e-commerce businesses to operate effectively.

From a behavioral perspective, users have become accustomed to the seamless and intuitive experiences offered by modern e-commerce platforms. By leveraging the MERN stack, developers can create user-friendly interfaces that meet these expectations. The stack's robustness and performance ensure a smooth browsing and purchasing experience, reducing user frustration and increasing the likelihood of repeat visits and customer satisfaction.

In conclusion, the behavioral feasibility of an e-commerce project developed using the MERN stack is high. Its ability to handle complex user interactions, scalability, support for real-time features, and the familiarity of users with modern e-commerce platforms make it a suitable choice for building successful and user-centric e-commerce applications.

CHAPTER 3

DATABASE DESIGN

3.1 Introduction

Database design in a MERN (MongoDB, Express, React, Node.js) project refers to the process of structuring and organizing the database schema to efficiently store and manage data for the application. It involves designing the database schema, defining relationships between entities, and optimizing data retrieval and manipulation.

In a MERN project, MongoDB is commonly used as the database, which is a NoSQL document database. The following aspects are essential in the database design process:

1. Schema Design: The schema design involves determining the structure and organization of data within the MongoDB collections. It includes identifying the entities, their attributes, and the relationships between them. The schema design should reflect the application's requirements and support efficient data access and manipulation.

2. Data Modeling: Data modeling in MongoDB involves creating models for each entity in the application. This includes defining the fields, data types, and indexes for efficient querying. Proper data modeling ensures data integrity, performance, and scalability.

3. Relationship Mapping: While MongoDB is a NoSQL database, it still allows for defining relationships between entities. In the database design, relationships between entities need to be identified and implemented using techniques like embedding, referencing, or a combination of both. This ensures consistency and facilitates data retrieval.

4. Indexing: Indexes improve the performance of database queries by enabling faster data retrieval. During the database design phase, appropriate indexes should be created on frequently queried fields to optimize query execution and improve application responsiveness.

5. Data Validation: MongoDB allows for data validation at the database level, ensuring data integrity and consistency. Database design should include defining validation rules and constraints to enforce data validation, such as field types, required fields, and unique values.

6. Scalability and Performance: The database design should consider scalability and performance requirements. Techniques like sharding and replication can be used to distribute data across multiple servers and ensure high availability and performance as the application scales.

Overall, database design in a MERN project plays a crucial role in ensuring efficient data storage, retrieval, and manipulation. It involves designing the database schema, modeling entities and relationships, optimizing data access through indexing, ensuring data integrity through validation, and considering scalability and performance requirements. A well-designed database enhances the overall performance and user experience of the MERN application.

3.2 Database Tables

In MongoDB, the term "tables" is replaced with "collections." Collections are the primary containers for storing and organizing data in a MongoDB database. Unlike traditional relational databases, MongoDB is a NoSQL document-oriented database, where data is stored as flexible and schema-less JSON-like documents.

Collections in MongoDB can be thought of as analogous to tables in a relational database, but with some fundamental differences:

1. Document Structure: Instead of rows and columns, collections store documents. Each document is a self-contained data entity represented as a JSON-like structure. Documents within a collection can have varying structures and fields, offering flexibility in data modeling.

2. Schema Flexibility: MongoDB does not enforce a strict schema like traditional databases. Within a collection, documents can have different fields and structures, allowing for easy and dynamic modifications to the data model without the need for explicit schema migrations.

3. Document Indexing: MongoDB provides powerful indexing capabilities to optimize data retrieval. Indexes can be created on specific fields within a collection to improve query performance. Indexes allow for faster searching, sorting, and filtering of data.

4. Embedded Documents and Arrays: MongoDB supports nesting documents and arrays within a document. This enables the modeling of complex and hierarchical data structures. Embedded documents can be used to represent one-to-one or one-to-many relationships between data entities.

5. CRUD Operations: MongoDB supports CRUD operations (Create, Read, Update, and Delete) for working with collections. Documents can be inserted, queried, updated, and deleted using MongoDB's query language and APIs.

Collections in MongoDB are designed to provide high performance and scalability. They can handle large amounts of data and can be distributed across multiple servers using MongoDB's built-in sharing and replication features.

Overall, MongoDB's collections provide a flexible and scalable approach to organizing and storing data, allowing for efficient data retrieval and manipulation in a NoSQL document-oriented database system.

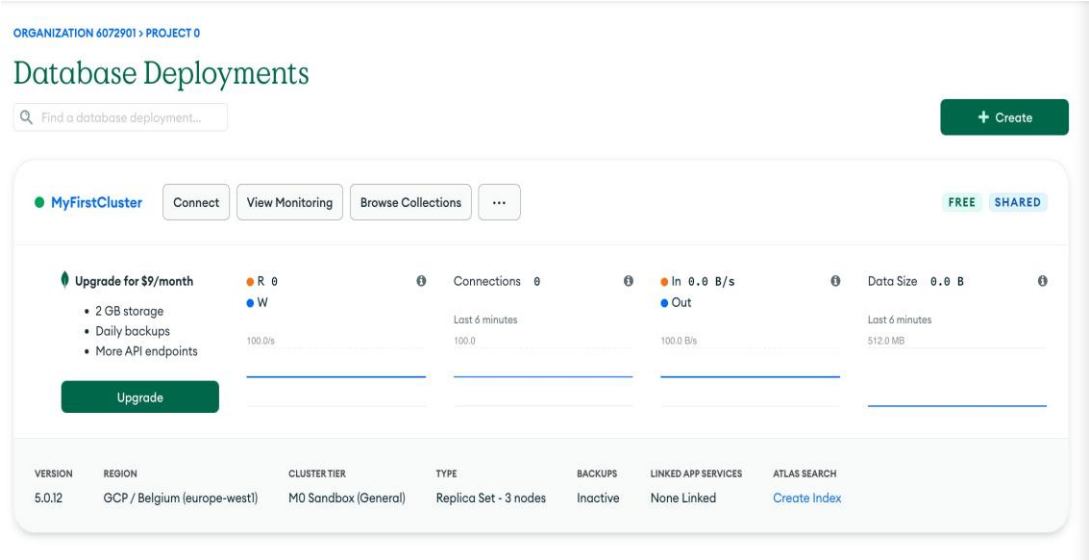


Figure 3.1 MongoDB Database

3.3 Flow Chart

A flowchart is a graphical representation or diagram that depicts the sequential flow of activities or steps in a project, process, or system. It uses various shapes, symbols, and arrows to illustrate the order and dependencies of tasks or decisions.

In a project, a flowchart serves as a visual tool to communicate and understand the workflow or logic of the project. It helps in analyzing and designing the project's structure, identifying potential bottlenecks, and facilitating efficient project management.

The key components of a flowchart include:

1. Start and End Points: These symbols indicate the beginning and conclusion of the flowchart, representing the starting and ending tasks or activities in the project.

2. Process/Task Symbols: These symbols represent the actions or activities that are performed as part of the project. Each process symbol denotes a specific task or action that needs to be executed.

3. Decision Symbols: Decision symbols are used to represent points where a choice or decision needs to be made. They typically involve yes/no or true/false questions and direct the flow of the project based on the decision outcome.

4. Connectors/Arrows: Arrows or connectors depict the flow and direction between different symbols and indicate the logical sequence of tasks or decisions.

By visualizing the project's flow through a flowchart, project managers and team members can gain a better understanding of the project's structure, dependencies, and potential bottlenecks. It helps in identifying areas for optimization, streamlining processes, and ensuring a clear and logical progression of tasks.

Flowcharts also aid in communication and documentation. They provide a visual reference that can be easily shared and understood by stakeholders, allowing for effective collaboration, decision-making, and troubleshooting.

Overall, flowcharts are valuable tools in project management as they provide a clear and visual representation of the project's workflow, facilitating analysis, planning, and effective execution of tasks.

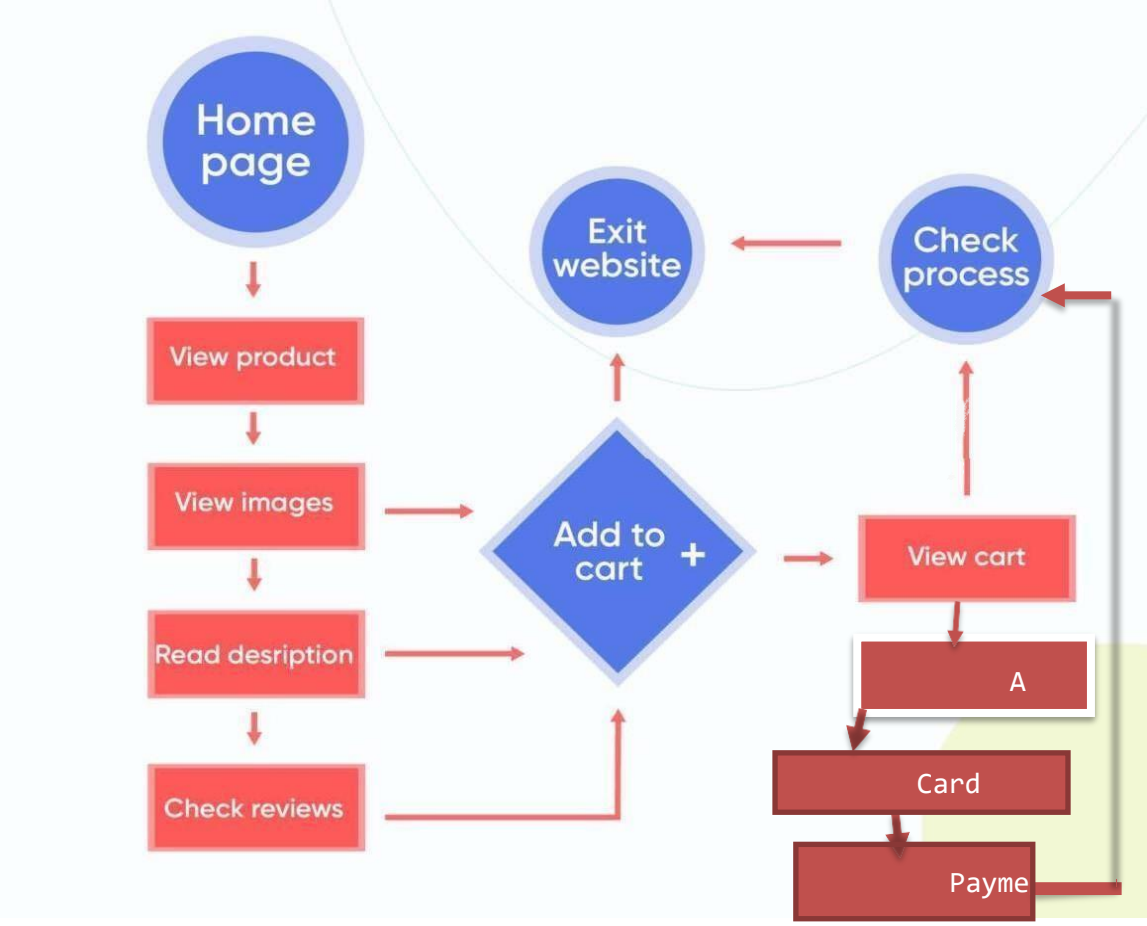


Figure 3.2 Flowchart

3.4 Use Case Diagram

A use case diagram is a visual representation of the functional requirements of a system or project from the perspective of its users. It showcases the interactions between users (referred to as actors) and the system, depicting various use cases or functionalities the system provides. Use case diagrams are widely used in software development and project management to understand and communicate system behavior and requirements.

The key components of a use case diagram include:

1. Actors: Actors represent the different types of users or external systems interacting with the project. They can be individuals, roles, or other systems that interact with the project to achieve specific goals.

2. Use Cases: Use cases represent specific functionalities or actions that the system provides to its users. They capture the system's behavior and define the interactions between actors and the system. Each use case represents a specific user goal or a complete business process.

3. Relationships: Relationships between actors and use cases are depicted using lines or connectors. The relationships can be associations, indicating that an actor interacts with a particular use case, or dependencies, indicating that a use case relies on another use case.

4. System Boundary: The system boundary is a box that encapsulates all the use cases and actors. It defines the scope of the system being represented by the use case diagram.

Use case diagrams offer several benefits in project management:

1. Requirements Analysis: Use case diagrams help in identifying and analyzing the functional requirements of a project. They provide a high-level overview of the

system's behavior and allow stakeholders to understand the system's scope and functionalities.

2. Communication and Collaboration: Use case diagrams serve as a communication tool between stakeholders, project managers, developers, and designers. They provide a visual representation of the project's functionality, facilitating discussions, clarifying requirements, and ensuring a shared understanding among team members.

3. System Design: Use case diagrams provide insights into the system's design and structure. They assist in identifying major modules or components and their interactions, helping in the creation of a well-organized system architecture.

4. Test Planning: Use case diagrams serve as a basis for test planning and test case creation. Each use case can be mapped to specific test scenarios, ensuring comprehensive test coverage.

Overall, use case diagrams are valuable tools in project management as they help in capturing and communicating system requirements, analyzing system behavior, guiding system design, and facilitating effective collaboration among stakeholders.

Following are the use case diagram -:

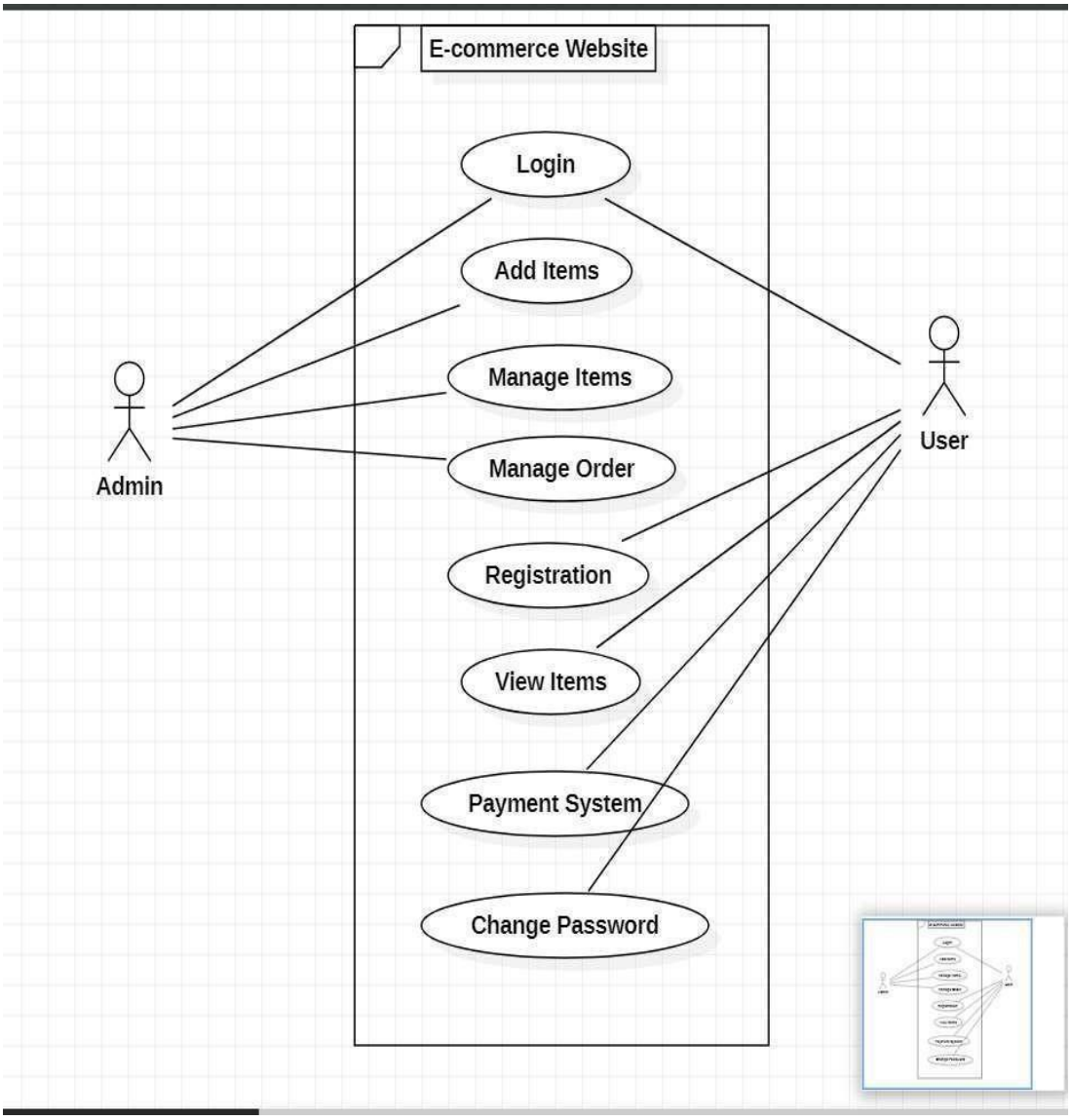


Figure 3.3 Ecommerce Use Case Diagram

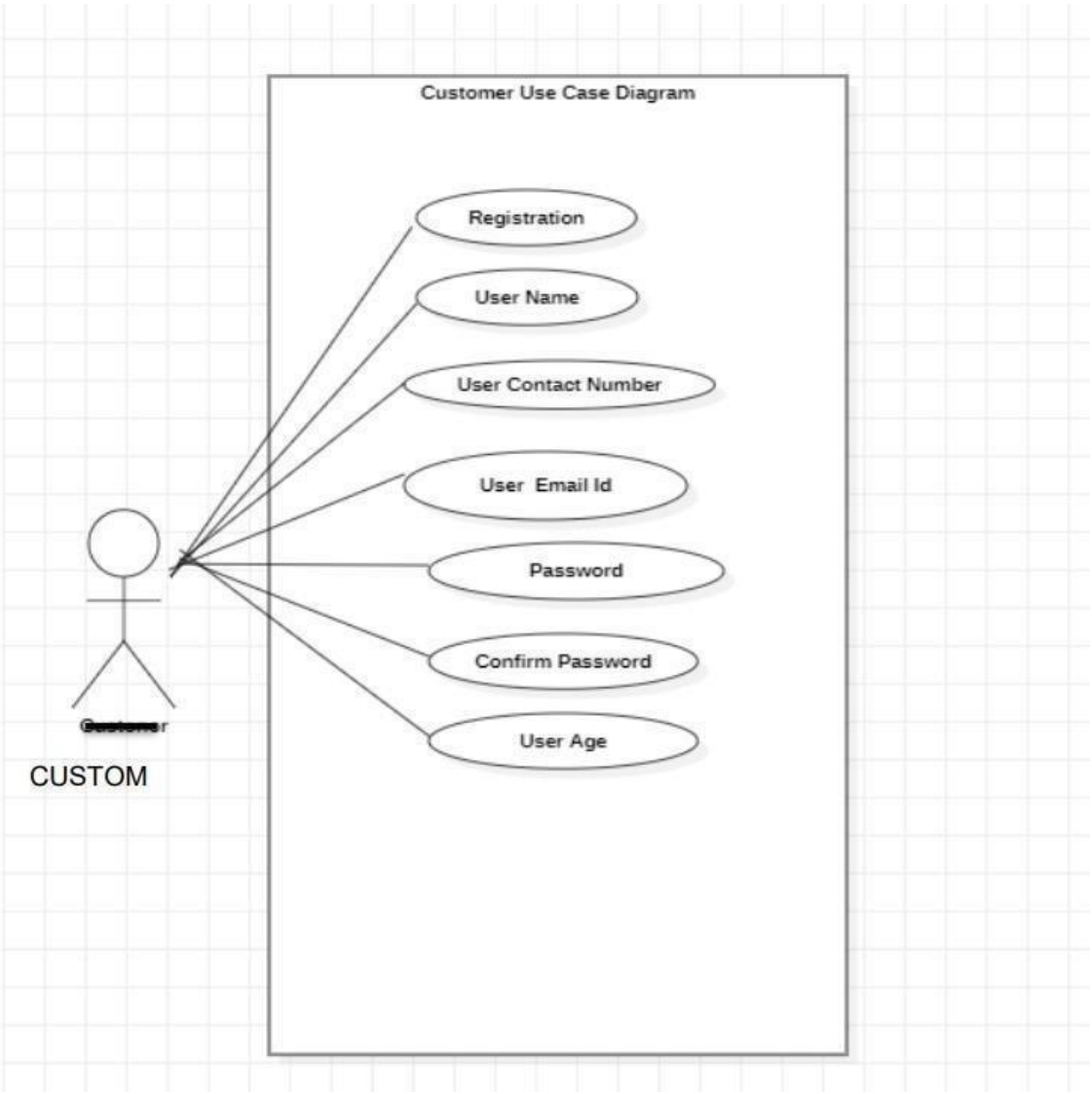


Figure 3.4 Customer Use Case Diagram

3.5 Sequence Diagram

A sequence diagram is a type of interaction diagram in UML (Unified Modeling Language) that represents the sequence of interactions between objects or components within a system. It illustrates the flow of messages exchanged between objects and the order in which these messages are sent and received. Sequence diagrams are widely used in software development and project management to visualize the dynamic behavior of a system and understand how different components collaborate to accomplish specific tasks.

The key elements of a sequence diagram include:

1. Objects: Objects represent the instances of classes or components participating in the interactions. Each object is depicted as a rectangle with its name on the top.

2. Lifelines: Lifelines represent the lifespan of an object during the sequence of interactions. They are vertical lines that extend from the object's rectangle, indicating the time period in which the object exists.

3. Messages: Messages depict the communication between objects in the system. They are represented by arrows with labels indicating the type of message (e.g., method calls, signals, or events) and the parameters being passed. Messages can be synchronous (blocking), asynchronous (non-blocking), or self-referencing.

4. Activation: Activation bars represent the period of time when an object is executing a particular message or operation. They are shown as vertical rectangles on the lifeline of the object, indicating the duration of the object's activity.

5. Return Messages: Return messages indicate the response sent by an object after processing a received message. They help depict the flow of control between objects and show the order of method calls and their corresponding return values.

Sequence diagrams offer several benefits in project management:

1. Behavior Visualization: Sequence diagrams provide a visual representation of the dynamic behavior of a system, making it easier to understand how components interact and collaborate to achieve specific tasks. They help in identifying the sequence of operations and message flows, leading to a better understanding of the system's behavior.

2. Requirement Validation: Sequence diagrams can be used to validate and refine system requirements. By visualizing the sequence of interactions, stakeholders can identify missing or incomplete requirements, detect potential design flaws, and refine the system's behavior before development.

3. Communication and Collaboration: Sequence diagrams act as a communication tool between stakeholders, developers, and designers. They facilitate effective collaboration by providing a shared understanding of the system's interactions and behavior, ensuring that everyone is on the same page.

4. Error Detection and Troubleshooting: Sequence diagrams can assist in identifying potential errors or bottlenecks in the system's behavior. By visualizing the message flows, developers can identify areas where messages are not delivered or processed correctly, aiding in troubleshooting and debugging.

5. Documentation: Sequence diagrams serve as documentation artifacts, capturing the dynamic behavior of the system. They can be referenced during development, testing, and maintenance phases to ensure consistent understanding and implementation of the system's interactions.

Overall, sequence diagrams are valuable tools in project management as they help in visualizing and understanding the dynamic behavior of a system, validating requirements, facilitating communication and collaboration, aiding in error detection, and serving as documentation artifacts.

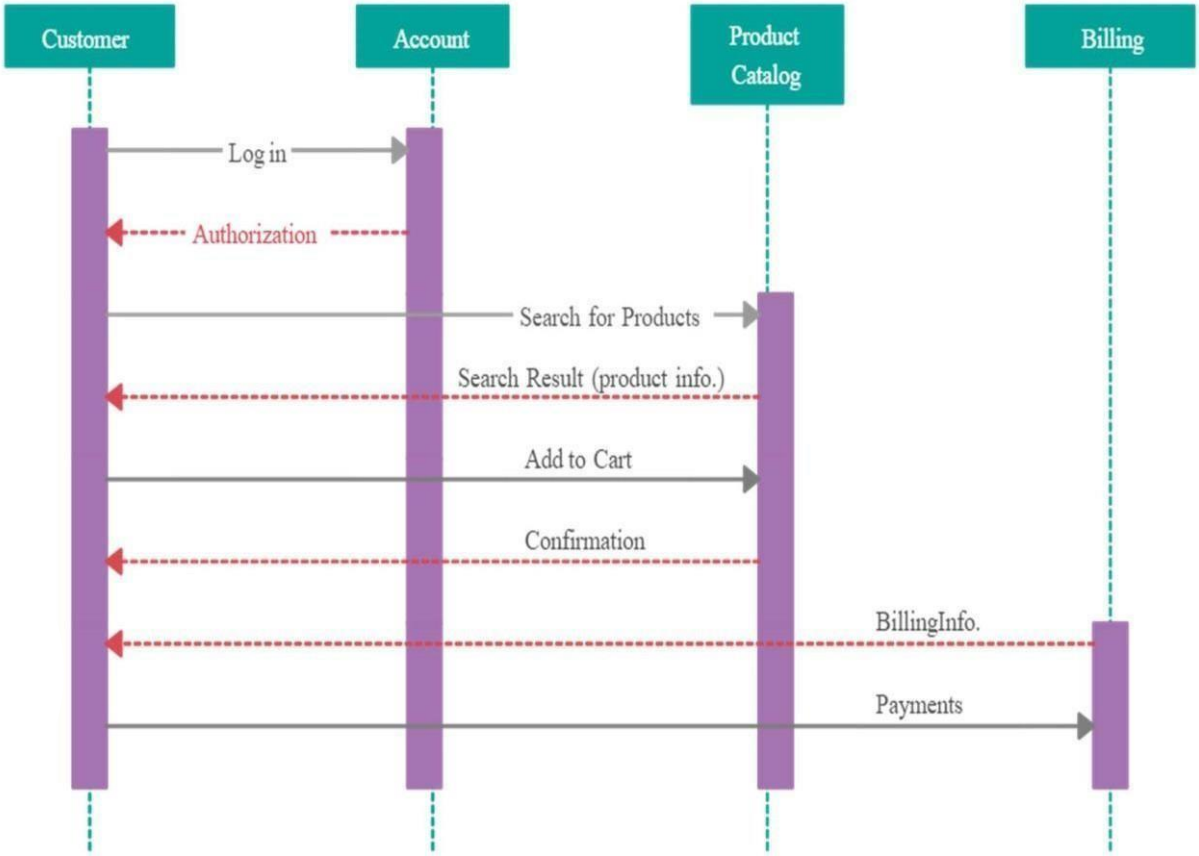


Figure 3.5 Sequence Diagram

3.6 Activity Diagram

An activity diagram is a type of behavior diagram in UML (Unified Modeling Language) that represents the flow of activities and actions within a system or process. It provides a visual representation of the sequence of activities, decisions, and control flows involved in achieving a specific goal or completing a task. Activity diagrams are widely used in project management to model and analyze business processes, workflows, and system behavior.

The key components of an activity diagram include:

1. Initial Node: The initial node represents the starting point of the activity diagram. It indicates where the process or activity begins.

2. Actions: Actions represent individual tasks or actions performed within the system or process. They are depicted as rounded rectangles and describe the specific activities that need to be executed.

3. Control Flow: Control flows are represented by arrows and show the sequence of activities and decisions within the diagram. They illustrate the order in which activities are executed, branching based on decisions, and merging of multiple flows.

4. Decision Node: Decision nodes are diamond-shaped symbols that represent points in the process where a decision needs to be made. They have multiple outgoing flows based on different conditions or criteria.

5. Fork and Join Nodes: Fork nodes represent points in the process where multiple parallel activities can be performed simultaneously. Join nodes represent points where the parallel activities merge back into a single flow.

6. Final Node: The final node represents the endpoint of the activity diagram, indicating the completion of the process or activity.

Activity diagrams offer several benefits in project management:

1. Process Modeling: Activity diagrams help in modeling and understanding complex business processes and workflows. They provide a clear visualization of the sequence of activities and the dependencies between them, facilitating process analysis and optimization.

2. Requirement Analysis: Activity diagrams aid in analyzing system requirements by visually representing the flow of activities. They help stakeholders and project teams identify and clarify the necessary steps and actions required to achieve a specific goal.

3. Communication and Collaboration: Activity diagrams act as a communication tool, enabling stakeholders, project managers, and development teams to have a shared understanding of the system's behavior. They help in effective collaboration, discussions, and decision-making.

4. Validation and Testing: Activity diagrams can be used to validate and verify system behavior. They assist in designing test cases and scenarios to ensure comprehensive testing of the system's functionality and logic.

5. Documentation: Activity diagrams serve as documentation artifacts that capture the process flow and system behavior. They provide a visual reference that can be utilized during system development, maintenance, and training.

Overall, activity diagrams are valuable tools in project management as they help in modeling and understanding business processes, analyzing requirements, facilitating communication and collaboration, aiding in validation and testing, and serving as documentation artifacts.

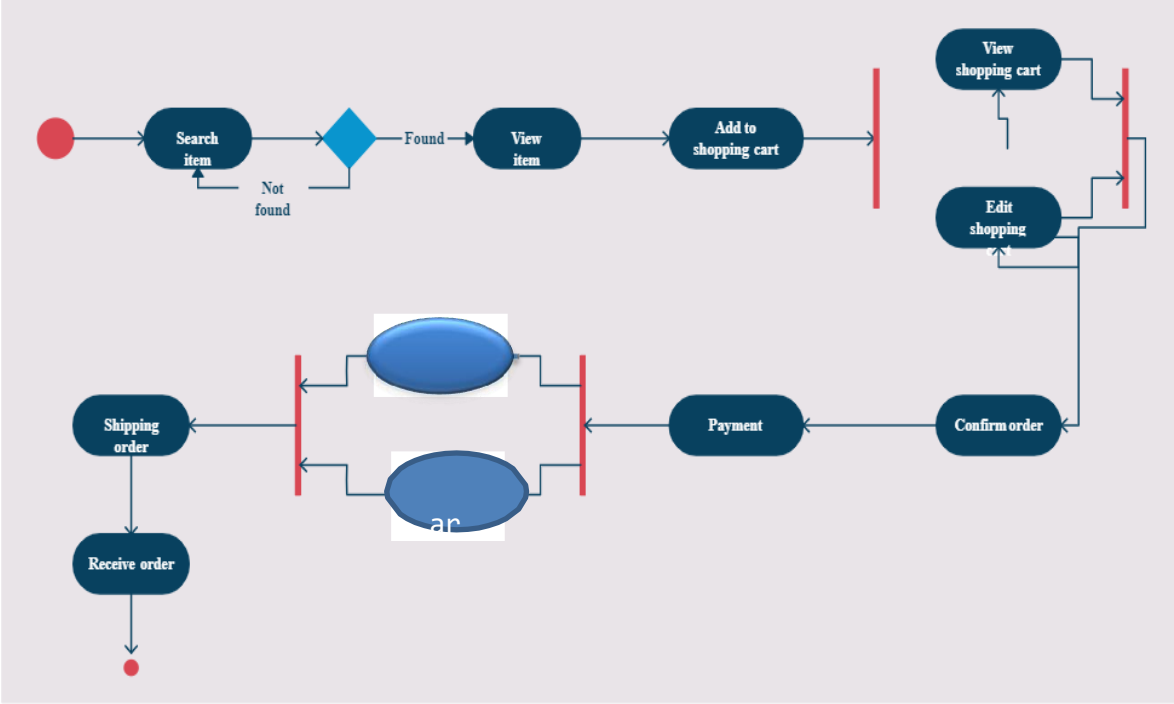


Figure 3.6 Activity Diagram

3.7 Collaborative Diagram

A collaboration diagram, also known as a communication diagram, is a type of UML (Unified Modeling Language) diagram that depicts the interactions and relationships between objects or components in a system. It focuses on how objects collaborate and communicate with each other to accomplish specific tasks or achieve a common goal.

In a collaboration diagram, the emphasis is on the structural organization of objects and the messages exchanged between them. The diagram represents the objects as rectangular shapes, with lines and arrows indicating the messages being passed between the objects. The messages can be synchronous (blocking) or asynchronous (non-blocking), and they demonstrate the flow of control or information during the system's execution.

The key elements of a collaboration diagram include:

1. Objects: Objects are represented as rectangles, depicting the instances of classes or components involved in the system. Each object is labeled with its name and may include additional attributes or properties.

2. Messages: Messages are represented by arrows connecting the objects. They indicate the communication or interaction between objects. The messages can be labeled to specify the purpose or content of the communication.

3. Roles: Roles are depicted as smaller rectangles attached to the objects, indicating the specific responsibilities or functions performed by the objects within the collaboration.

4. Associations: Associations represent the relationships between objects, showing the connections or dependencies between them. They can be labeled to provide additional information about the relationship.

Collaboration diagrams provide a visual representation of how objects collaborate and communicate in a system, helping to understand the overall system structure and the interactions between its components. They facilitate communication, design, and implementation by illustrating the flow of messages and the dependencies between objects. Collaboration diagrams are particularly useful in analyzing system behavior, identifying communication patterns, and ensuring that the system's design promotes effective collaboration among its components.

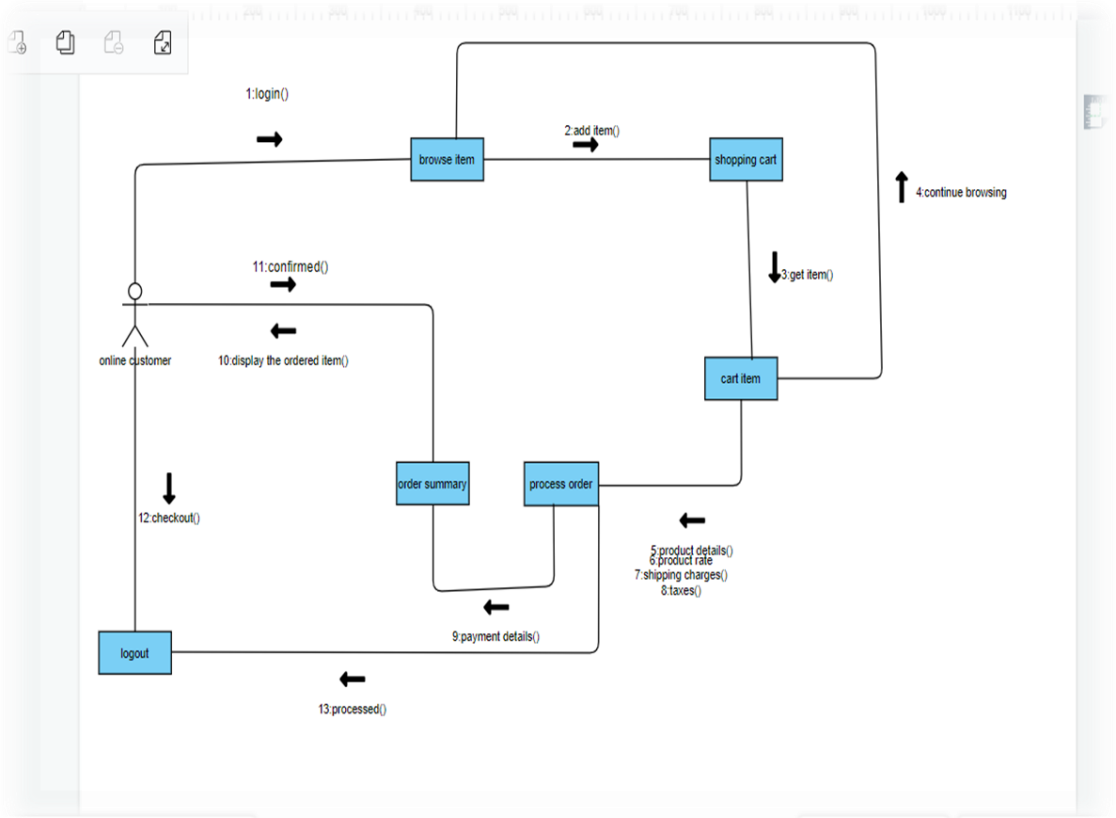


Figure 3.7 Collaboration Diagram

CHAPTER 4

FORM DESIGN

4.1 Input / Output Form (Screen Shot)

Register Screen

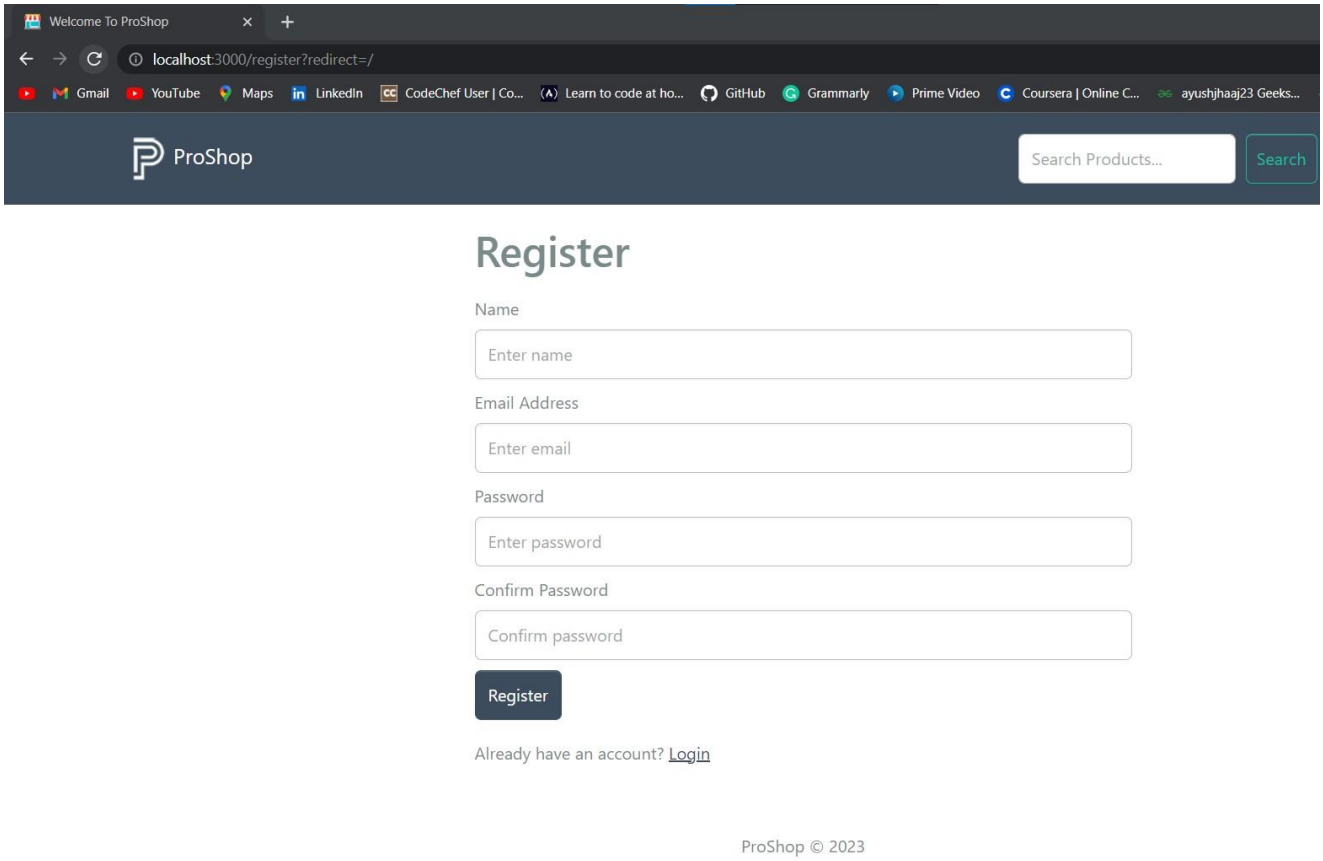


Figure 4.1 Register Screen

SignIn Screen

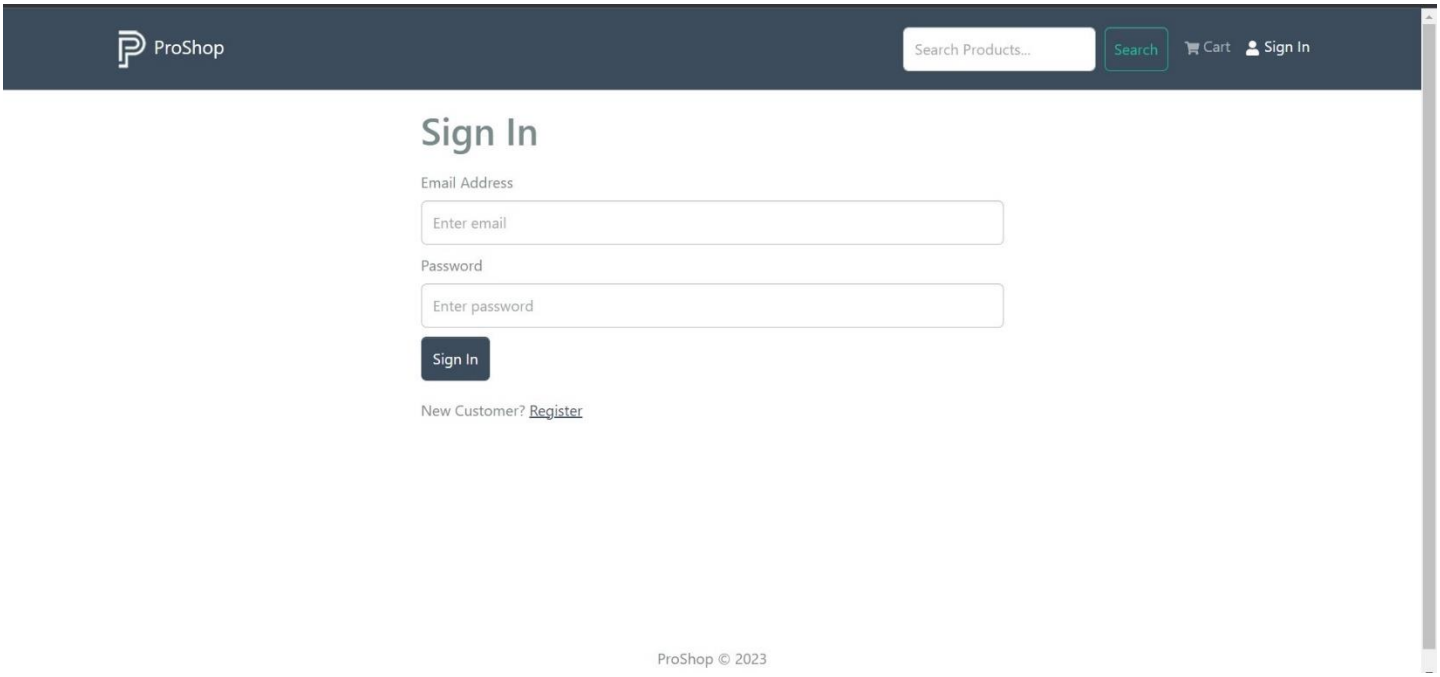


Figure 4.2 SignIn Screen

Home Screen

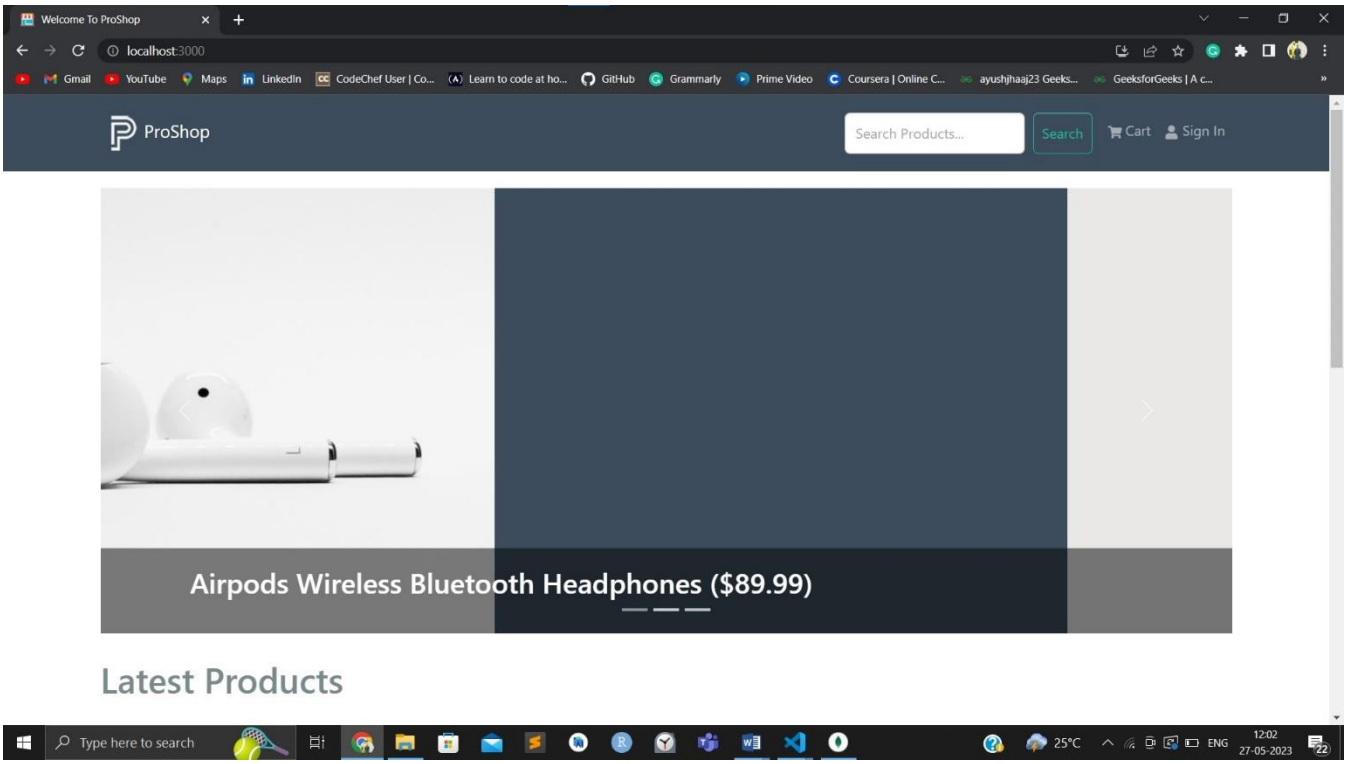


Figure 4.3 Home Screen

User Profile

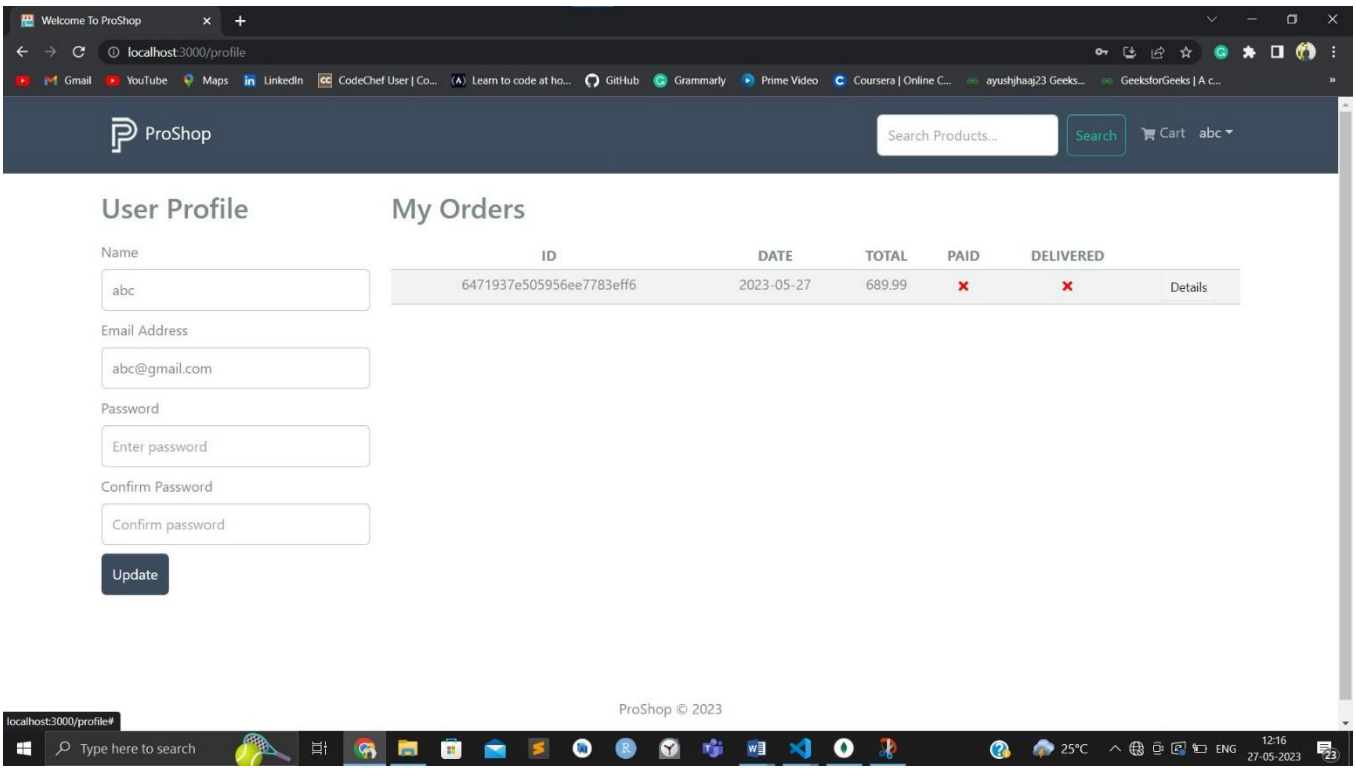


Figure 4.4 User Profile

Following are the step to buy a product -:

Step 1:

Search your desired product

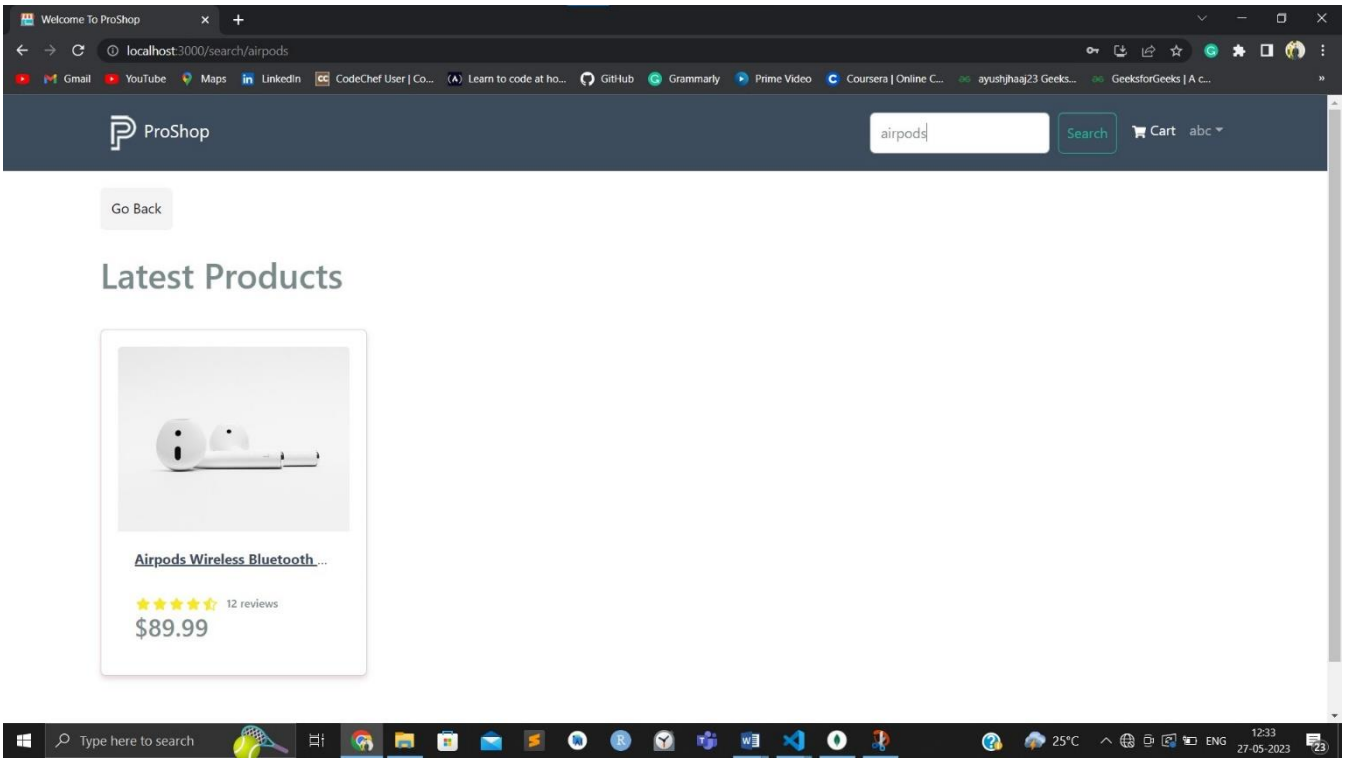


Figure 4.5 Search your product

Step 2:

See your product details

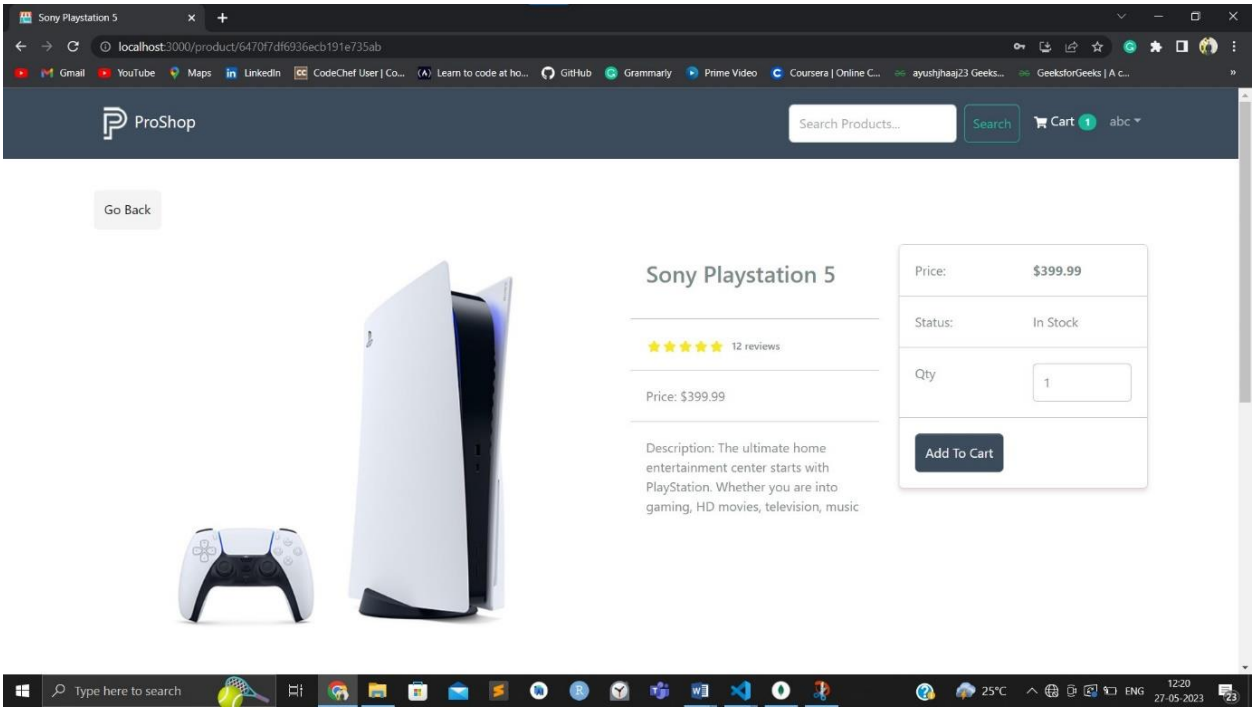


Figure 4.6 See product details

Step 3:

Add product to the Cart

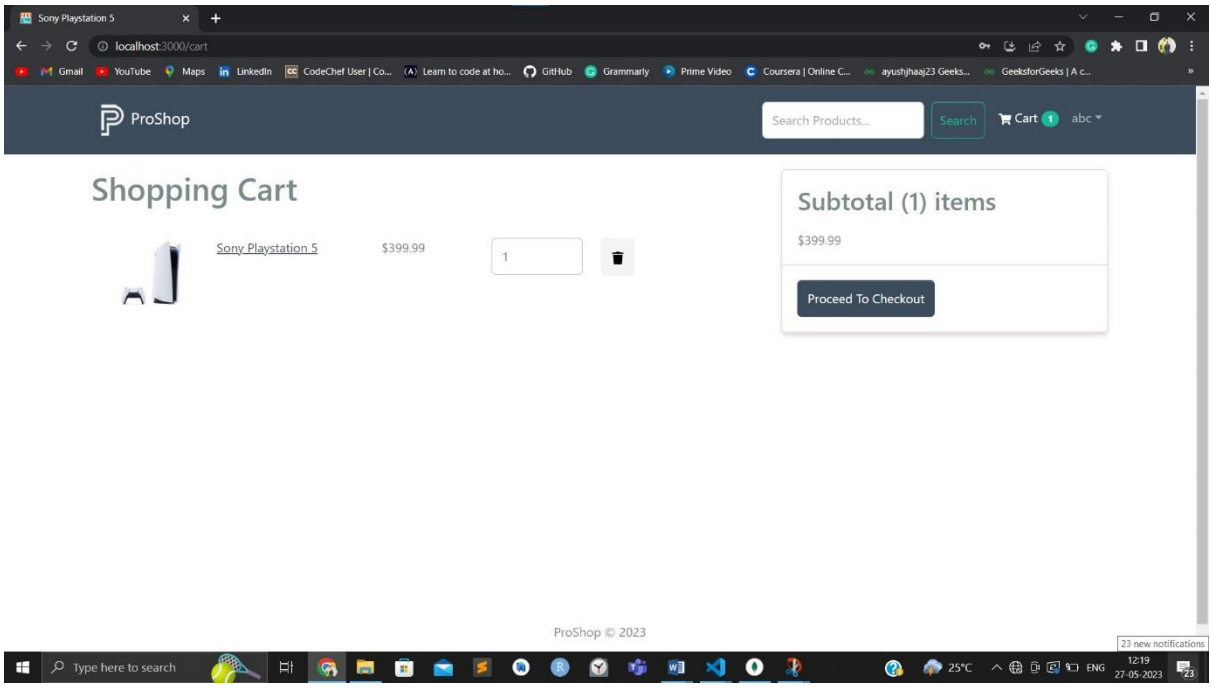


Figure 4.7 Add to Cart

Step 4:

Fill the Shipping details like your Address

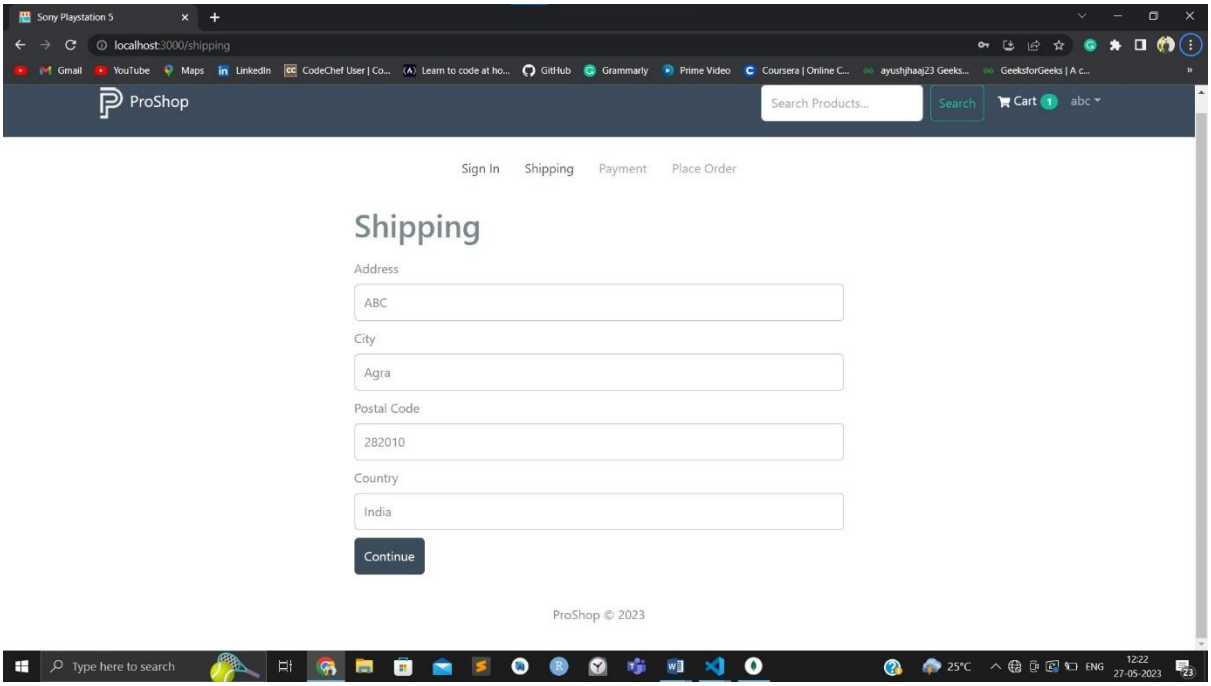


Figure 4.8 Shipping Details

Step 5:

Select the Payment Method (Payment Gateway)

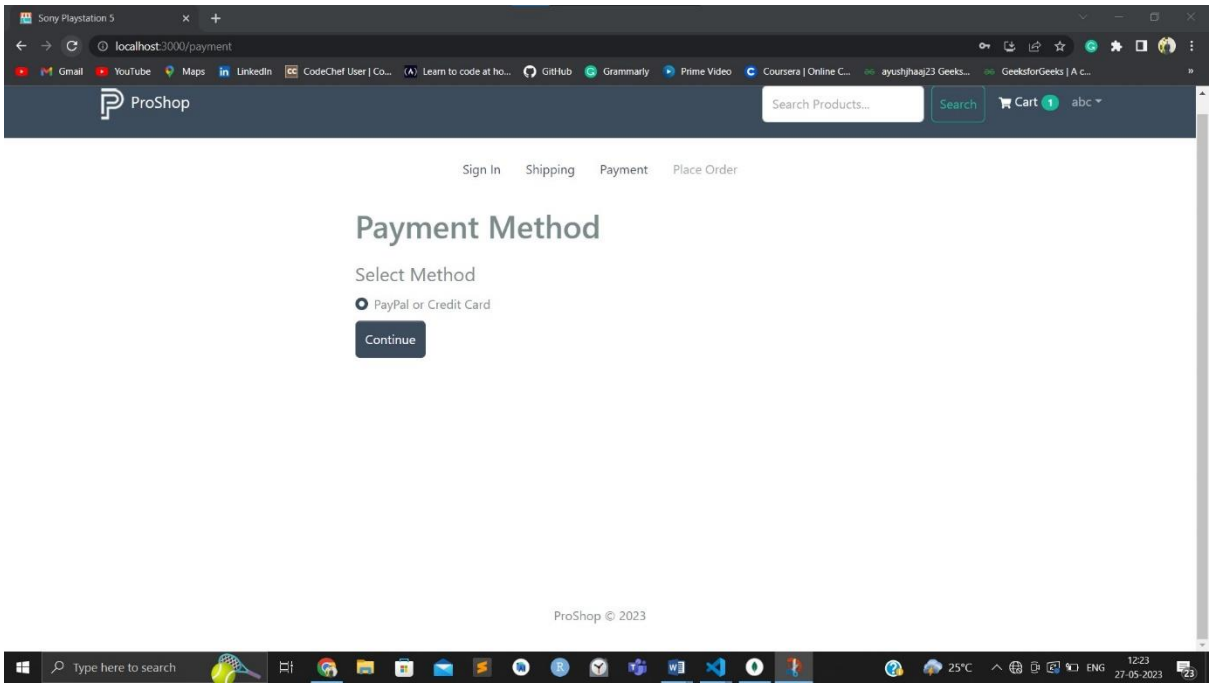


Figure 4.9 Payment Gateway

Step 6:

Order Summary will be generated

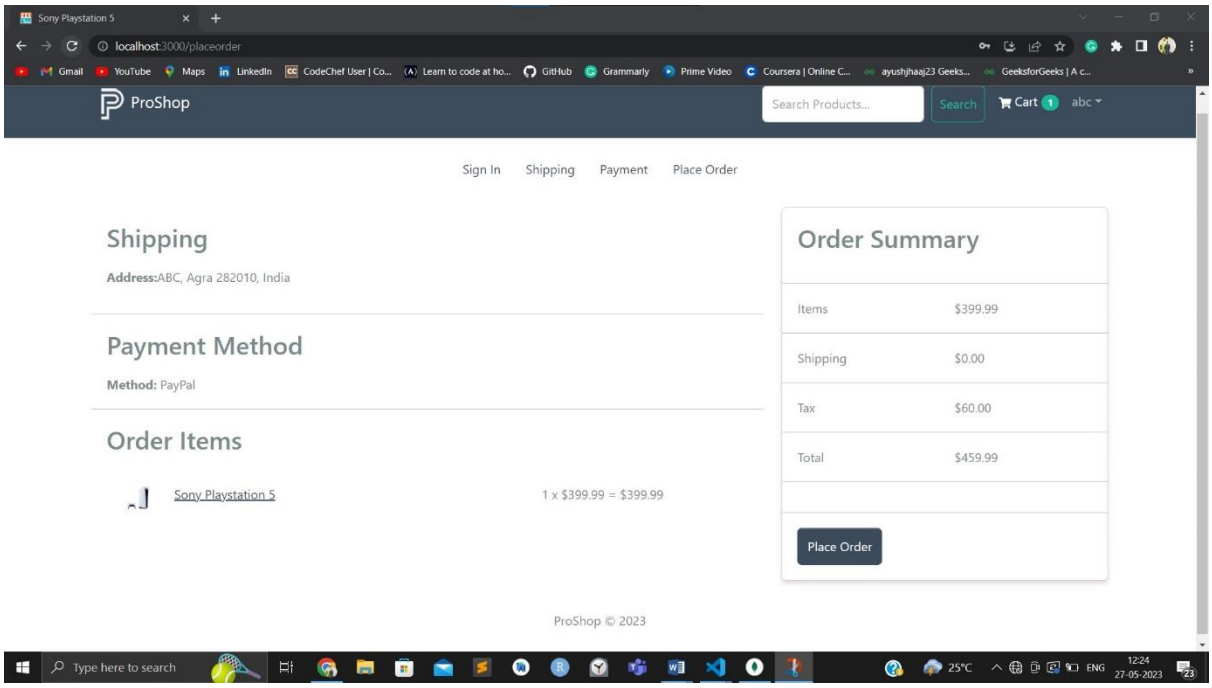


Figure 4.10 Order Summary

Step 7:

Order Status will be generated

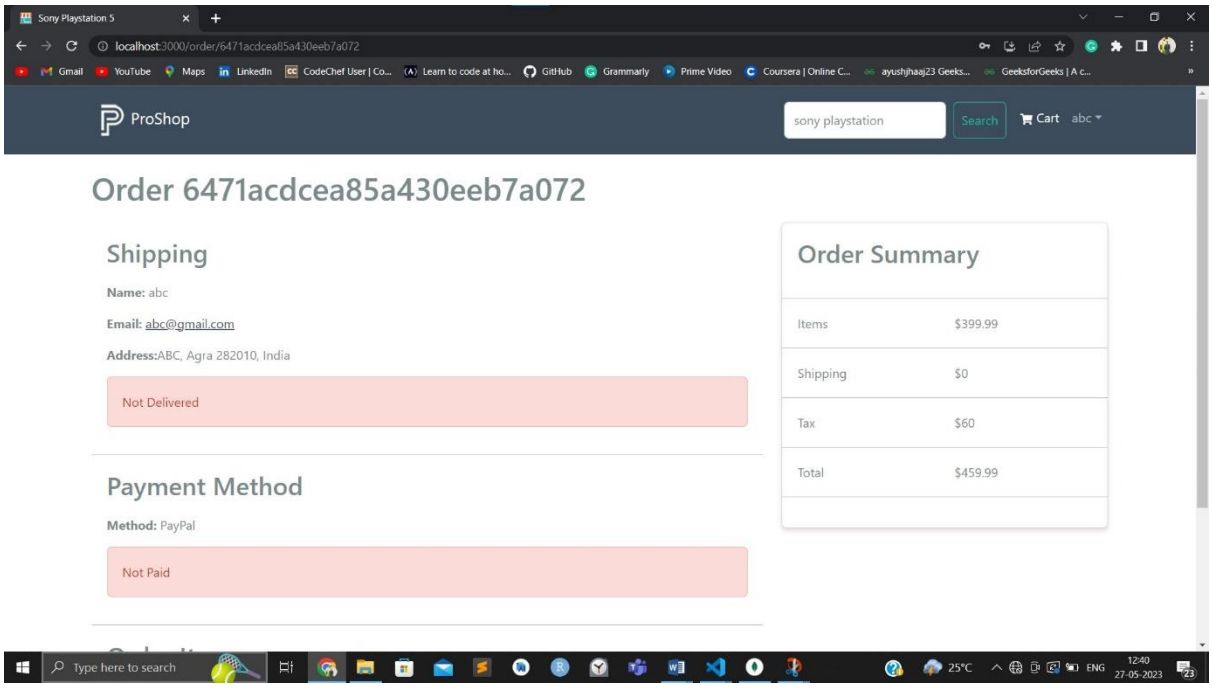


Figure 4.11 Order Status

Admin Controls

Products

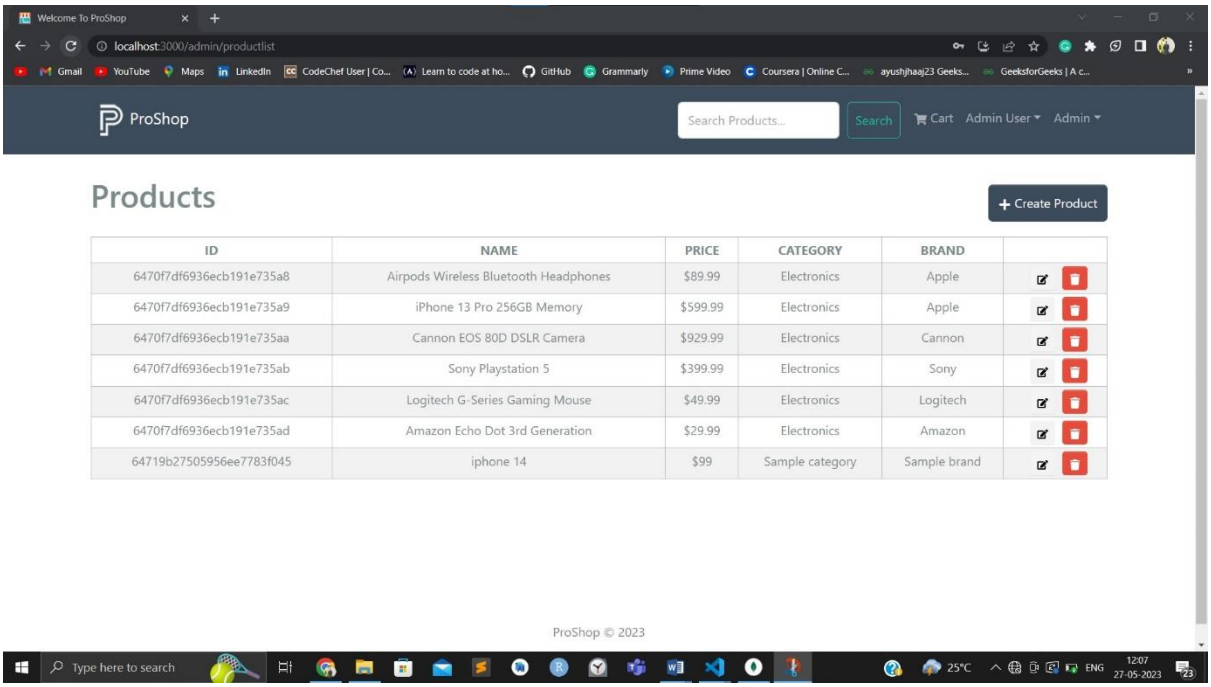


Figure 4.12 Create Product

Order

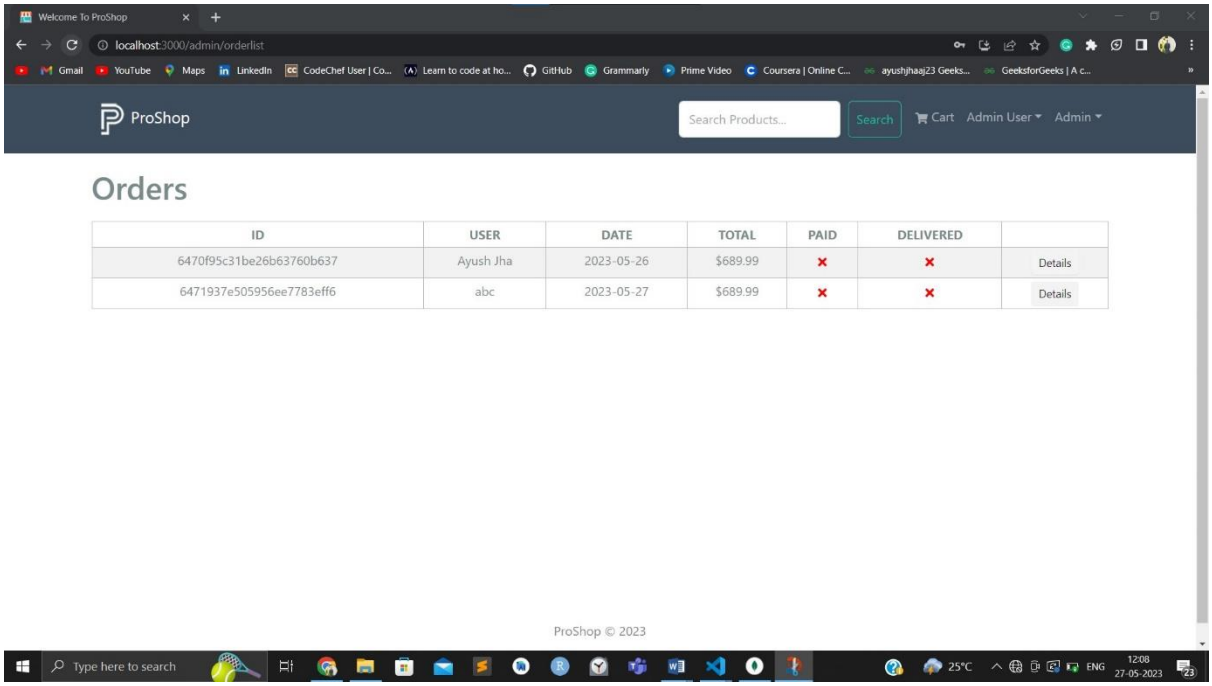


Figure 4.13 Total Orders

User

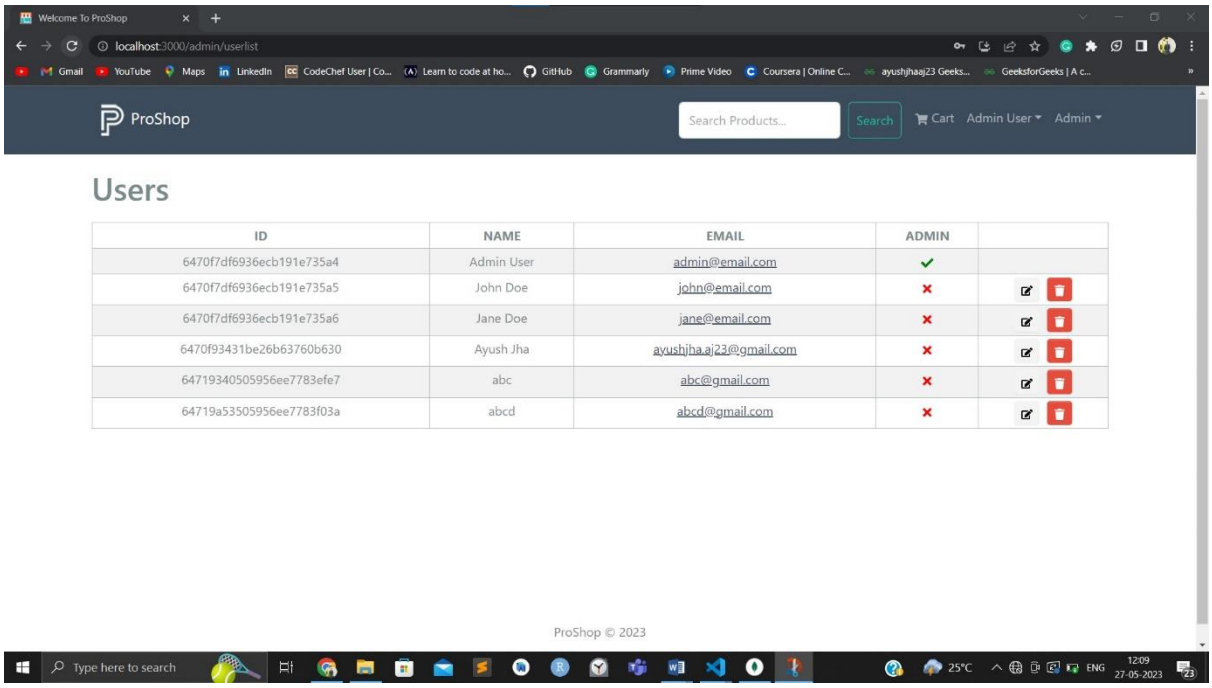


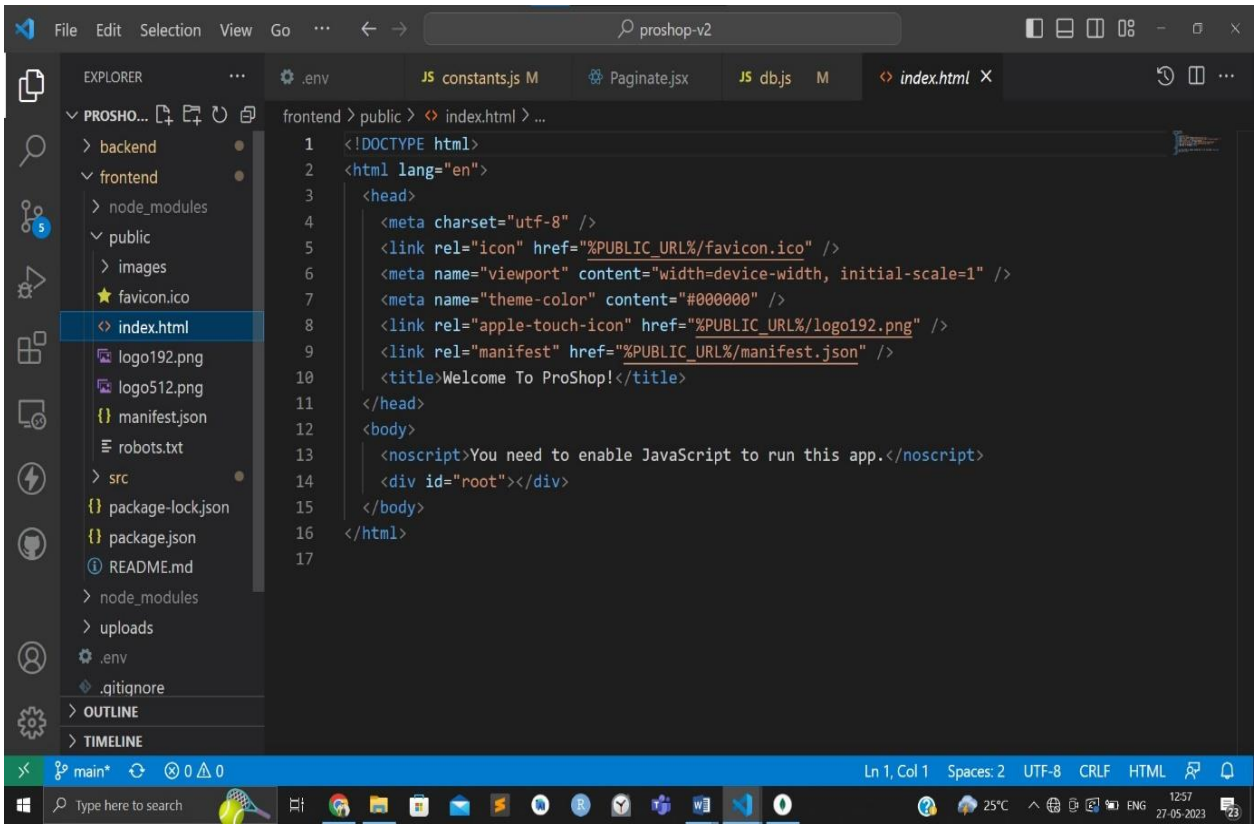
Figure 4.14 Total Users

CHAPTER 5

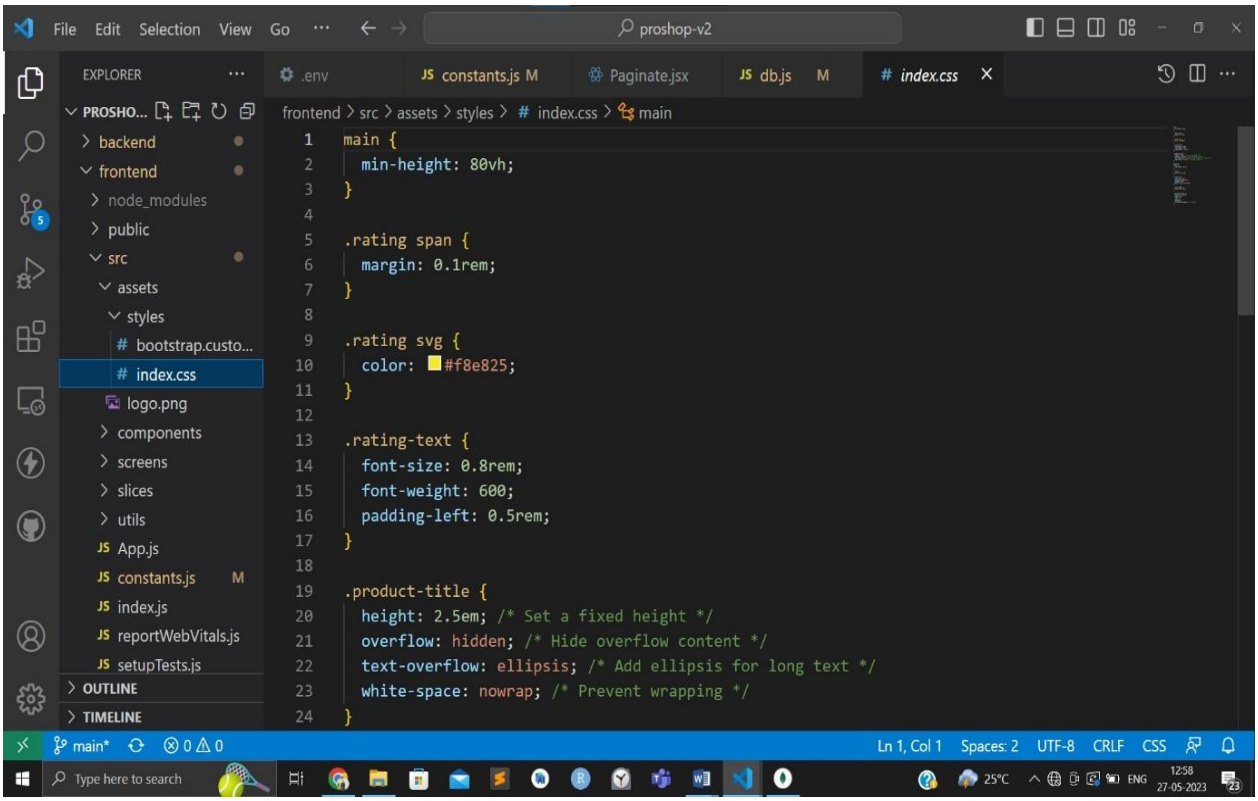
CODING

Front-End

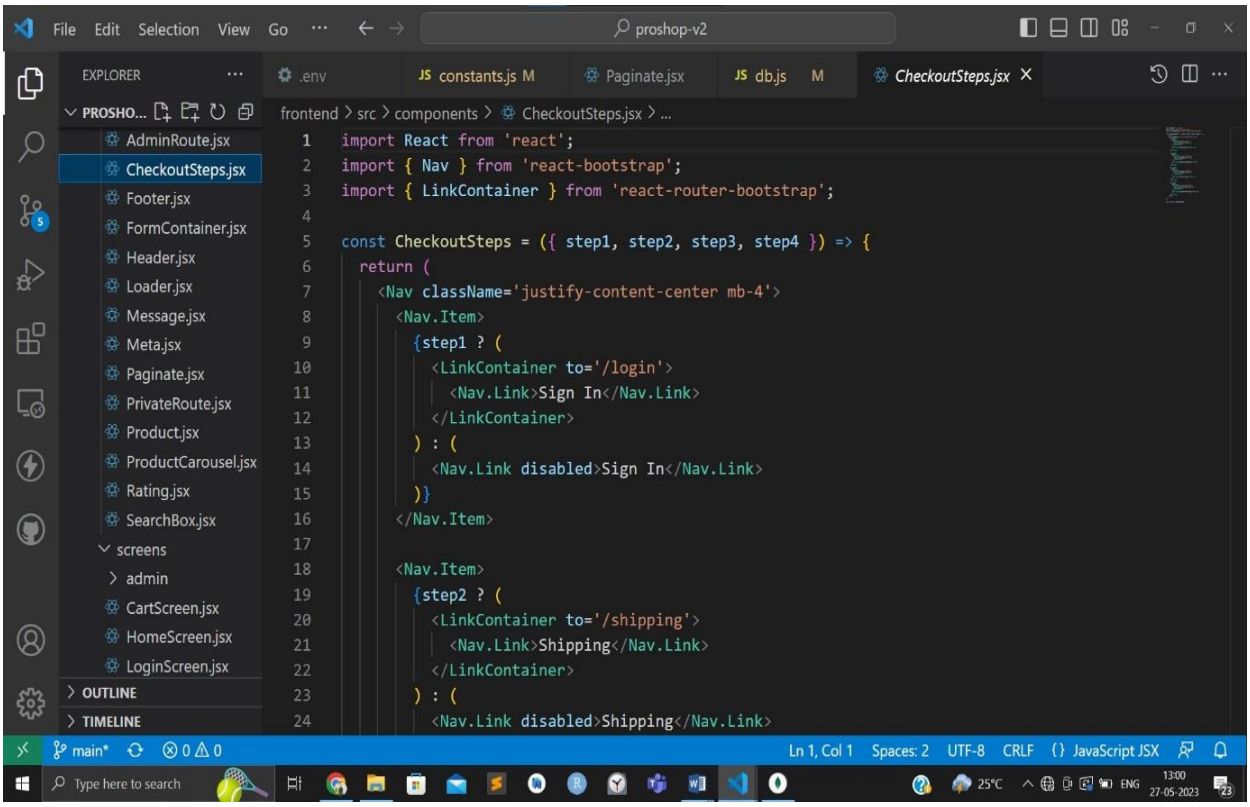
1. Index.html



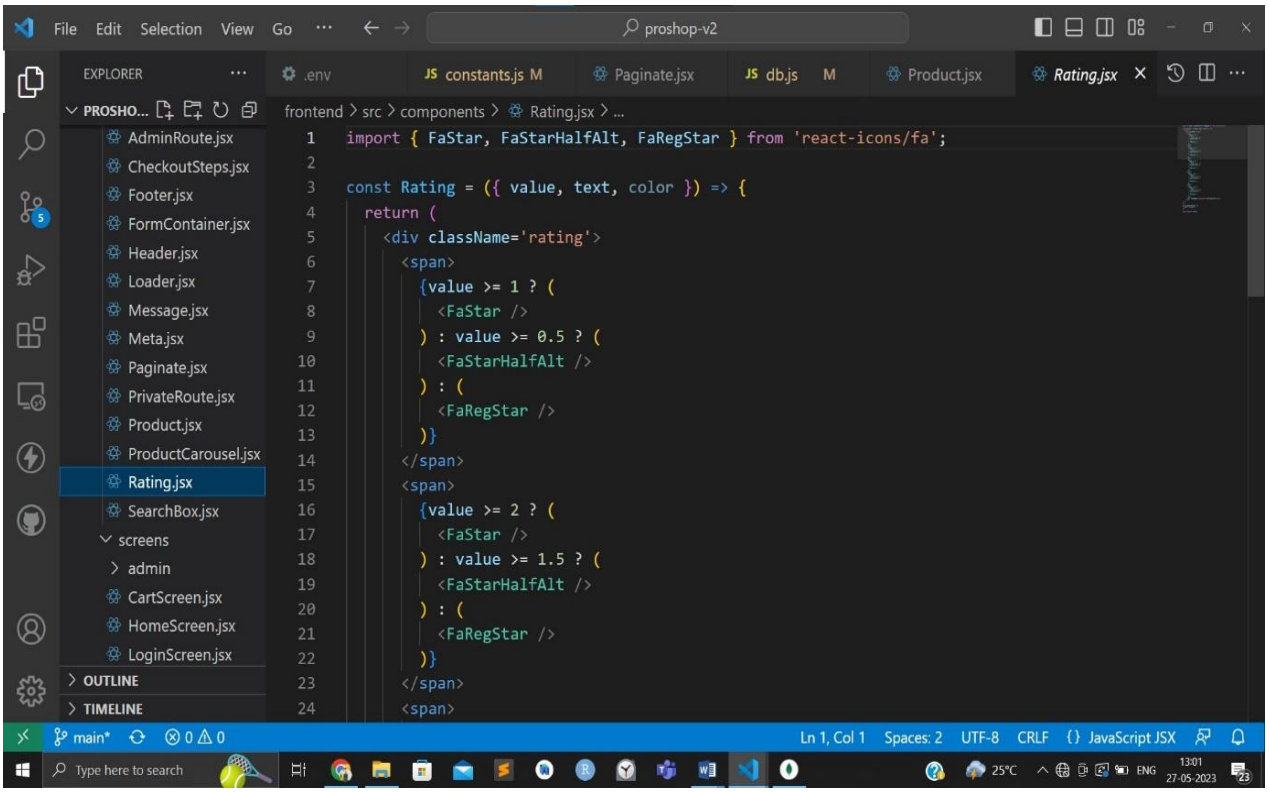
2. Index.css


File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
index.css
frontend > src > assets > styles > # index.css > main
PROSHO...
backend
frontend
node_modules
public
src
assets
styles
bootstrap.custo...
index.css
logo.png
components
screens
slices
utils
App.js
constants.js M
index.js
reportWebVitals.js
setupTests.js
OUTLINE
TIMELINE
1 main {
2 min-height: 80vh;
3 }
4
5 .rating span {
6 margin: 0.1rem;
7 }
8
9 .rating svg {
10 color: #f8e825;
11 }
12
13 .rating-text {
14 font-size: 0.8rem;
15 font-weight: 600;
16 padding-left: 0.5rem;
17 }
18
19 .product-title {
20 height: 2.5em; /* Set a fixed height */
21 overflow: hidden; /* Hide overflow content */
22 text-overflow: ellipsis; /* Add ellipsis for long text */
23 white-space: nowrap; /* Prevent wrapping */
24 }
main*
0 0
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
CSS
Type here to search
25°C
ENG
12:58
27-05-2023

3. CheckOutSteps.jsx



4. Rating.jsx

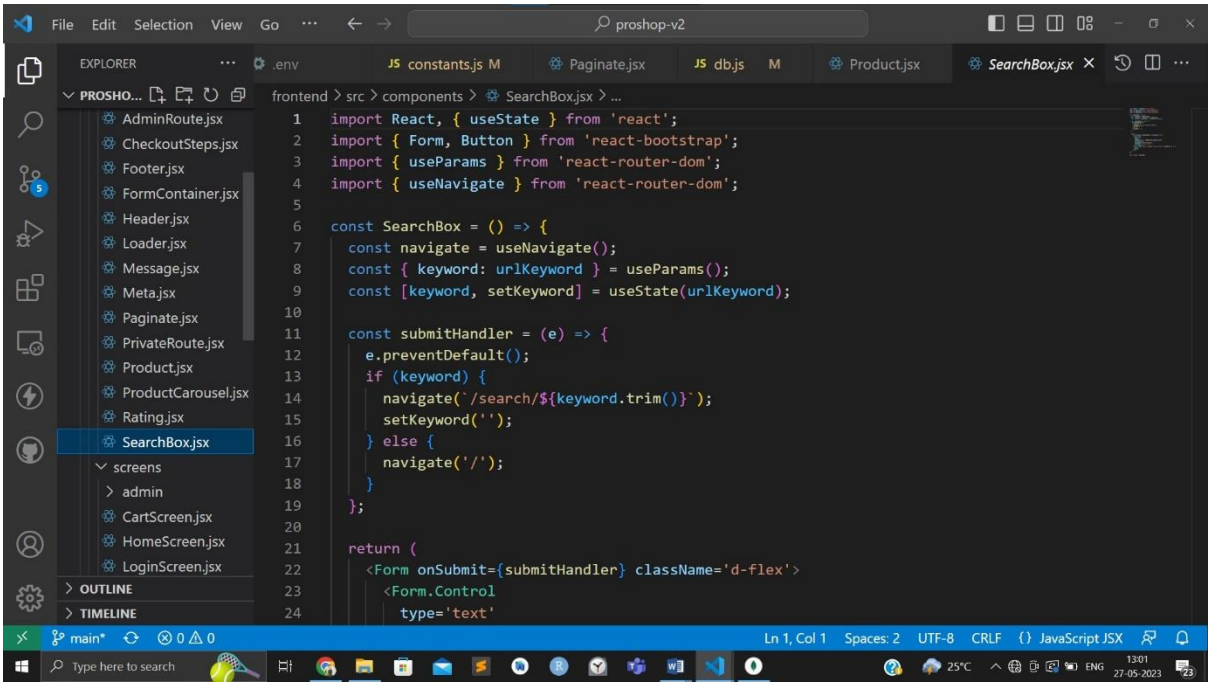

File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
Product.jsx
Rating.jsx
frontend > src > components > Rating.jsx > ...
AdminRoute.jsx
CheckoutSteps.jsx
Footer.jsx
FormContainer.jsx
Header.jsx
Loader.jsx
Message.jsx
Meta.jsx
Paginate.jsx
PrivateRoute.jsx
Product.jsx
ProductCarousel.jsx
Rating.jsx
SearchBox.jsx
screens
admin
CartScreen.jsx
HomeScreen.jsx
LoginScreen.jsx
OUTLINE
TIMELINE
import { FaStar, FaStarHalfAlt, FaRegStar } from 'react-icons/fa';
const Rating = ({ value, text, color }) => {
return (

{value >= 1 ? (
<FaStar />
) : value >= 0.5 ? (
<FaStarHalfAlt />
) : (
<FaRegStar />
)}

{value >= 2 ? (
<FaStar />
) : value >= 1.5 ? (
<FaStarHalfAlt />
) : (
<FaRegStar />
)}

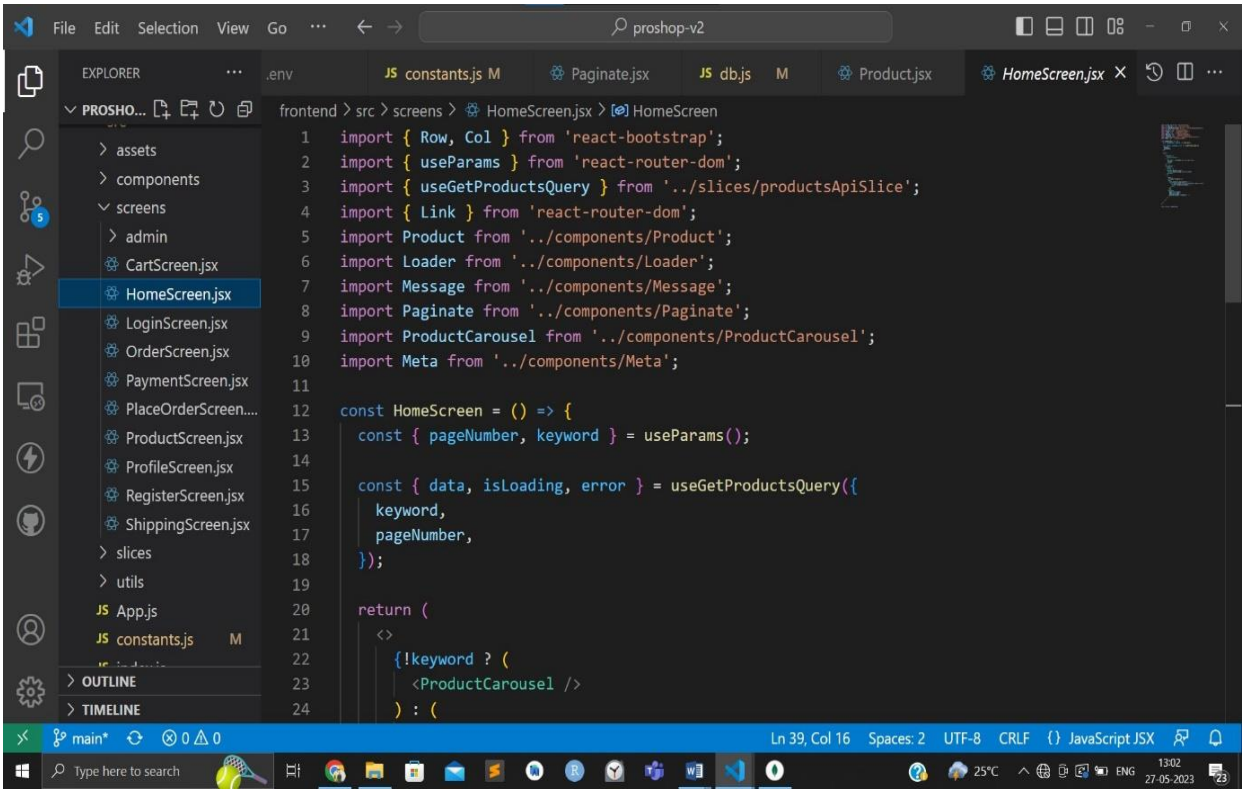
main*
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
JavaScript JSX
Type here to search
25°C
ENG
13:01
27-05-2023

5. SearchBox.jsx

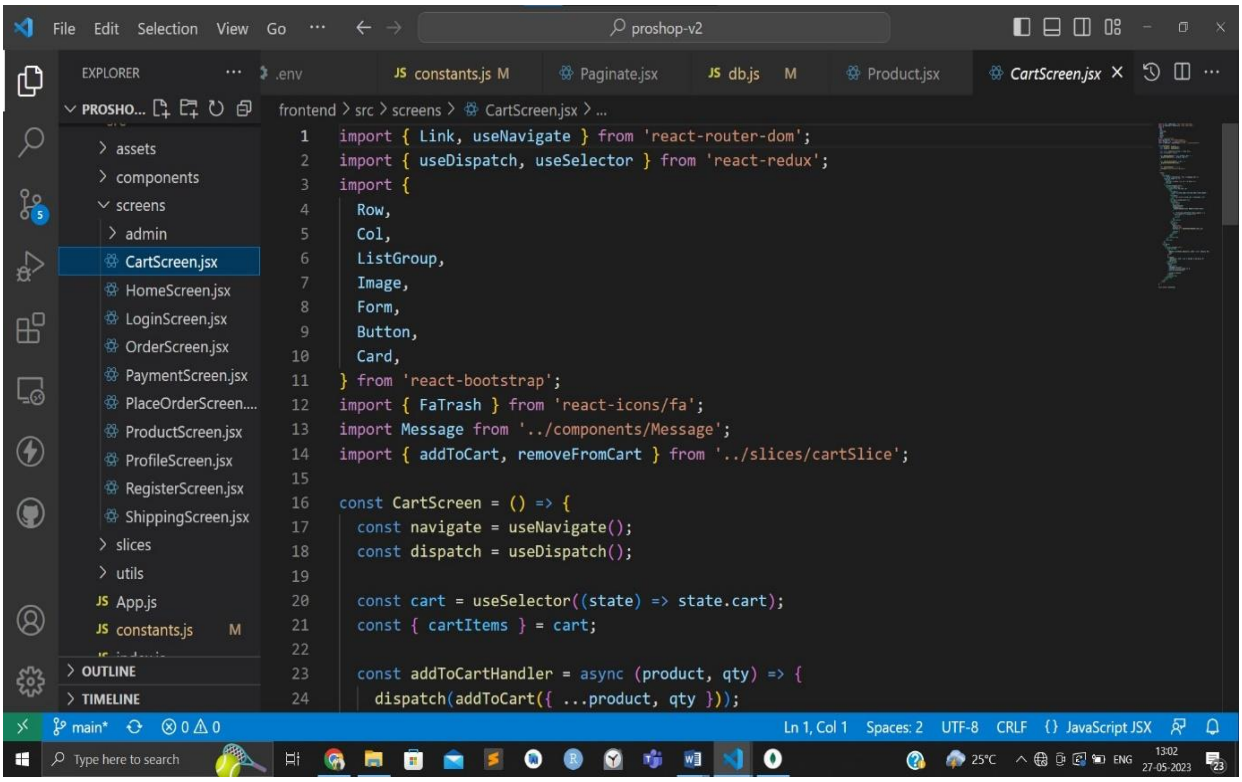


Screens

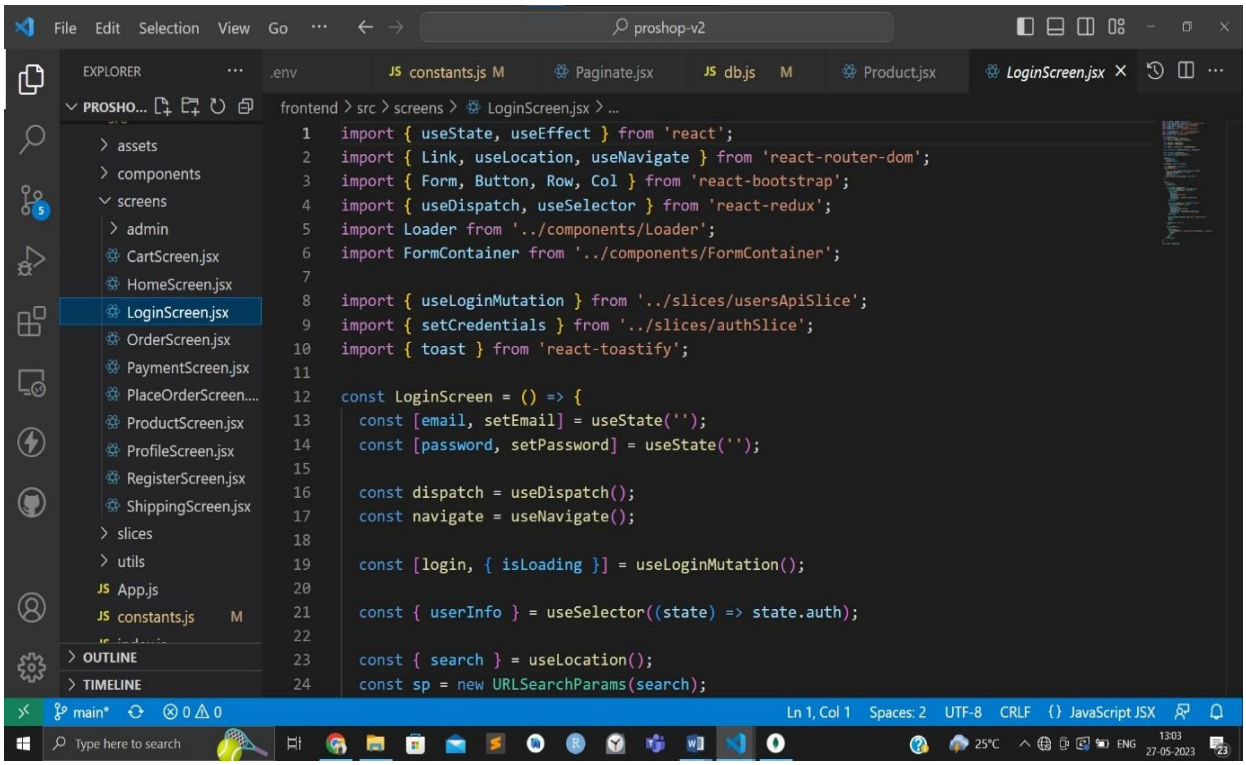
1. Home Screen



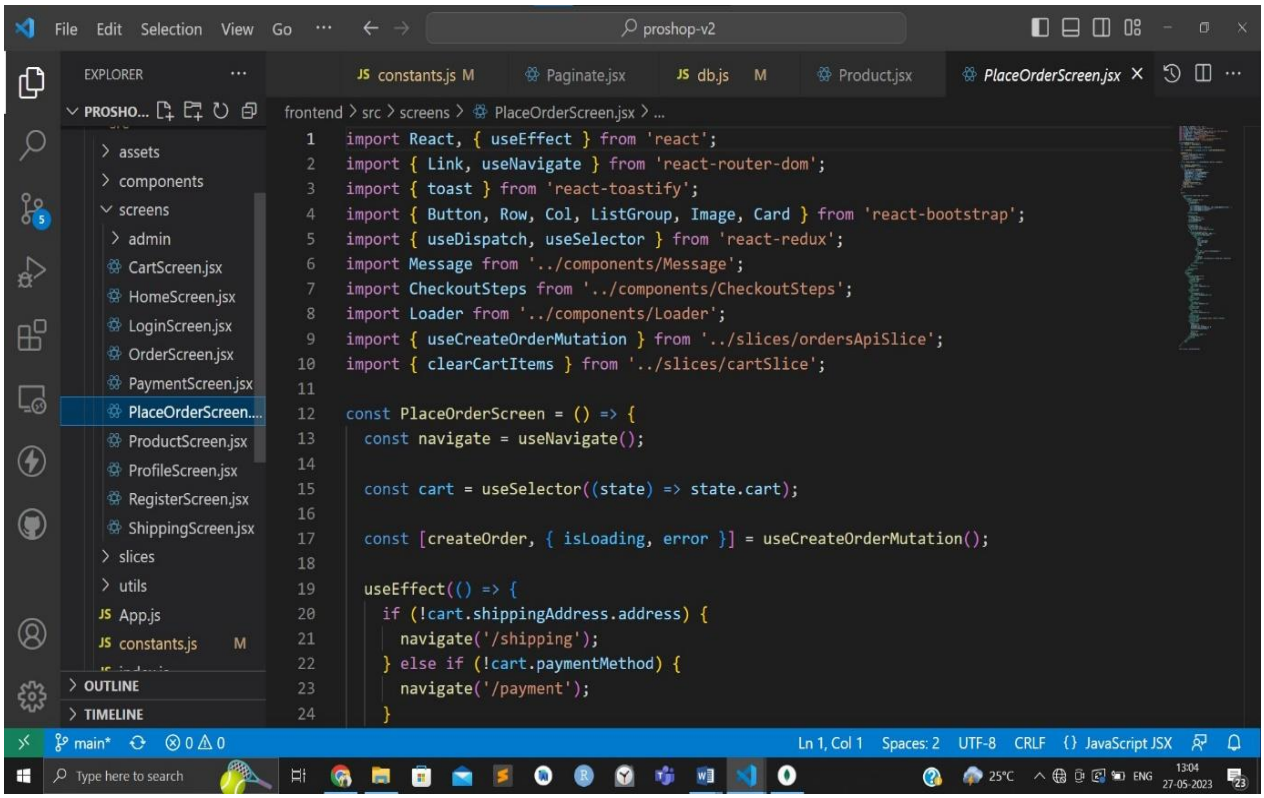
2. Cart Screen



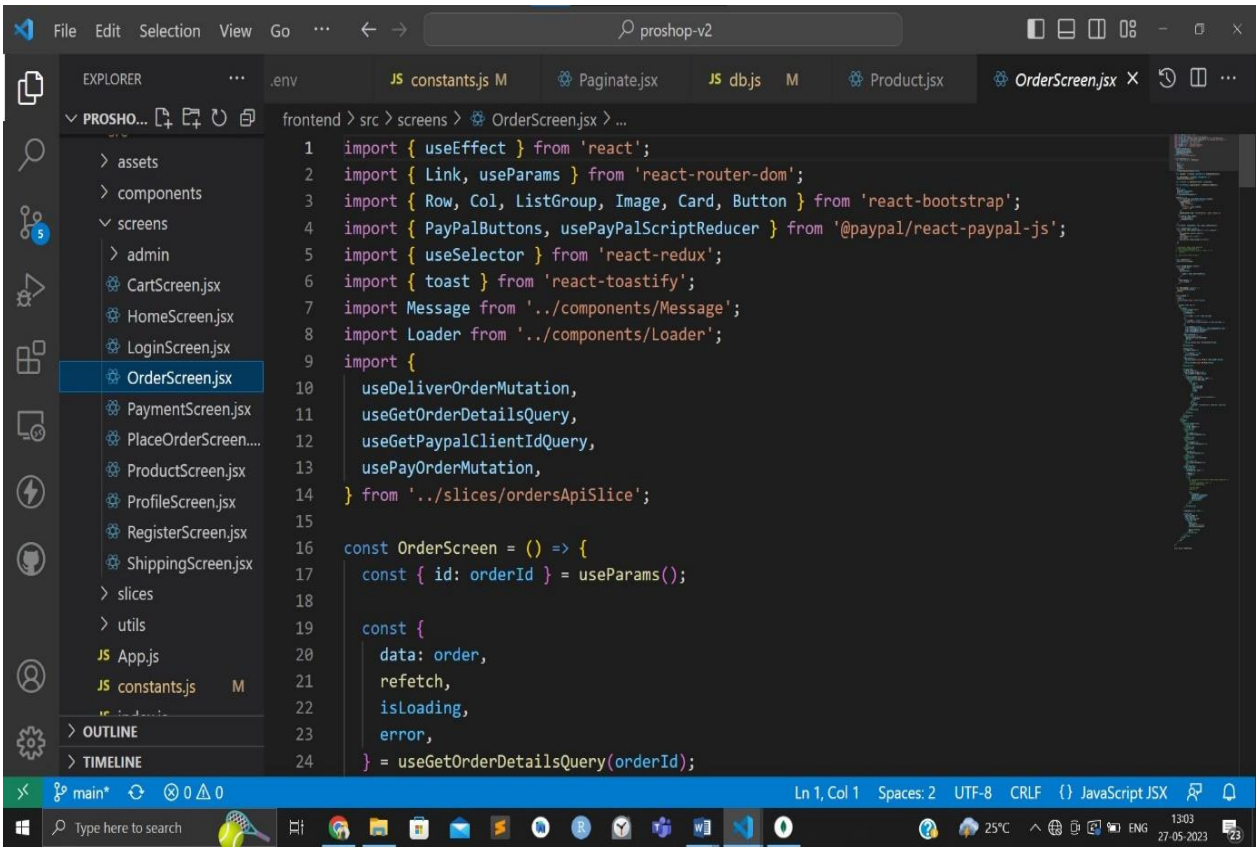
3. Login Screen



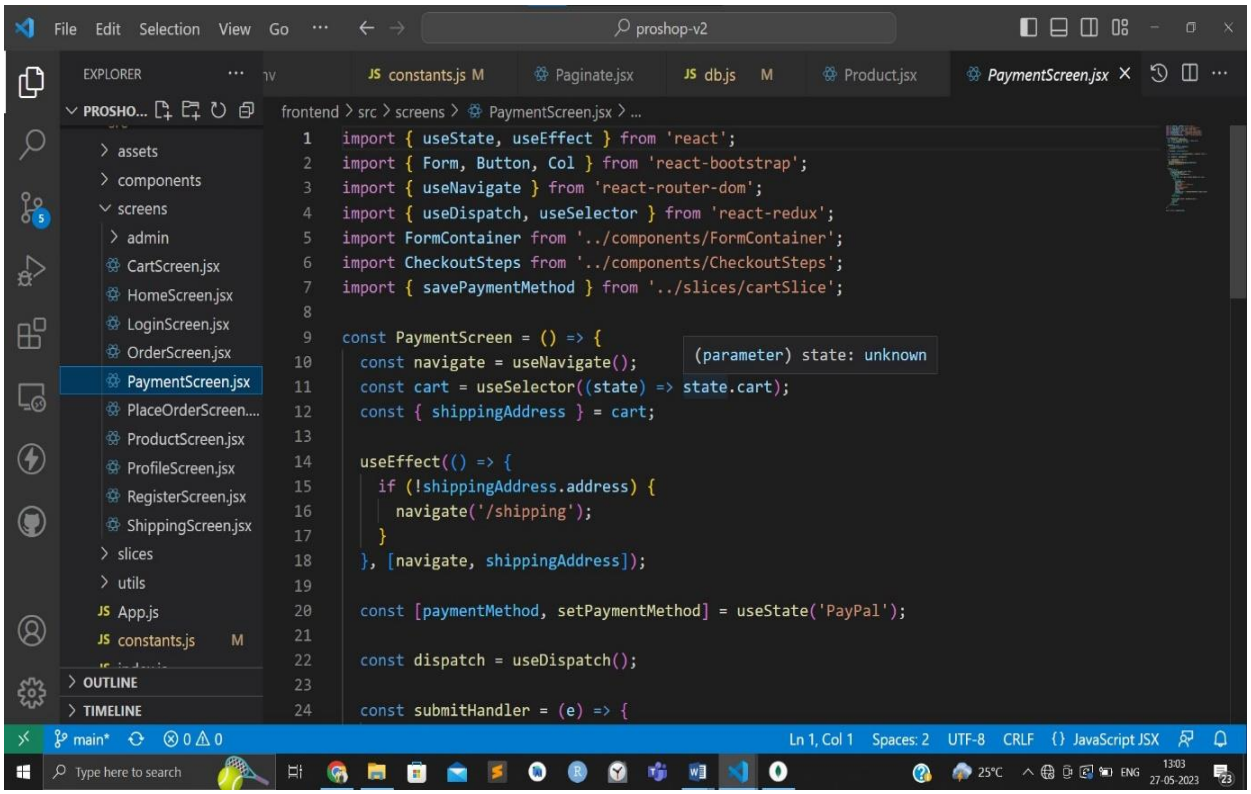
4. OrderPlacedScreen



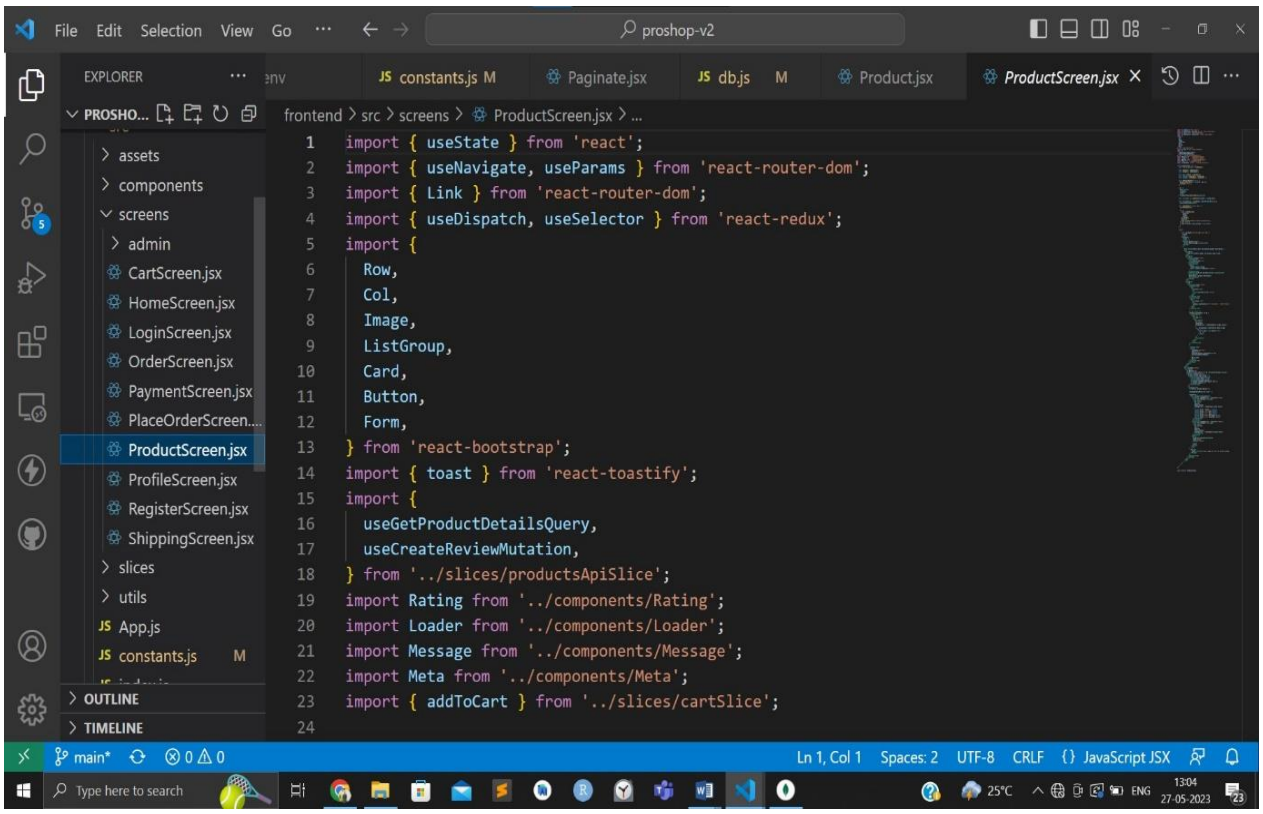
5. Order Screen



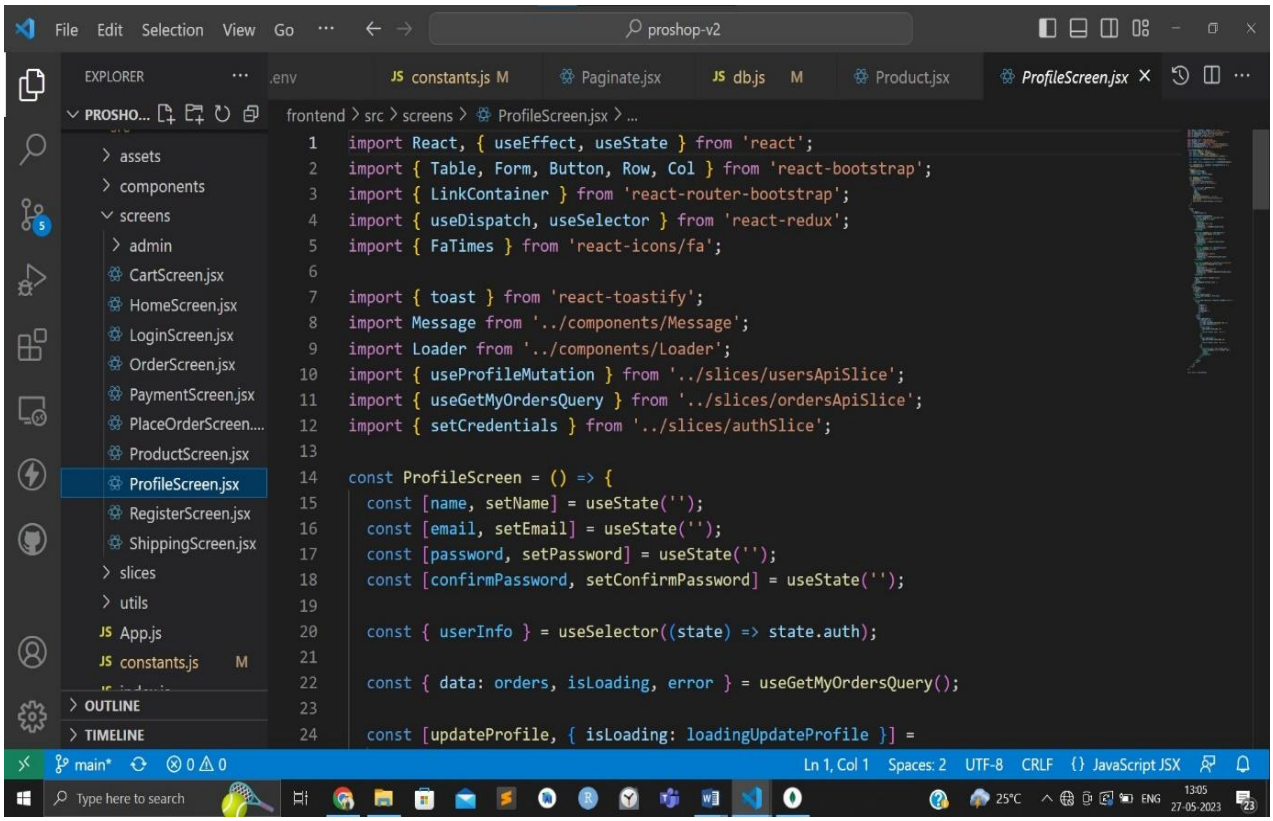
6. Payment Screen



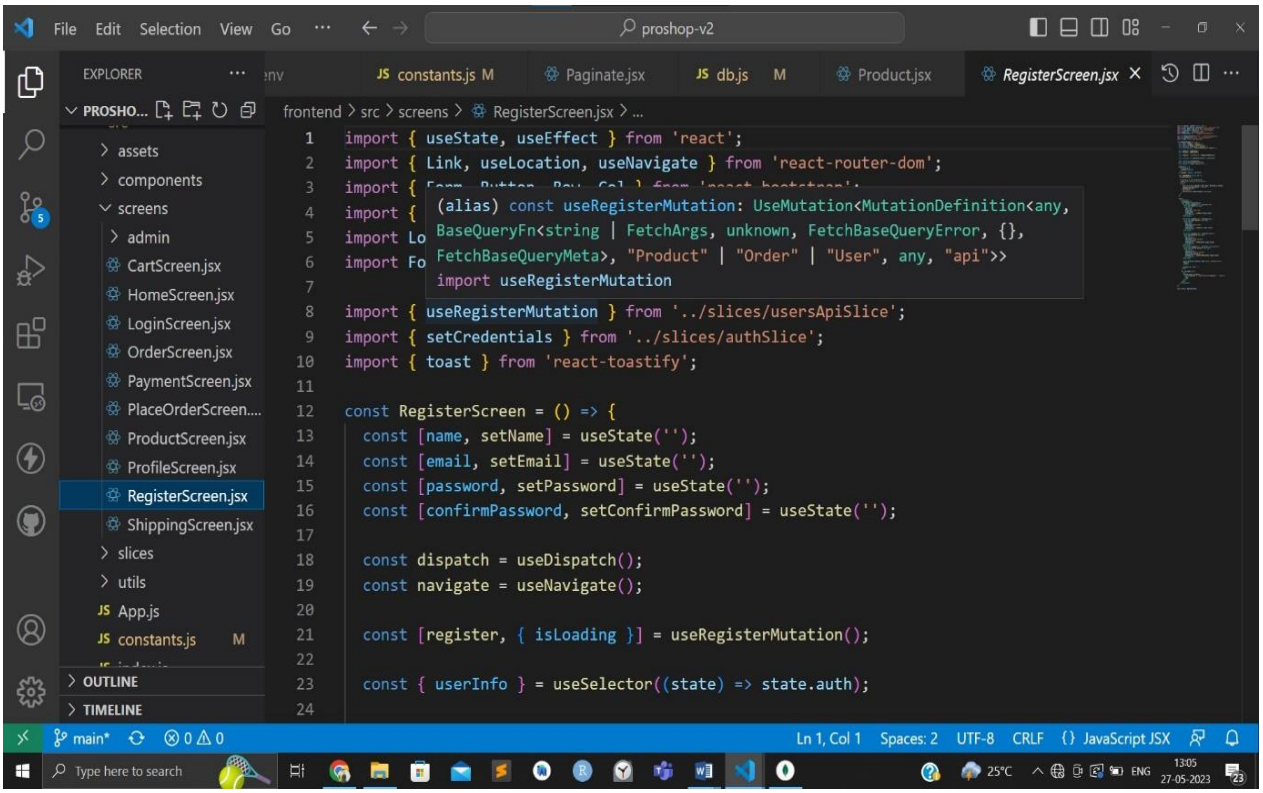
7. Product Screen



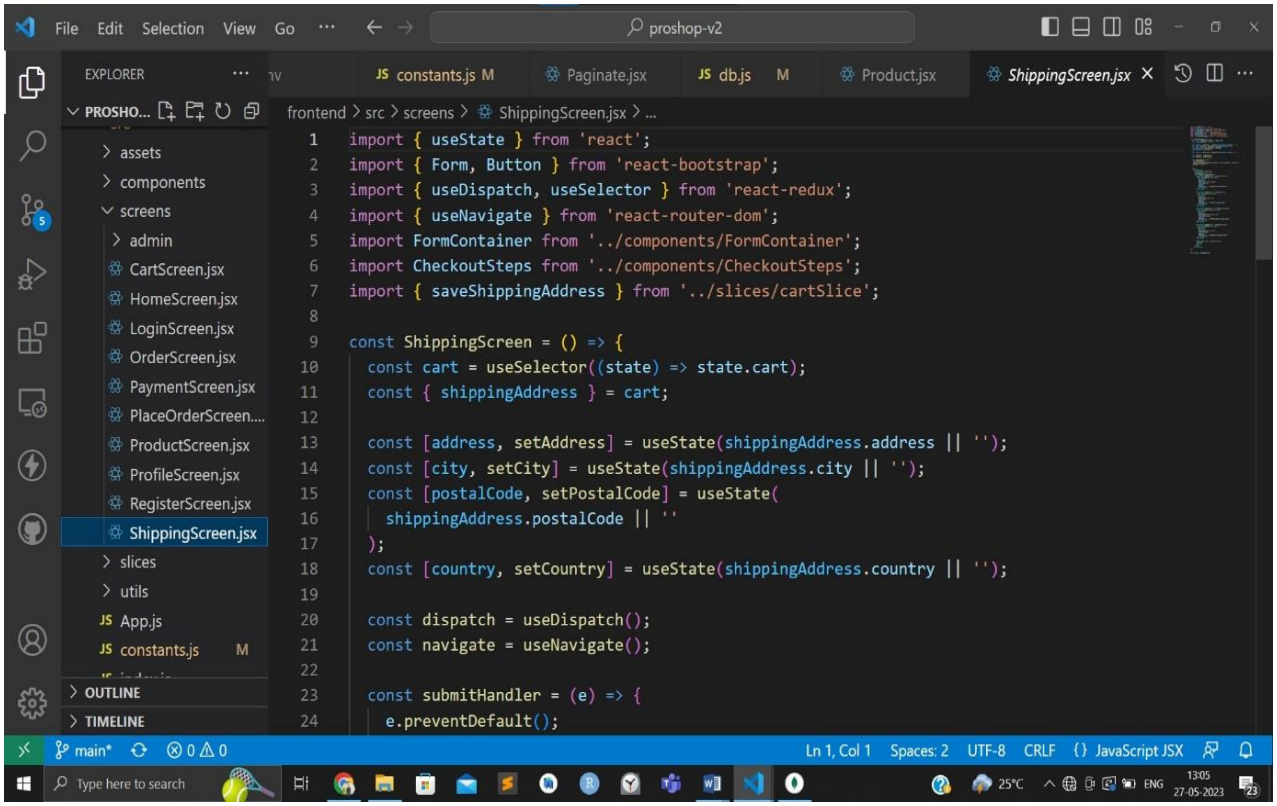
8. Profile Screen



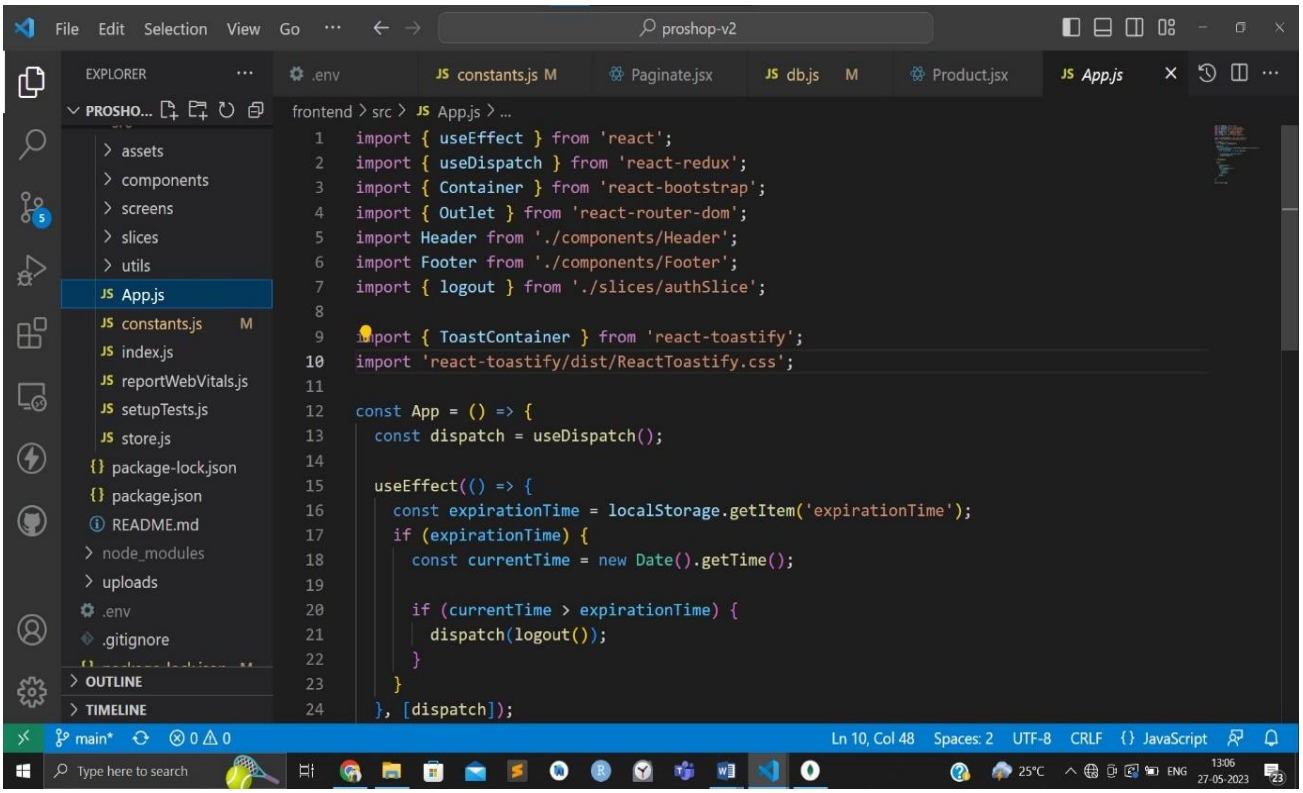
9. Register Screen



10. Shipping Screen

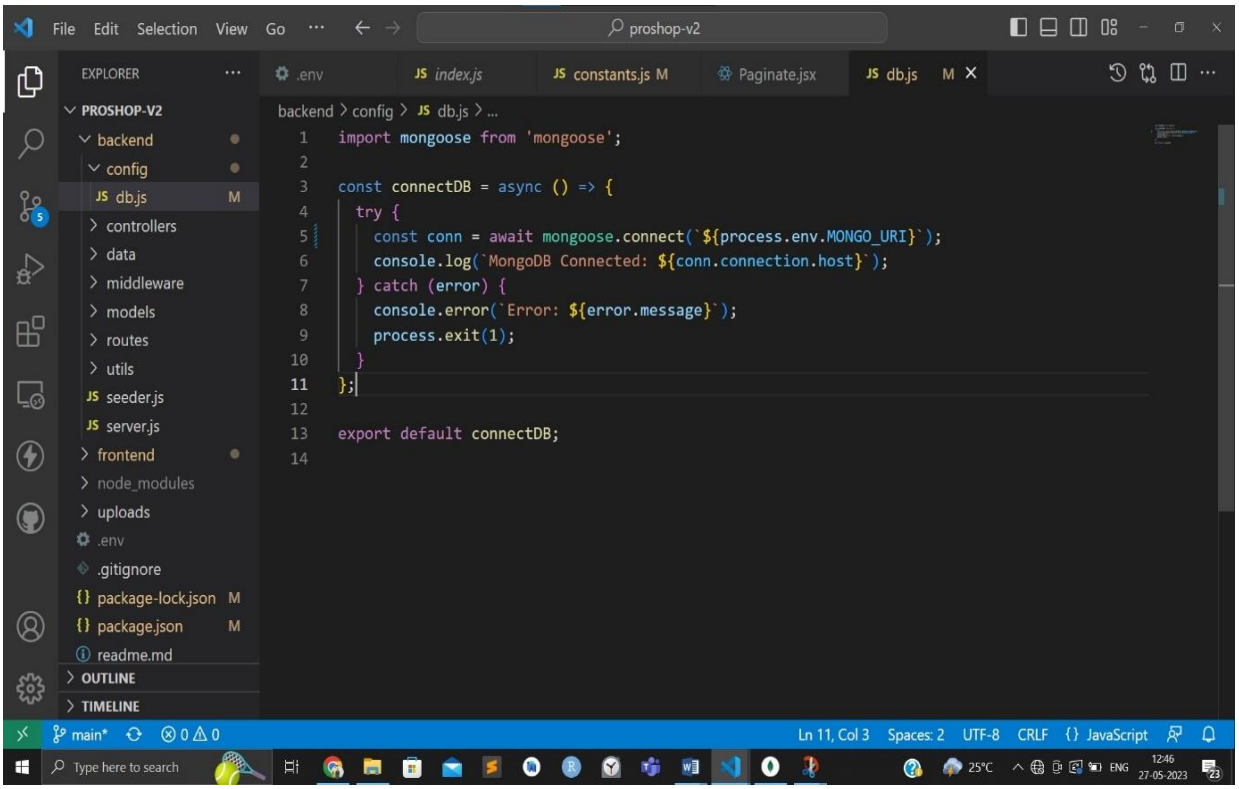


App.js


File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
Product.jsx
App.js
frontend > src > App.js > ...
assets
components
screens
slices
utils
App.js
constants.js M
index.js
reportWebVitals.js
setupTests.js
store.js
package-lock.json
package.json
README.md
node_modules
uploads
.env
.gitignore
OUTLINE
TIMELINE
import { useEffect } from 'react';
import { useDispatch } from 'react-redux';
import { Container } from 'react-bootstrap';
import { Outlet } from 'react-router-dom';
import Header from './components/Header';
import Footer from './components/Footer';
import { logout } from './slices/authSlice';
import { ToastContainer } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
const App = () => {
const dispatch = useDispatch();
useEffect(() => {
const expirationTime = localStorage.getItem('expirationTime');
if (expirationTime) {
const currentTime = new Date().getTime();
if (currentTime > expirationTime) {
dispatch(logout());
}
}
}, [dispatch]);
main*
Ln 10, Col 48
Spaces: 2
UTF-8
CRLF
JavaScript
Type here to search
25°C
ENG
13:06
27-05-2023

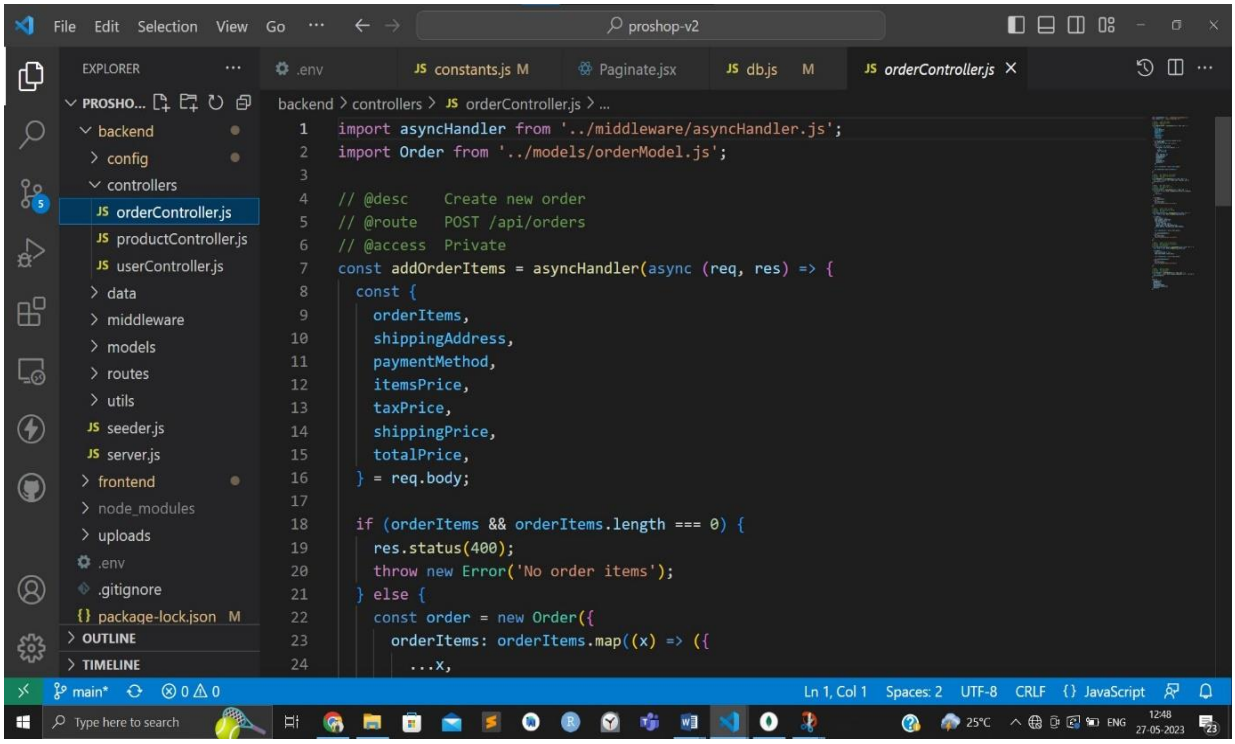
Back-End

Db.js

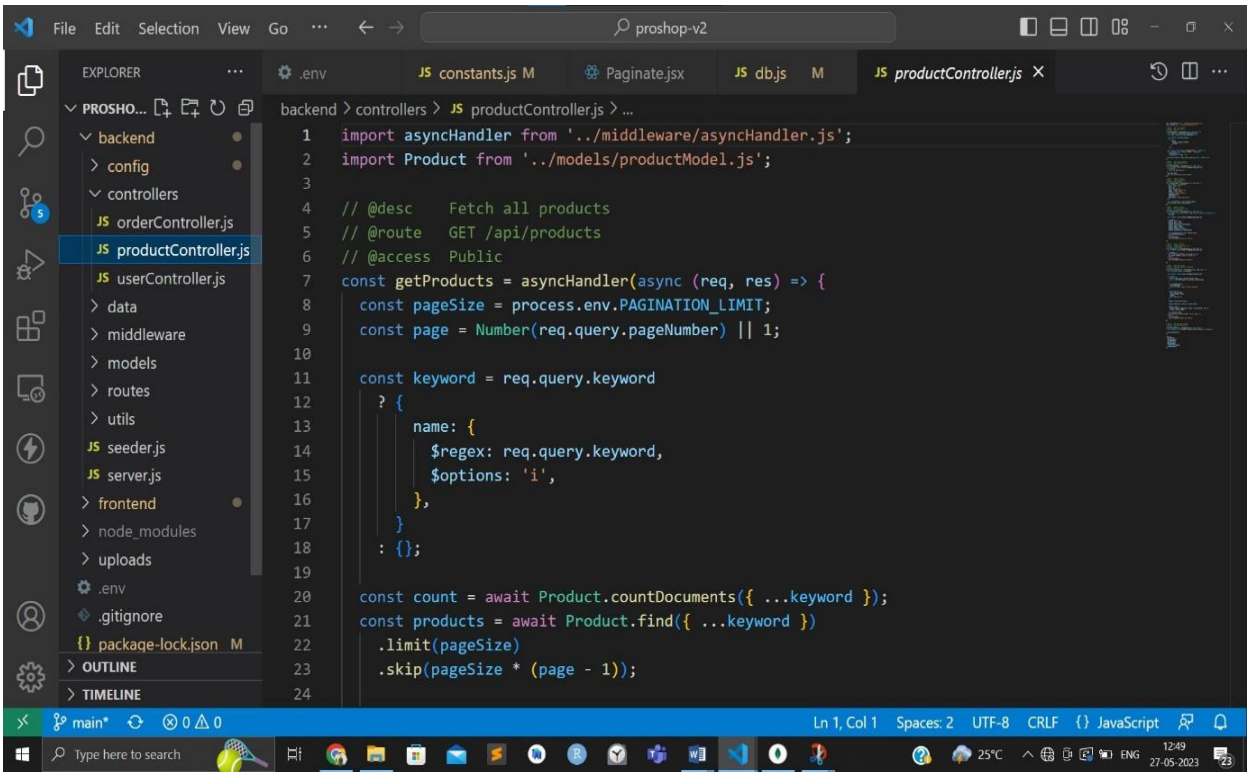


Controllers

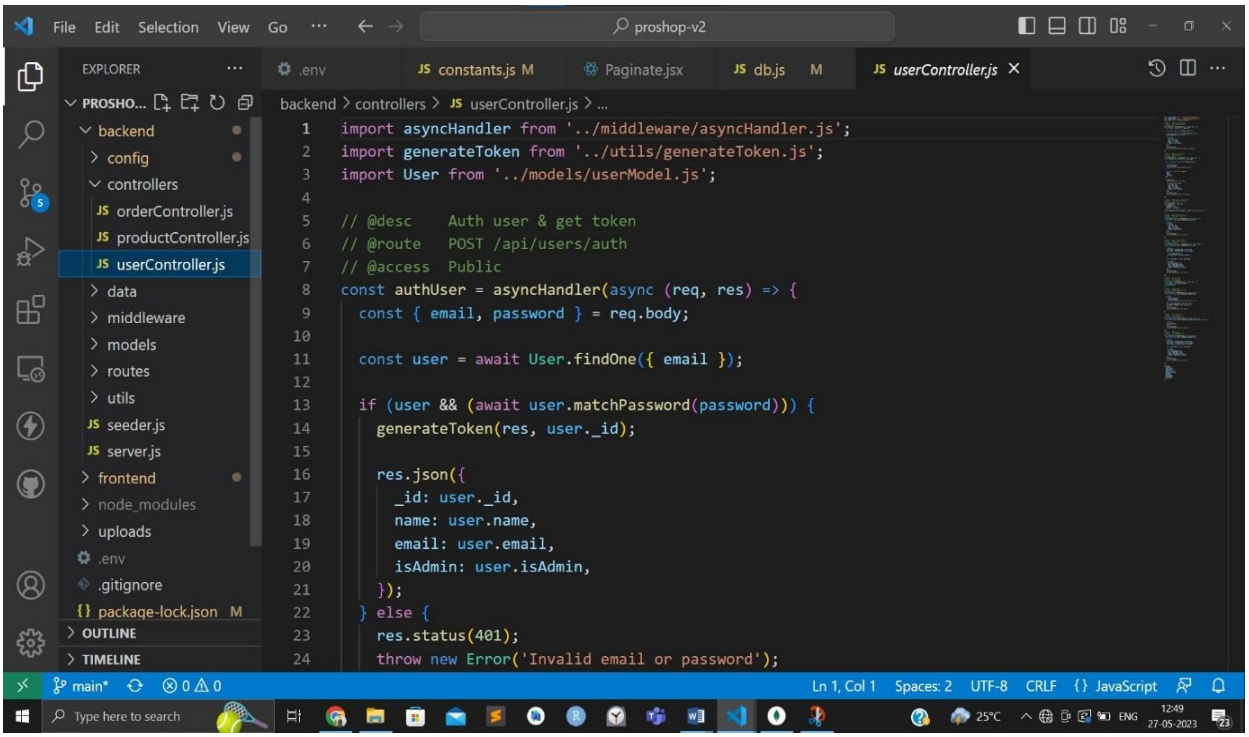
1. Order Controller



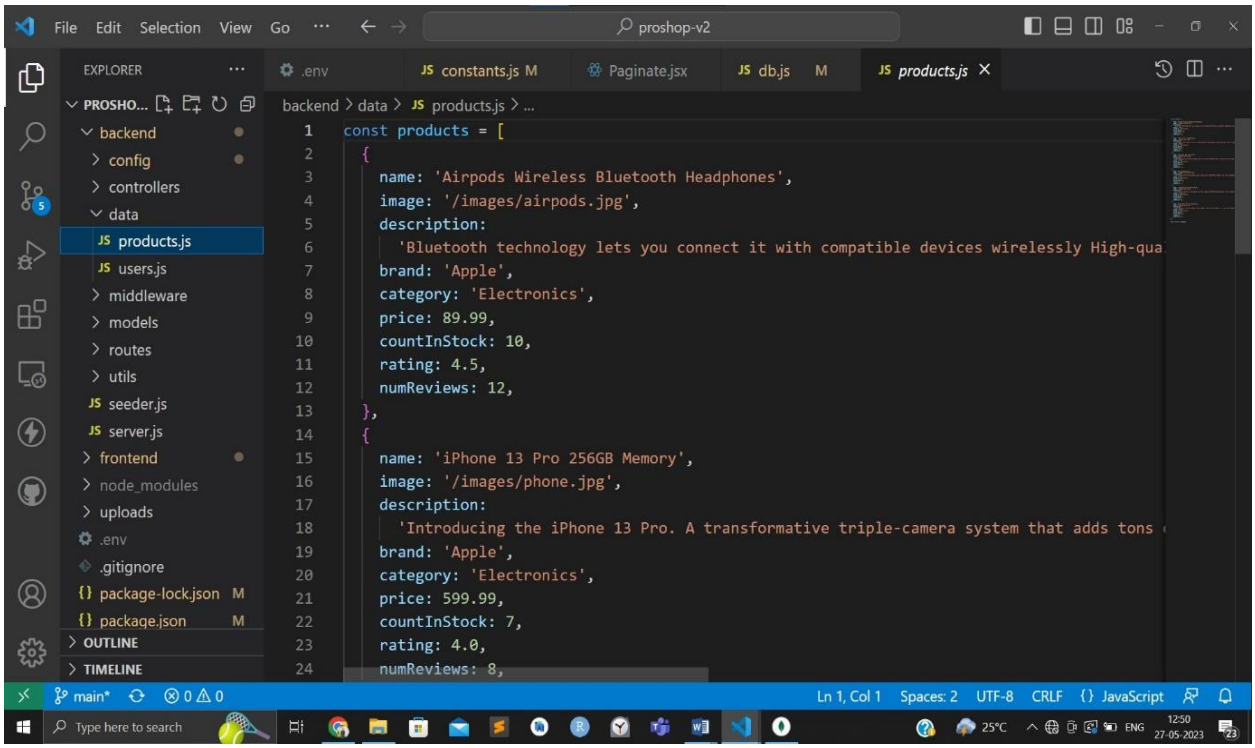
2. Product Controller



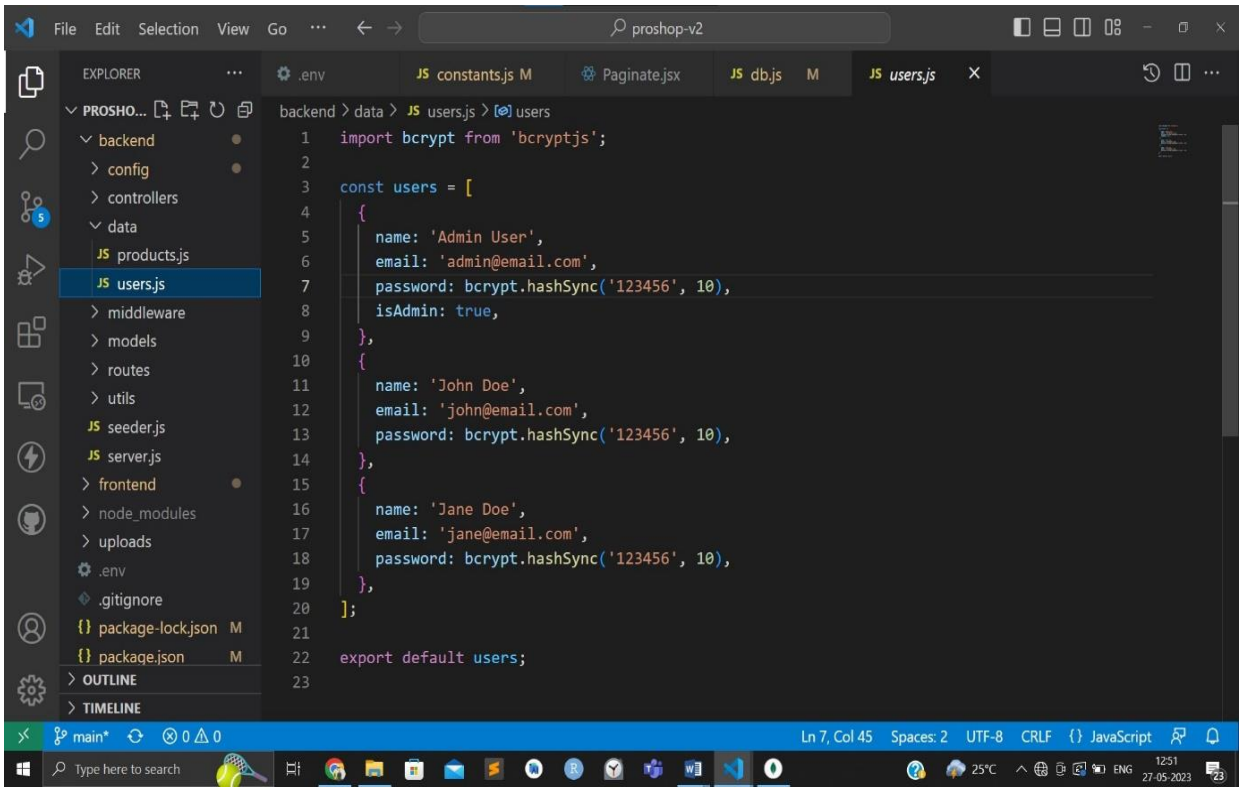
3. User Controller



Product – Data

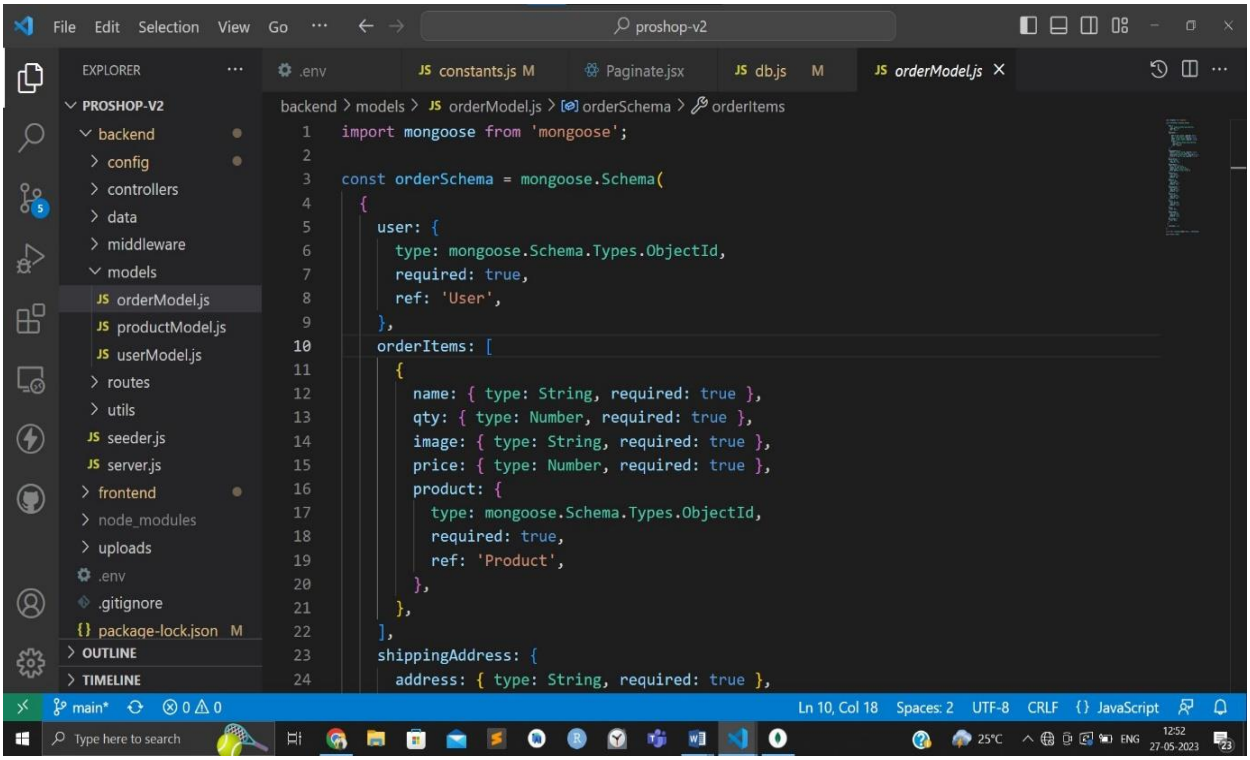


User – Data

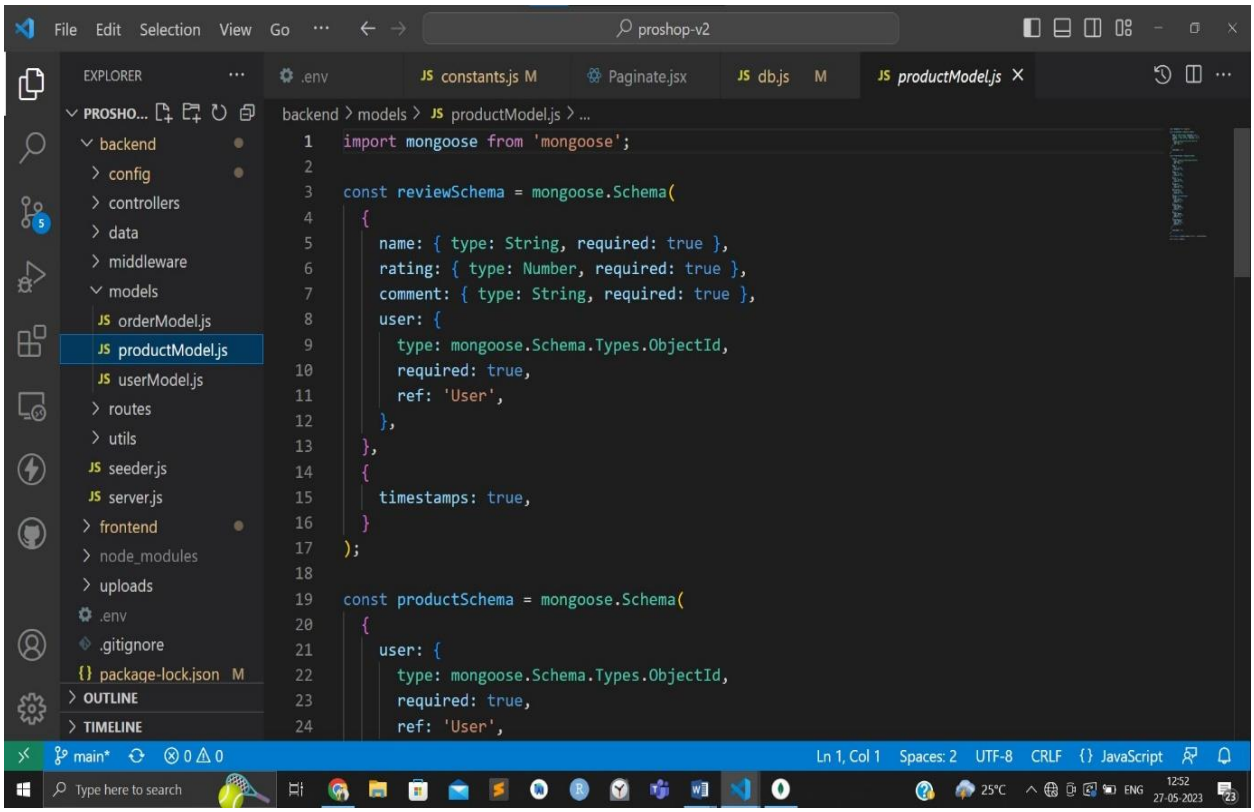

File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
users.js
backend > data > users.js > users
backend
config
controllers
data
products.js
users.js
middleware
models
routes
utils
seeder.js
server.js
frontend
node_modules
uploads
.env
.gitignore
package-lock.json M
package.json M
OUTLINE
TIMELINE
import bcrypt from 'bcryptjs';
const users = [
{
name: 'Admin User',
email: 'admin@email.com',
password: bcrypt.hashSync('123456', 10),
isAdmin: true,
},
{
name: 'John Doe',
email: 'john@email.com',
password: bcrypt.hashSync('123456', 10),
},
{
name: 'Jane Doe',
email: 'jane@email.com',
password: bcrypt.hashSync('123456', 10),
},
];
export default users;
main*
Ln 7, Col 45
Spaces: 2
UTF-8
CRLF
JavaScript
Type here to search
25°C
ENG
1251
27-05-2023

Models

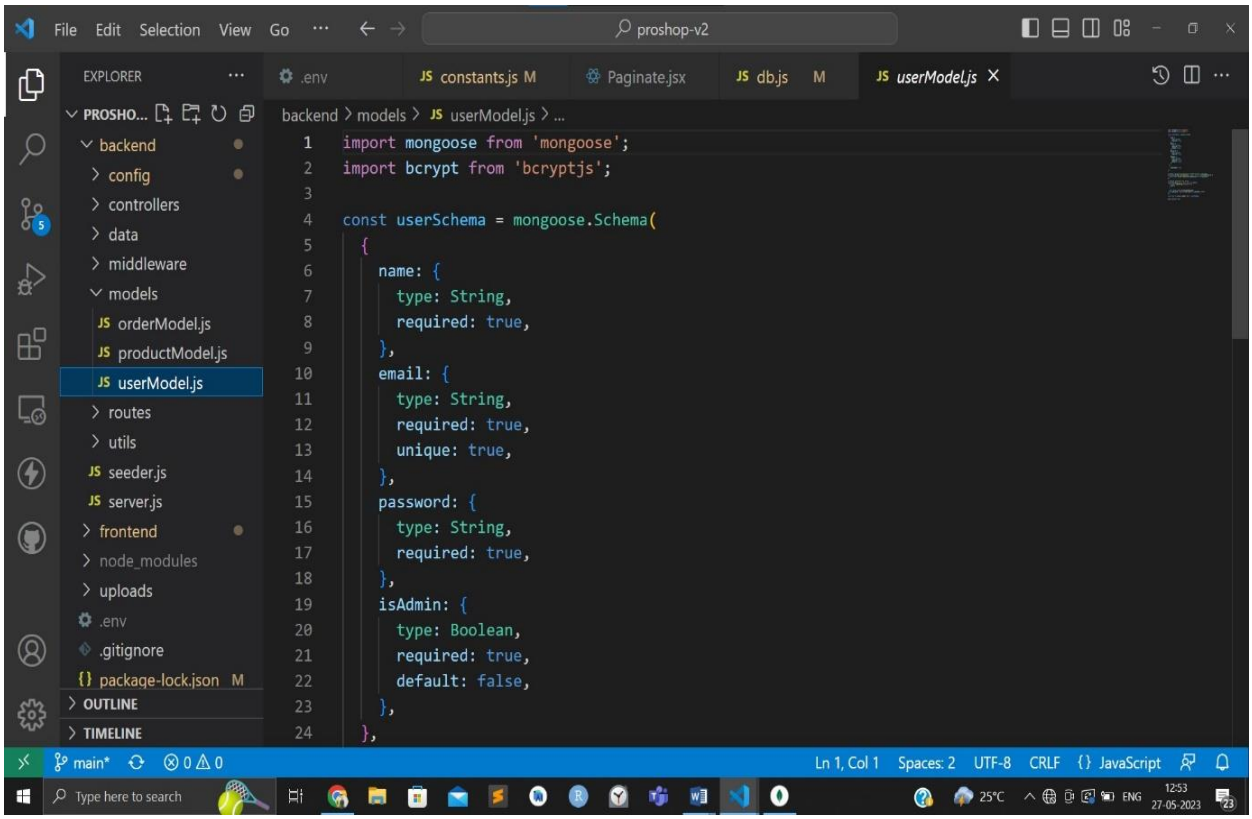
1. Order Model



2. Product Model

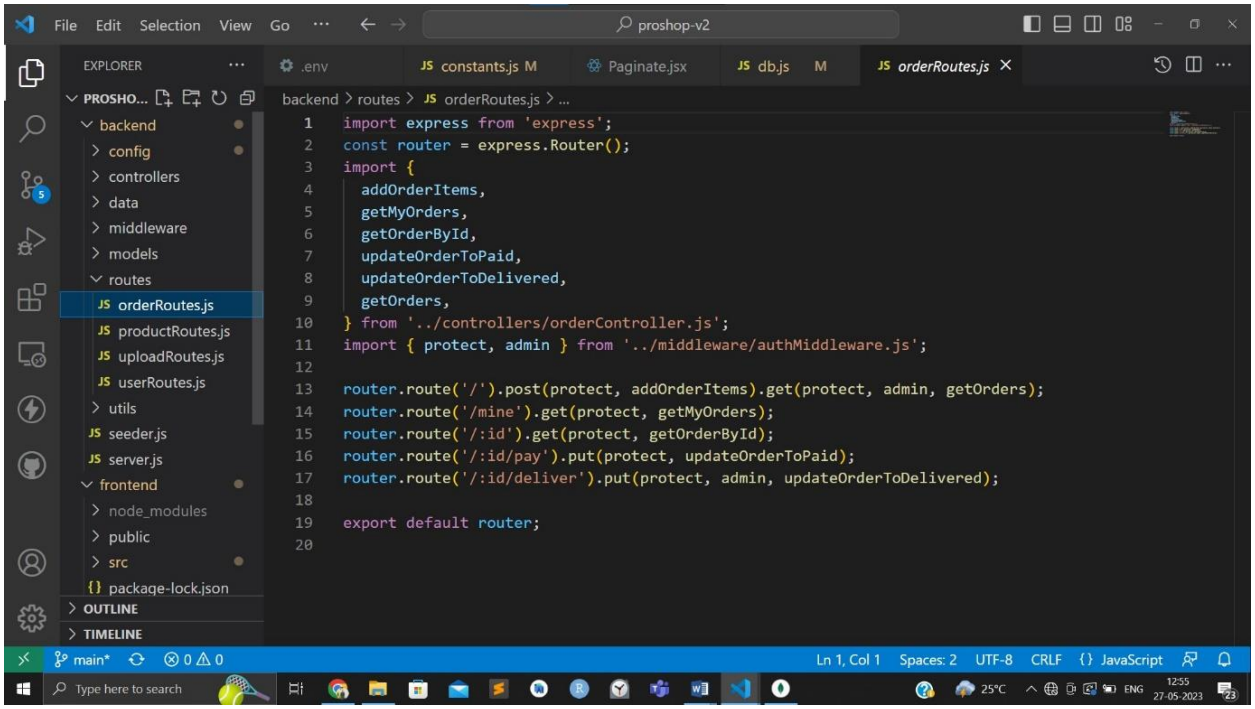


3. User Model

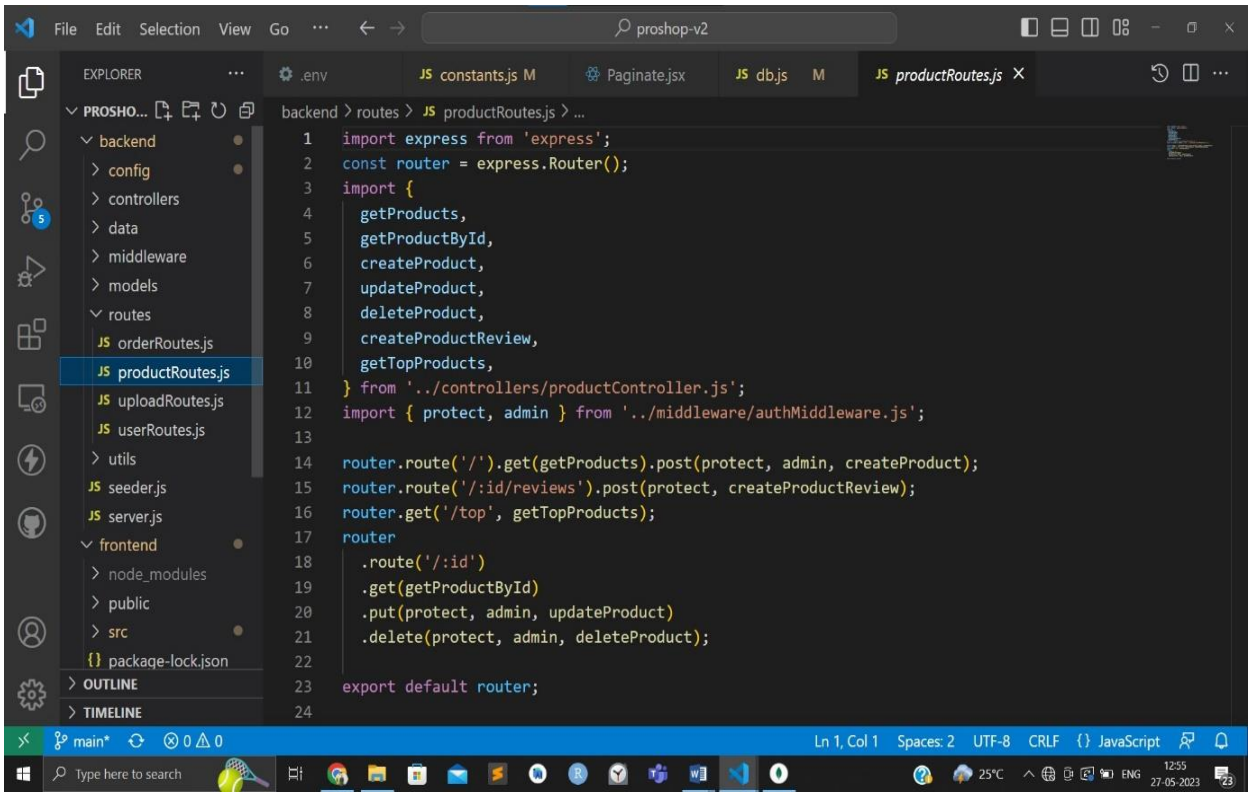

File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
userModel.js
backend > models > userModel.js > ...
PROSHO...
backend
config
controllers
data
middleware
models
orderModel.js
productModel.js
userModel.js
routes
utils
seeder.js
server.js
frontend
node_modules
uploads
.env
.gitignore
package-lock.json M
OUTLINE
TIMELINE
import mongoose from 'mongoose';
import bcrypt from 'bcryptjs';
const userSchema = mongoose.Schema(
{
name: {
type: String,
required: true,
},
email: {
type: String,
required: true,
unique: true,
},
password: {
type: String,
required: true,
},
isAdmin: {
type: Boolean,
required: true,
default: false,
},
},
main*
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
{} JavaScript
Type here to search
25°C
ENG
12:53
27-05-2023

Routes

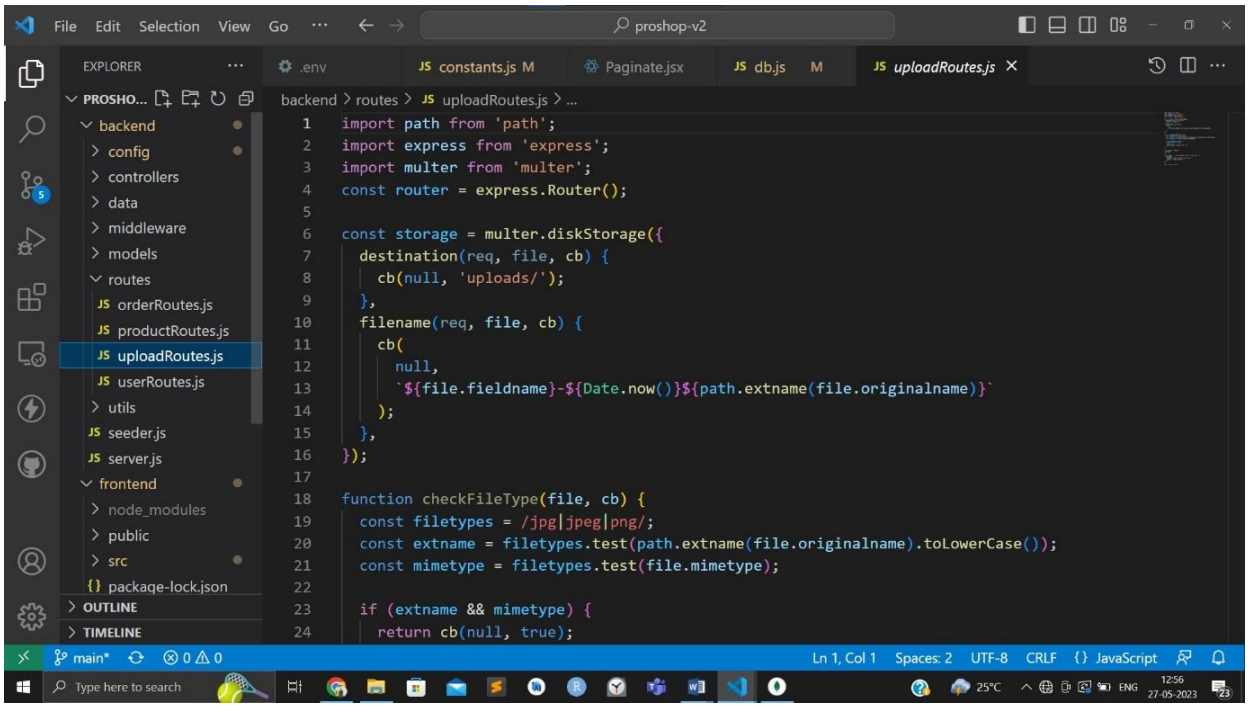
1. Order Routes



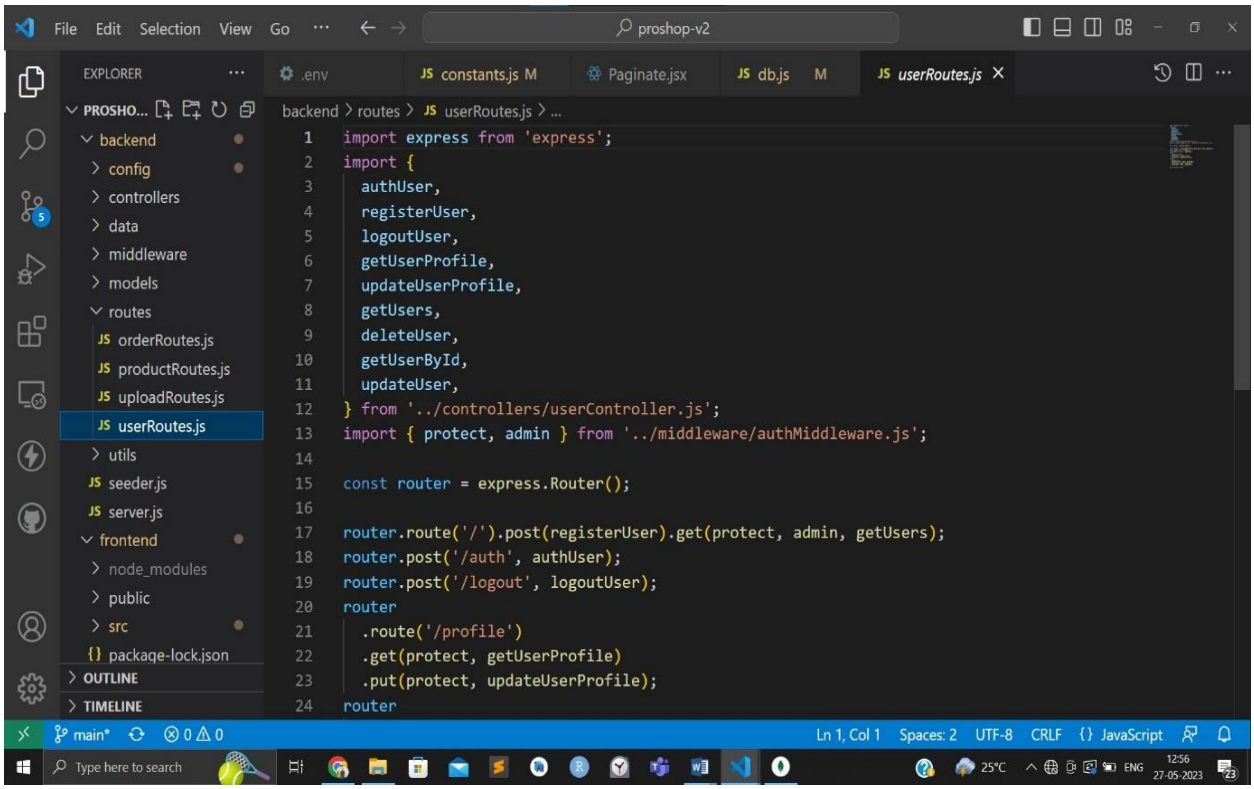
2. Product Routes



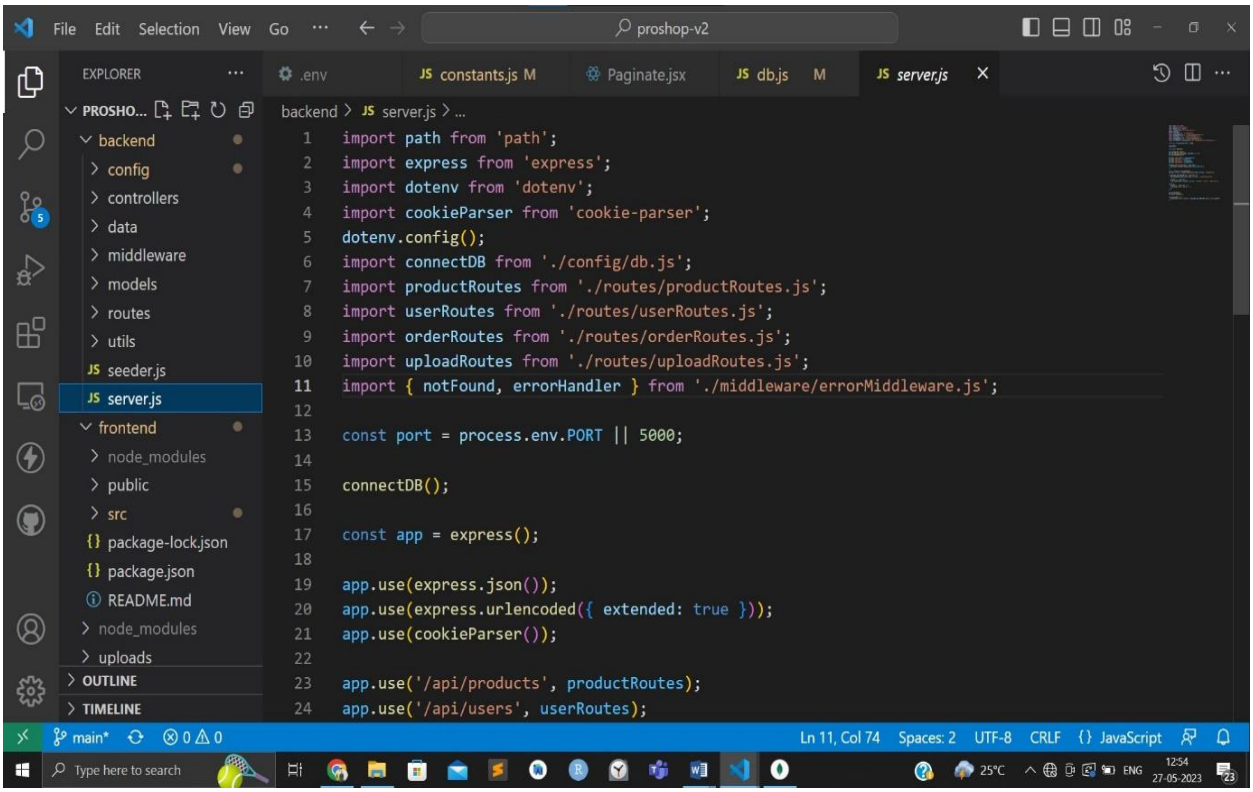
3. Upload Routes



4. User Route


File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
userRoutes.js
backend > routes > userRoutes.js > ...
backend
config
controllers
data
middleware
models
routes
orderRoutes.js
productRoutes.js
uploadRoutes.js
userRoutes.js
utils
seeder.js
server.js
frontend
node_modules
public
src
package-lock.json
OUTLINE
TIMELINE
import express from 'express';
import {
authUser,
registerUser,
logoutUser,
getUserProfile,
updateUserProfile,
getUsers,
deleteUser,
getUserById,
updateUser,
} from '../controllers/userController.js';
import { protect, admin } from '../middleware/authMiddleware.js';
const router = express.Router();
router.route('/').post(registerUser).get(protect, admin, getUsers);
router.post('/auth', authUser);
router.post('/logout', logoutUser);
router
.route('/profile')
.get(protect, getUserProfile)
.put(protect, updateUserProfile);
router
main*
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
JavaScript
Type here to search
25°C
ENG
12:56
27-05-2023

Server.js


File Edit Selection View Go
proshop-v2
EXPLORER
.env
constants.js M
Paginate.jsx
db.js M
server.js
backend > server.js > ...
backend
config
controllers
data
middleware
models
routes
utils
seeder.js
server.js
frontend
node_modules
public
src
package-lock.json
package.json
README.md
node_modules
uploads
OUTLINE
TIMELINE
1 import path from 'path';
2 import express from 'express';
3 import dotenv from 'dotenv';
4 import cookieParser from 'cookie-parser';
5 dotenv.config();
6 import connectDB from './config/db.js';
7 import productRoutes from './routes/productRoutes.js';
8 import userRoutes from './routes/userRoutes.js';
9 import orderRoutes from './routes/orderRoutes.js';
10 import uploadRoutes from './routes/uploadRoutes.js';
11 import { notFound, errorHandler } from './middleware/errorMiddleware.js';
12
13 const port = process.env.PORT || 5000;
14
15 connectDB();
16
17 const app = express();
18
19 app.use(express.json());
20 app.use(express.urlencoded({ extended: true }));
21 app.use(cookieParser());
22
23 app.use('/api/products', productRoutes);
24 app.use('/api/users', userRoutes);
main*
Ln 11, Col 74
Spaces: 2
UTF-8
CRLF
JavaScript
Type here to search
25°C
ENG
12:54
27-05-2023

CHAPTER 6

TESTING

6.1 Introduction

As per ANSI/IEEE 1059, Testing in Software Engineering is a process of evaluating a software product to find whether the current software product meets the required conditions or not. The testing process involves evaluating the features of the software product for requirements in terms of any missing requirements, bugs or errors, security, reliability and performance.

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Testing is an integral part of any software website project, ensuring its quality, reliability, and user satisfaction. It involves a systematic process of evaluating various components and functionalities to identify defects, errors, and potential issues. By conducting thorough testing, developers can identify and rectify bugs before the website is launched, thus minimizing the risk of negative user experiences.

The testing process typically starts with the creation of test cases that outline the expected behavior and requirements of the website. These test cases are designed to cover different scenarios and functionalities, such as user registration, login, content display, data submission, and navigation. The website is then subjected to different types of testing, including unit testing, integration testing, system testing, and acceptance testing.

Unit testing focuses on verifying the functionality of individual components or modules, ensuring that each one works as intended. Integration testing ensures that different components can work together seamlessly. System testing evaluates the entire website's performance and functionality, simulating real-world usage scenarios. Lastly, acceptance testing involves verifying whether the website meets the specified requirements and is ready for deployment.

Throughout the testing process, various techniques and tools are used to identify and track defects, including manual testing, automated testing, and bug tracking systems. Testers closely analyze test results, report any issues, and collaborate with developers to resolve them.

By implementing a comprehensive testing strategy, software website projects can deliver a high-quality product that meets user expectations, enhances user experience, and builds trust among users. Testing plays a crucial role in ensuring a successful launch and continued performance of the website.

6.1.1 Types of Software Testing

Typically Testing is classified into three categories-:

- Functional Testing
- Non-Functional Testing or Performance Testing
- Maintenance (Regression and Maintenance)

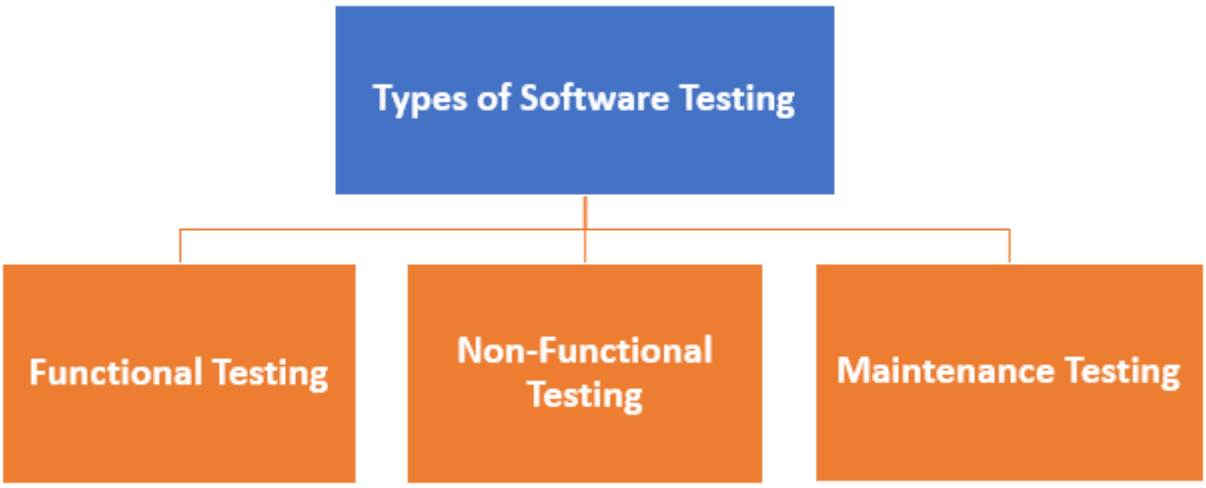


Figure 6.1 Types of Testing

6.1.2 What are the benefits of Software Testing?

Here are the benefits of using software testing:

- Cost-Effective: It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
- Security: It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
- Product quality: It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
- Customer Satisfaction: The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.
- Testing helps in identifying defects or bugs in the software website project. Testing validates and verifies whether the software website meets the specified requirements and functions as intended.

6.1.3 What is a Test Case?

A Test Case is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, pre-condition, post-condition developed for specific test scenario to verify any requirement. The test case includes specific variables or conditions, using which a testing engineer can compare expected and actual results to determine whether a software product is functioning as per the requirements of the customer.

A test case is a detailed specification of inputs, execution conditions, expected outcomes, and test procedures designed to validate a specific functionality or aspect of a software website. It serves as a set of instructions or guidelines for testers to follow during the testing process.

A test case typically consists of several components. Firstly, it includes the preconditions, which outline the necessary setup or state required for the test to be executed. Secondly, it specifies the input values or actions that need to be provided to the system under test. These inputs can range from user interactions to data inputs or system configurations.

Next, the test case defines the expected outcomes or results based on the specified inputs. It describes the anticipated behavior or response of the website, such as correct data display, error messages, or successful completion of a task. Additionally, the test case includes the steps or procedures that testers should follow to execute the test accurately.

Test cases should be comprehensive, covering different scenarios and edge cases to ensure maximum coverage. They should be clear, concise, and unambiguous, enabling testers to perform the test consistently and accurately. Test cases play a crucial role in ensuring the thoroughness and effectiveness of the testing process, aiding in the identification of defects and validating the website's functionality.

6.2 TEST CASES

Introduction to Test Cases in an E-commerce Website:

In an e-commerce website, test cases are crucial for ensuring that the platform functions as expected and delivers a seamless and satisfactory user experience. Test cases provide a systematic approach to verify the various features, functionalities, and workflows of the website. They help in identifying defects, ensuring proper integration between different components, and validating the website's performance. Test cases in an e-commerce website cover a wide range of scenarios, including user interactions, payment processing, product listings, search functionality, shopping cart management, and order fulfillment.

Types of Test Cases for an E-commerce Website:

1. User Registration and Login: Test cases should cover the registration process, including validations for required fields, email verification, and password strength. Login test cases can include scenarios like valid credentials, incorrect passwords, and account lockout after multiple failed attempts.

2. Product Search and Listing: Test cases should ensure accurate search results based on keywords, filters, and categories. Additionally, test cases can cover product sorting, pagination, and accurate display of product details, images, and pricing.

3. Shopping Cart Management: Test cases should include adding products to the cart, quantity updates, removing items, and verifying the correct calculation of subtotal, taxes, and discounts. Additionally, test cases can cover scenarios like empty cart, out-of-stock items, and cart persistence across sessions.

4. Checkout and Payment: Test cases should validate the checkout process, including entering shipping and billing information, selecting payment methods, applying coupons or vouchers, and verifying order total calculation. Payment gateway integration should be thoroughly tested using valid and invalid payment details.

5. Order Management: Test cases should cover scenarios like order placement, order confirmation emails, order status updates, cancellation or modification of orders, and viewing order history.

6. User Reviews and Ratings: Test cases should verify the functionality of submitting and displaying user reviews and ratings for products. Test cases should cover scenarios like validating the review form, verifying the display of reviews, and rating calculation.

7. Security and Performance: Test cases should ensure that the website handles user data securely, including encryption of sensitive information, secure transmission, and protection against common vulnerabilities. Performance test cases can include load testing, stress testing, and scalability testing to verify the website's ability to handle high traffic and transaction volumes.

8. Mobile Responsiveness: Test cases should cover different screen sizes, devices, and orientations to ensure a consistent and user-friendly experience across mobile devices.

It is important to generate a comprehensive set of test cases covering these areas to ensure the quality, functionality, and reliability of an e-commerce website. The test cases should be designed to simulate real-world scenarios and user interactions, ensuring that the website meets the expectations of users and performs optimally in a production environment.

6.2.1 TEST CASES FOR HOME PAGE:

- Verify that home page is displayed after login or not.
- Verify that User name is displayed on homepage or not.
- Verify that featured products are present on home page or not.
- Verify that Search functionality is present on home page or not.
- Verify the home page of application on different browsers.
- Verify the alignment on the home page.
- Verify that products displayed on home page are clickable or not.
- Verify that Products displayed on home page are categorized or not.

6.2.2 TEST CASES FOR PRODUCT SEARCH FUNCTIONALITY:

- Verify that the search field accepts alphabets numbers or symbols.
- Verify that after entering search text and clicking on search icon, the search should work.
- Verify that the search results should be as per the search query.
- Verify that user should be able to search based on product name, brand name or product specification.
- Verify that filter should be present for filtering the search results bases on Brand, Price, reviews or ratings

6.2.3 TEST CASES FOR PRODUCT DETAILS PAGE:

- Verify that the images of product are displayed correctly or not.
- Verify that the price of product is displayed or not.
- Verify that product reviews are mentioned or not.

6.2.4 TEST CASES FOR CART PAGE:

- Verify that when user clicks on Add to Cart, then the product should be moved to cart.
- Verify that user is able to continue shopping after adding items to cart.
- Verify that the quantity of item should be incremented if user adds same item in cart again.
- Verify that the items in cart should be present if user logs out and logs in again.

CHAPTER 7

CONCLUSION

In conclusion, an e-commerce website built using the MERN (MongoDB, Express.js, React.js, and Node.js) stack holds significant promise and potential for the future of online commerce. The MERN stack offers scalability, performance, and responsiveness, enabling businesses to handle increasing volumes of traffic and transactions. Additionally, the mobile compatibility of MERN-based websites ensures accessibility and convenience for users, tapping into the growing market of mobile commerce.

Moreover, MERN-based e-commerce websites can leverage advanced technologies like artificial intelligence and machine learning to personalize the shopping experience and provide tailored recommendations, enhancing customer satisfaction and conversion rates. The stack's robust security measures instill trust in customers and protect their sensitive information, crucial in an era where cyber security is a top concern.

The future of MERN-based e-commerce websites also lies in their ability to integrate with emerging technologies such as virtual reality, augmented reality, and the Internet of Things. By embracing these innovations, businesses can create immersive shopping experiences and capitalize on the evolving demands of tech-savvy consumers.

Furthermore, MERN-based e-commerce websites have the potential to expand globally, thanks to localization features and multiple payment gateway integrations. By

catering to diverse markets and customer segments, businesses can unlock new opportunities and drive growth.

To succeed in the dynamic e-commerce landscape, it is vital for MERN-based websites to stay abreast of technological advancements, continuously adapt to changing customer expectations, and deliver exceptional user experiences. By doing so, businesses can harness the full potential of the MERN stack and create thriving e-commerce platforms that meet the demands of the future.

In conclusion, an e-commerce website plays a crucial role in the digital landscape, providing a platform for online businesses to showcase and sell their products or services to customers worldwide. It offers convenience, accessibility, and a wide range of options for consumers, making it an essential part of modern-day commerce.

A well-designed and properly functioning e-commerce website is essential for the success of an online business. It should provide a seamless user experience, ensuring easy navigation, intuitive interfaces, and efficient transaction processes. Additionally, the website should be visually appealing, engaging, and responsive across various devices.

Security is paramount in an e-commerce website. It should incorporate robust measures to protect customer information, including encryption, secure payment gateways, and adherence to privacy regulations. Trust is crucial in online transactions, and a secure website builds confidence among customers, leading to increased conversions and repeat business.

Effective search functionality, accurate product listings, and comprehensive product information are vital for an e-commerce website. Customers should be able to find desired products easily, view relevant details, and make informed purchasing decisions. Additionally, features like product reviews and ratings contribute to customer trust and aid in the decision-making process.

Smooth and seamless checkout and payment processes are key to minimizing cart abandonment. An e-commerce website should offer multiple payment options, address validation, and clear shipping and return policies. Order management features, including

order tracking, order history, and personalized customer accounts, enhance the overall customer experience.

Continuous testing and monitoring of an e-commerce website are necessary to ensure its functionality, performance, and reliability. Test cases should cover various scenarios, including user registration, product search, shopping cart management, checkout, payment processing, and security. Regular updates and maintenance are crucial to address any bugs or issues, improve performance, and add new features to meet evolving customer expectations.

In today's competitive online marketplace, an e-commerce website serves as the face of a business. It reflects the brand identity, values, and customer-centric approach. By delivering an exceptional user experience, maintaining security, and continuously enhancing features, an e-commerce website can attract and retain customers, drive sales, and contribute to the growth and success of an online business.

CHAPTER 8

FUTURE SCOPE

The future scope of a MERN (MongoDB, Express.js, React.js, and Node.js) e-commerce website is highly promising and lucrative. As technology continues to advance and online shopping becomes increasingly popular, the demand for efficient and user-friendly e-commerce platforms is expected to grow exponentially. Here are some key aspects that highlight the future potential of MERN-based e-commerce websites:

1. Scalability and Performance: MERN stack offers scalability, allowing e-commerce websites to handle large amounts of traffic and transactions. With Node.js as the backend, the website can efficiently handle concurrent user requests, ensuring a seamless shopping experience even during peak times. Additionally, React.js provides a fast and interactive user interface, enhancing performance and customer satisfaction.

2. Mobile Commerce (M-commerce): Mobile devices have become an integral part of our lives, and the future of e-commerce heavily relies on mobile-friendly platforms. MERN stack allows for the development of responsive and mobile-compatible websites, enabling users to shop conveniently on their smartphones or tablets. With the continuous growth of mobile usage, MERN-based e-commerce websites are well-positioned to tap into the expanding market of M-commerce.

3. Personalization and Recommendation Systems: E-commerce websites are increasing leveraging artificial intelligence and machine learning techniques to provide personalized shopping experiences. By integrating recommendation systems and

customer behavior analytics, MERN-based e-commerce websites can offer tailored product suggestions and promotions based on user preferences, leading to higher conversion rates and customer satisfaction.

4. Enhanced Security Measures: As the number of online transactions rises, the need for robust security measures becomes paramount. MERN stack provides a secure environment for e-commerce websites by leveraging best practices in authentication, encryption, and data protection. This instills trust in customers and ensures the confidentiality of their personal and financial information.

5. Integration with Emerging Technologies: MERN-based e-commerce websites have the potential to integrate with emerging technologies such as virtual reality (VR), augmented reality (AR), and Internet of Things (IOT). These technologies can enhance the shopping experience by allowing customers to visualize products in a virtual environment or enabling seamless connectivity between devices for a smart shopping experience.

6. Global Reach and Market Expansion: The internet has made it possible for businesses to reach a global audience. MERN-based e-commerce websites can leverage localization features and multiple payment gateway integrations to expand their reach beyond borders. With proper localization strategies and language support, these websites can tap into new markets and cater to diverse customer segments worldwide.

So we can say that, the future of MERN-based e-commerce websites is bright and filled with opportunities. With their scalability, performance, mobile compatibility, personalization capabilities, security measures, integration with emerging technologies, and global reach, these websites are well-positioned to capitalize on the ever-growing online shopping trend. By staying up to date with the latest technological advancements and continuously adapting to changing customer expectations, MERN-based e-commerce websites can thrive in the dynamic and competitive e-commerce landscape.

REFERENCES

E-commerce Definition – What is E-commerce? [Internet]. Shopify.com. Available from: https://www.shopify.com/encyclopedia/what-is-ecommerce

2. Advantages of E-commerce [Internet]. Thebalancesmb.com 2019 [cited 20 November 2019]. Available from: https://www.thebalancesmb.com/ecommercepros-and-cons-1141609

3. JavaScript [Internet]. Mozilla.org. Available from: https://developer.mozilla.org/en-US/docs/Web/JavaScript

4. NodeJS Introduction [Internet]. Tutorialspoint.com. Available from: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

5. NodeJS Pros and Cons [Internet]. Mindinventory.com. Available from: https://www.mindinventory.com/blog/pros-and-cons-of-node-js-web-app-development/

6. NodeJS use cases [Internet]. Credencys.com. Available from: https://www.credencys.com/blog/node-js-development-use-cases/

7. Express.js Introduction [Internet]. Mozilla.org. Available from: https://developer.mozilla.org/en-US/docs/Learn/Server side/Express_Nodejs/Introduction

8. MongoDB [Internet]. Mongodb.com. Available from: https://docs.mongodb.com/manual/introduction/

BIBLIOGRAPHY

1. **https://www.google.co.in/ https://reactjs.org/**
2. **https://reactjs.org/tutorial/tutorial.html**
3. **https://www.w3schools.com/css/**
4. **https://developer.mozilla.org/en-US/docs/Web/CSS**
5. **DXARIsAIJx01kaPQIdaB5Hzc51o70jqj6PXYY7-dG75LXtpqLH91akvoXoPveArVEaAqtCEALw_w cB&gclsrc=aw.ds**
6. **https://webandcrafts.com/blog/scope-of-ecommerce/**
7. **https://astischool.com/technology-and-computing/feasibility-study- of-online-shopping-article/**
8. NodeJS Introduction [Internet]. Tutorialspoint.com. Available from:
9. https://www.tutorialspoint.com/nodejs/nodejs_introduction.html
10. JavaScript [Internet]. Mozilla.org. Available from: https://developer.mozilla.org/en-US/docs/Web/JavaScript
11. E-commerce Application using MERN stack by Quang Nhat Mai
12. MongoDB [Internet]. Mongodb.com. Available from:
13. https://docs.mongodb.com/manual/introduction/
14. https://reactjsexample.com/e-commerce-website-using-the-mern-stack/
15. https://github.com/topics/mern-ecommerce
16. https://www.google.co.in/
17. https://reactjs.org/
18. https://www.w3schools.com/css/
19. https://developer.mozilla.org/en-US/docs/Web/CSS
20. https://www.youtube.com/watch?v=1rc2zYqexLI
21. https://www.youtube.com/watch?v=HseGVOM85W4