

# **E-SHOP EXPRESS**

**A PROJECT REPORT  
for  
Project (KCA-451)  
Session (2023-24)**

**Submitted by**

**SHASHIRANJAN JHA  
(2200290140141)  
SHARAD NIRANKARI  
(2200290140140)**

**Submitted in partial fulfilment of the  
Requirements for the Degree of**

## **MASTER OF COMPUTER APPLICATION**

**Under the Supervision of  
Dr. Shashank Bhardwaj  
(Associate Professor)**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS  
KIET Group of Institutions, Ghaziabad  
Uttar Pradesh-201206**

**(MAY 2024)**

# CERTIFICATE

Certified that **Shashiranjana Jha 2200290140141, Sharad Nirankari 2200290140140** have carried out the project work having “**eShopExpress**” (**Project-KCA451**) for **Master of Computer Application** from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Shashiranjana Jha (2200290140141)**

**Sharad Nirankari (2200290140140)**

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

**Date:**

**Dr. Shashank Bhardwaj**  
**Associate Professor**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

**Dr. Arun Tripathi**  
**Head**  
**Department of Computer Applications**  
**KIET Group of Institutions, Ghaziabad**

## **E-SHOP EXPRESS**

**SHASHIRANJAN JHA**

**SHARAD NIRANKARI**

### **ABSTRACT**

The advent of e-commerce has revolutionized the retail landscape, offering unprecedented convenience and accessibility to consumers worldwide. Amidst this digital transformation, "**eShopExpress**" emerges as a comprehensive e-commerce platform designed to streamline the online shopping experience.

This project report delves into the conceptualization, development, and implementation of eShopExpress, highlighting its key features, technological architecture, and user interface design. By leveraging cutting-edge technologies such as machine learning algorithms, real-time analytics, and secure payment gateways, eShopExpress aims to enhance user engagement, optimize operational efficiency, and drive business growth. Through meticulous market analysis, user feedback integration, and iterative refinement, eShopExpress embodies a customer-centric approach, ensuring seamless navigation, personalized recommendations, and robust transaction security.

Furthermore, this report explores the challenges encountered during the development phase and proposes strategic recommendations for future enhancements and scalability. Ultimately, eShopExpress represents a paradigm shift in the e-commerce landscape, offering a dynamic platform for businesses to thrive in the digital age.

## ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Dr. Shashank Bhardwaj** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

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

Fortunately, I have many understanding friends, who have helped me a lot on 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 kind 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.

**Shashiranjana Jha**

**(2200290140141)**

**Sharad Nirankari**

**(2200290140140)**

# TABLE OF CONTENTS

	PAGE NO.
<b>Certificate</b>	<b>i</b>
<b>Acknowledgement</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Table of Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vi</b>
<b>1. INTRODUCTION</b>	
1.1. Problem Definition	1
1.2. Project Overview	2
1.3. Theoretical Background	3
1.3.1 The Mern Stack	4
1.3.1.1 MongoDB	4
1.3.1.1 Express	5
1.1.1.1 React	5
1.3.1.4 Node	8
1.3.1.5 Mongoose	9
1.4. Hardware & Software Requirements	10
<b>2. LITERATURE SURVEY</b>	
2.1. Existing System	13
2.2. Proposed System	14
2.3. Feasibility Study	15
2.3.1 Economic Feasibility	15
2.3.1 Technical Feasibility	15
2.3.1 Operational Feasibility	16
2.3.1 Legal Feasibility	16

<b>3.</b>	<b>SYSTEM ANALYSIS DESIGN</b>	
3.1.	Project Specification	17
3.1.1	General Requirements	17
3.1.2	Functional Requirements	17
3.1.3	Security Requirements	18
3.2.	System Architecture	19
3.3.	Design/Diagrams and FlowCharts	20
<b>4.</b>	<b>OUTPUTS &amp; RESULT</b>	
4.1	Home Page	26
4.2	Signup Page	26
4.3	Sign-In Page	27
4.4	Product Page	28
4.5	Cart Page	28
<b>5.</b>	<b>TESTING</b>	
5.1.	Introduction	30
6.1.1	Understanding Testing Methodologies	30
5.2.	Test Case-1: User Registration	30
5.3.	Test Case-2: Adding Items to Cart	30
<b>6.</b>	<b>CONCLUSION</b>	32
<b>7.</b>	<b>REFERENCES</b>	33

## LIST OF FIGURES

<b>Fig. No.</b>	<b>Name of Figure</b>	<b>Page No.</b>
1.1	The architecture of the MERN stack	3
1.2	The architecture design of MongoDB	4
1.3	An example of tree-structure of Document Object Model	6
1.4	The difference between Virtual DOM and real DOM	7
1.5	NodeJS event loop	8
1.6	The relationship between Mongoose, NodeJS and MongoDB	10
3.1	Architecture design	19
3.2	3-Layer Architectural Pattern	19
3.3	Use Case Diagram	20
3.4	Block Diagram	20
3.5	Sequence Diagram	21
3.6	Client Side Activity Diagram	22
3.7	Admin Side Activity Diagram	23
3.8	DataFlow Diagram	24
3.9	FlowChart	25
4.1	HomePage	26
4.2	Sign-In Page	27
4.3	Sign-Up Page	27
4.4	Product Page	28
4.5	Cart Page	29

# CHAPTER 1

## INTRODUCTION

### 1.1 PROBLEM DEFINITION

Nowadays, technology is growing incredibly fast. The rapid innovation of hardware devices makes software technologies advance as well, automatically taking the place of old technologies. Because of the significant expansion in the number of electronic devices that use the Internet and real-time feature, performance is key. By tradition, web development has been carried out by technologies such as JAVA servlets, ASP.NET or PHP. While those technologies are quite widespread and have good features with many years of development and are supported by a large community, they still have some limitations concerning today's need, which is performance. The MERN stack (MongoDB, Express, React and Node) with their simplicity and uniformity, has been recently developed to become a better solution for this performance issue.

The objectives of this thesis were to illustrate and understand the fundamental concepts and usage of each technology in the MERN stack, as well as their compatibilities and advantages as a complete stack in web application development. The thesis achieved that goal by utilizing these modern technologies and implementing a web application. The idea of this web application was for a startup run by the author's parents as they decided to open a book retail store. By researching, the author realized how e-commerce – an enormous platform is emerging at an extraordinary speed over the last decades all over the world and providing more advantages and conveniences as compared to physical stores. E-commerce has changed permanently the way businesses and consumers interact, which allows users to connect with their favorite shops and brands whenever and wherever they want and also helps stores to more actively approach consumers. It is believed that the growth of e-commerce for the next incoming years is increasing beyond measure rate with the release of modern technologies. Understanding this need, the author's solution was to create an e-commerce web application as an online bookstore for the startup to develop its business strategy.

This document structure was organized as follows. The first section brought in the goal of the thesis and the technologies used. Next, essential concepts and theoretical background of each technology in the stack were introduced along with the example, followed by the third section which demonstrated carefully and thoroughly the application development process, from back-end to front-end. In the end, this paper provided a discussion of the project with further improvements and gave a conclusion about the final product.



## 1.2 PROJECT OVERVIEW

The ‘Online E-commerce Web application’ Services department strives to provide solutions to develop and transfer easy and efficient ways in the digital age and to help reduce human pressure and time. To help support shop collections, digital initiatives, and external partner institution digital projects, It provides services that include the digitization of analogue objects, metadata management, digital preservation, and discovery and access of digital collections. “eShopExpress” is a web application written for all operating systems, designed to help users maintain and organize shop virtually. This software is easy to use for both beginners and advanced users. It features a familiar and well-thought-out, attractive user interface, combined with strong searching Insertion and reporting capabilities. The report generation facility of the shop system helps to get a good idea of which are the various items brought by the members, making it users possible to get the product easily.

The ‘Online E-commerce Web application’ Services department strives to provide solutions to develop and transfer easy and efficient way in the digital age and to help reduces human pressure and time. To help support shop collections, digital initiatives, and external partner institution digital projects, It provides services that include the digitization of analogue objects, metadata management, digital preservation, and discovery and access of digital collections.

“eShopExpress” is a web application written for all operating systems, designed to help users maintain and organize shop virtually. This software is easy to use for both beginners and advanced users. It features a familiar and well-thought-out, attractive user interface, combined with strong searching Insertion and reporting capabilities. The report generation facility of the shop system helps to get a good idea of which are the various items brought by the members, making it users possible to get the product easily.

### **The system aims to achieve the following objectives:**

- To design an online e-commerce system.
- To provide a solution to reduce and optimize the expenses of customer order management.
- To create an avenue where people can shop for products online.
- To develop a database to store information on products.

## 1.3 HEORETICAL BACKGROUND

The e-commerce application is constructed based on the MERN stack as the primary full-stack technology. This chapter will briefly discuss each technology in the MERN stack as well as its third-party support library Mongoose to develop the project.

### 1.3.1 MERN Stack

MERN is one of the notable variants based on the MEAN stack. Basically, the MEAN stack was initially established in 2013 by a MongoDB engineering team as a JavaScript-based stack to aid in the development. MEAN comprises four open-source components: MongoDB acts as the database, Express serves as the server framework, Angular as the client framework, and Node works as the environment for running JavaScript. By substituting the popular framework Angular with React – a client-side library – and combining them as the MERN stack, React can become a companion to the other technologies for developing JavaScript and JSON-oriented applications.

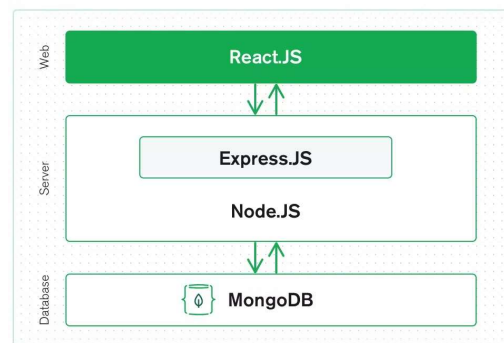


Fig 1.1 The architecture of the MERN stack

As illustrated, the MERN stack is constructed based on the well-known 3-tier architecture, which confirms that the MERN stack is full-stack application development. The MERN stack consists of three components: display layer powered by React, application layer with NodeJS and Express, and database tier provided by MongoDB.

#### 1.3.1.1 MongoDB

MongoDB is a cross-platform, open-source, NoSQL database that is mainly used for scalable large-volume data applications and tasks that do not function well in a relational database. It utilizes a document storage format known as BSON (Binary encoded JavaScript Object Notation). It is a non-relational database management system created by Dwight Merriman, Eliot Horowitz, and Kevin Ryan and became popular in the mid-2000s. MongoDB's design is based on collections and documents, as illustrated in

Figure 1.2 below, which replace the usage of tables and rows in conventional relational databases.

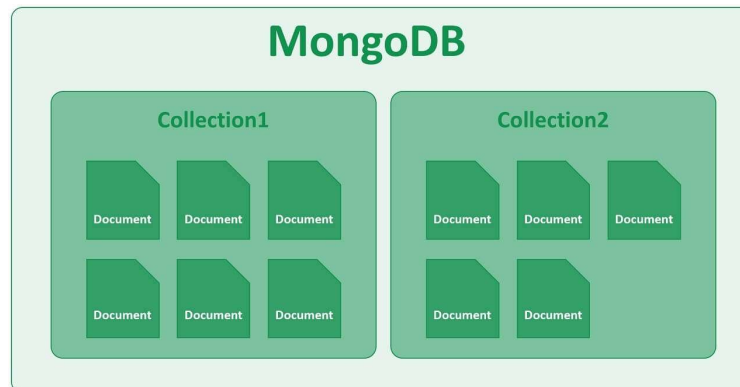


Fig 1.2 The architecture design of MongoDB

MongoDB also supports a variety of document operations, including adding, querying, updating, and deleting. MongoDB is suitable for various use cases due to the diversity of field values and powerful query languages. In addition, its ability to scale out to accommodate larger data volumes horizontally has contributed to its increasing success as the world's most popular NoSQL database.

**There are some crucial features of MongoDB:**

- **Scheme-less Database:** This feature allows a single collection to store numerous documents, each of which contains a varied number of fields, content, and size. Therefore, MongoDB offers tremendous flexibility to databases thanks to this fantastic feature.
- **Document Oriented:** All the data is kept in fields containing a clear structure with key-value pairs rather than rows and columns, which provides more flexibility than RDBMS.
- **Indexing:** Each field in MongoDB documents is indexed using primary and secondary indices, making retrieving, and searching for data more straightforward and faster. Without accurate indices, the database must manually search each document for the matching query, which is time-consuming and inefficient.
- **Scalability:** The horizontal scalability from MongoDB was offered through sharding, which refers to the process of distributing data across numerous servers. By utilizing the shard key, a significant quantity of data is divided into data chunks, and these data chunks are equally spread among shards that involve multiple physical servers. Moreover, it will add new machines to an existing database.

- **Replication:** With the benefit of replication, MongoDB enables high availability and redundancy by producing several copies of data and storing them on a separate server to protect the database against hardware failure, ensuring that the data can be recovered from another if one server fails.
- **Aggregation:** procedures on the dataset are enabled to provide a single or calculated output with three distinct aggregating methods, including pipeline aggregation, map-reduce function, and single-purpose aggregation.

#### 1.3.1.2 ExpressJS

Representing the letter “E” in the MERN stack, Express is a lightweight and versatile web application framework built on top of NodeJS. Thanks to the large community of support, it includes a rich collection of functionalities for developing web and mobile applications. Even though many support packages along with the functionality for better software creation, Express does not affect the performance of NodeJS.

Based on the GitHub repository, Express was established on May 22, 2010, by T.J. Holowaychuk. After that, Strong Loop acquired the project management rights in June 2014 until IBM owned the company in September 2015. Then, in January 2016, the NodeJS Foundation took over the management of Express, and Express is now the primary function of the NodeJS platform.

Express.js is a routing and middleware framework for managing the many routing options for a website. It operates between the request and response cycles. Middleware is invoked after the server receives the request and before the controller actions transmit the response. One or more pieces of middleware are executed to perform tasks, such as authorizing requests or parsing request content. Express applications are composed of a sequence of middleware function calls. Typically, the first middleware executed to process the request initiates the task pipeline. The initial middleware can either complete the request and provide it to the users or call the subsequent middleware to continue the request.

#### 1.3.1.3 ReactJs

React, representing the letter ‘R’ in the MERN stack, focuses on creating the View Layer, which is well-known for all visible parts of the page of an application. React is a multi-purposed, open-source JavaScript library used for building user interfaces based on UI components.

Since React was established to cope with sophisticated, large-scale user interfaces combined with real-time dynamic data and data binding, it has been steadily improving its single-page application development and frontend utilities for programmers of all levels.

- **Javascript Xml**

JSX (JavaScript XML) is a syntactic extension to JavaScript that is similar to HTML. Because JSX optimizes the translation to standard JavaScript and leverages its full power, it is much faster than the regular one. Even though React does not need the usage of JSX to construct React-based applications, it is suggested as it simplifies the development for developers whenever markup components and binding events are required. Rather than dividing markup and logic into different files, JSX enables developers to generate cleaner and more manageable codebases for their websites by combining rendering logic and user interface logic in the same components.

- **Virtual Dom**

Virtual DOM (or VDOM) is an abstract representation of DOM (Document Object Model), and its solutions are constructed on top of the regular DOM. DOM represents the UI of the program, and its model portrays the document as a collection of different nodes and objects to interact with the structure, layout, and content of the website through programming languages.

While regular DOM manipulation is slow due to the re-rendering of both the updated element and its children after getting new data, the Virtual DOM rerenders only the components that need updating, which fastens the rendering process and increases the performance. In order to further understand why Virtual DOM is fast and practical, the functionality of Virtual DOM must be discussed. The state of the DOM tree hierarchy is stored when Virtual DOM is utilized to render a page. Instead of constructing a new tree, a diffing approach is employed when UI modifications are necessary.

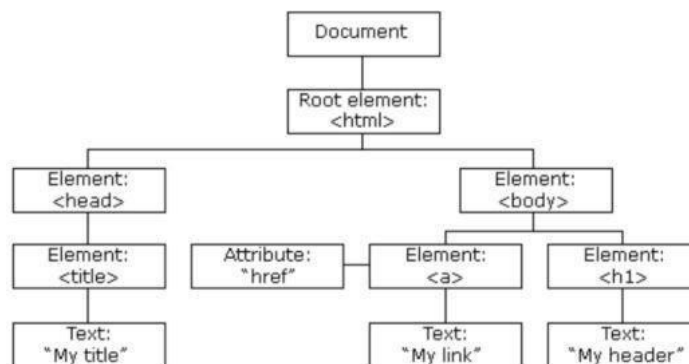


Fig 1.3 An example of tree-structure of Document Object Model

The Virtual DOM tree by comparing the current version to the prior one. This comparison employs a diffing algorithm to reduce the number of DOM operations/refreshes, leading to considerable boosting speed. The whole procedure is titled Reconciliation.

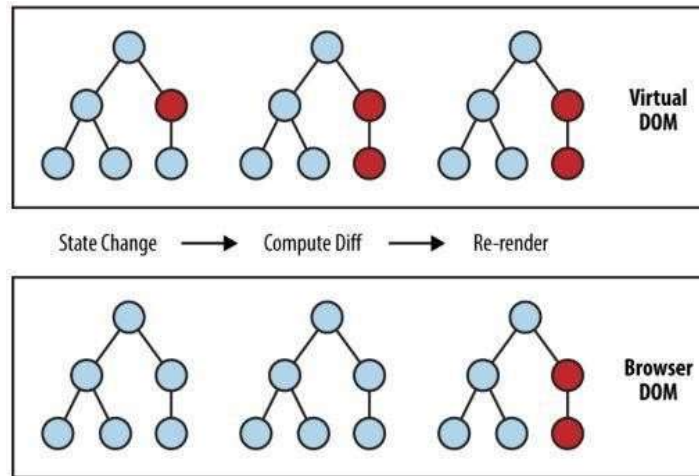


Fig 1.4 The difference between Virtual DOM and real DOM update process

- **Components**

Components are the primary concept of React, which encourages developers to separate complicated user interfaces into reusable and independent parts. React components are classified into two types: class-based components and functional components. The functional component is considered the most straightforward method to construct as it can be implemented as a JavaScript function to return JSX. The class-based component is constructed using the ECMAScript6 class syntax combined with the built-in React library class “Component”.

Components can refer to one another, which means one component can be a parent component containing several child components with no restriction on the amount of complexity. Moreover, both functional and class-based components adhere to one strict rule assigned by React: all React components must be pure functions in which their props are immutable. Props, which stands for Properties, are a collection of inputs passed as parameters to a component, while pure function depicts the state in which the function performs the logic without affecting the arguments. As a result, a React component behaves like a pure function while respecting its inputs and rendering the same result for the same props.

- **Hooks**

Prior to version 16.8, the majority of React components were class-based as class-based components offer life-cycle methods for component state management. Since

version 16.8, however, React has introduced a new notion called Hooks, which provides a new method to leverage state and other React capabilities inside a functional component. By using Hooks, a component's stateful logic can be isolated, tested separately, and reused without affecting the component hierarchy. In addition, Hooks enables the developer to break down a component into separate functions based on the relationship instead of life-cycle methods.

There are two built-in React hooks that should be discussed. First, the State hook, commonly known as the use State hook, enables component-level state management. It hooks into React's state by creating a state variable that React maintains. useState hook accepts and returns two results: the current state and a function to alter it. The component state can be efficiently initialized, utilized, and modified using the useState hook. The second hook is the Effect hook, also known as the use effect hook. useEffect hook assists programmers in managing component life cycles. The difficulty of separating related functionality and data into several class life cycles, such as componentDidMount, componentDidUpdate, and componentWillUnmount, has been thoroughly addressed by Effect Hook. A Reactcomponent can support multiple effects to isolate data manipulation issues.

#### 1.3.1.4 Node

NodeJS is an open-source, cross-platform JavaScript runtime environment designed for constructing scalable applications. NodeJS is independently built on top of Google Chrome's V8 runtime engine, which is well-known for working effectively outside of a browser. By utilizing an event-driven design and operating on a single-thread event loop, NodeJS allows asynchronous and non-blocking I/O optimization to enhance web application performanceand scalability, as shown in Figure 5 below.

Therefore, it provides an alternative approach fordevelopers to wait and fulfill requests for developing lightweight and real-time applications.

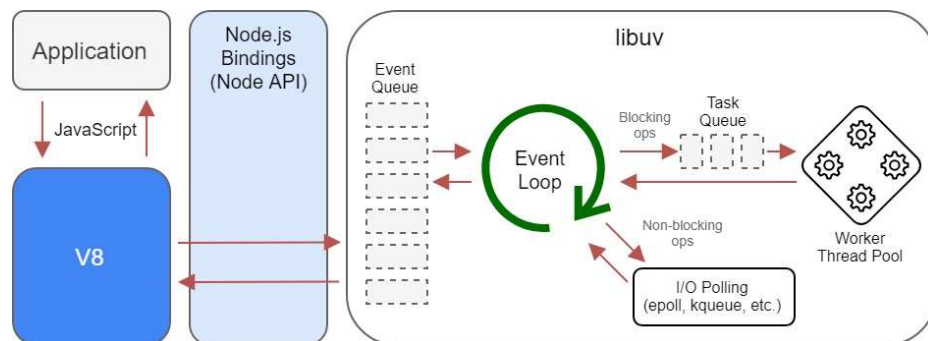


Fig 1.5 NodeJS event loop

- **Node package manager (NPM)**

Node package manager (NPM) is the NodeJS default package manager for applications, and it is utilized to maintain all of the NodeJS packages and modules along with the commandline client npm. Therefore, it facilitates time-consuming manual tasks by automating managing third-party packages, allowing developers to spend more time on the development process.

NPM was first published on January 12, 2010, by Isaac Z. Schlueter. It is installed alongside NodeJS and is utilized to install the required packages and modules in the NodeJS project. It is currently the world's largest software registry, with approximately two million packages at the end of March 2022.

### **1.3.1.5 Mongoose**

Mongoose is an object document mapping (ODM) library that is utilized for facilitating Node and MongoDB development. It is responsible for managing data relationships, performing schema validation, and serving as a middleman between objects in code and object representations in MongoDB. In addition, Mongoose offers multiple methods and functions that effectively facilitate the communication between NodeJS and MongoDB. Figure 6 below illustrates the relationship between Mongoose, NodeJS, and MongoDB.

As shown in Figure 6, Mongoose is utilized to create the interaction between Node and MongoDB through object mapping. After that, Mongoose forms the connection with MongoDB using a Mongo Driver. Therefore, the relationship between Mongoose, NodeJS, and MongoDB ensures data capabilities.

The first step in getting started with Mongoose, like with other ODM libraries, is to create a schema. As Mongoose's documentation page described, a schema specifies the data structure and property casting, along with the following techniques: instance methods, compound indexes, static Model methods, and middlewares. Once the first stage is finished, the developed schemas will be utilized to map to MongoDB collections and shape the data documents included inside each collection. The second stage required by programmers is to construct a Mongoose model. Models are composed of builders of schemas, with the primary responsibility of producing and scanning documents in the Mongo database. Querying, deleting, and updating documents in the database are additional capabilities of models worth mentioning.



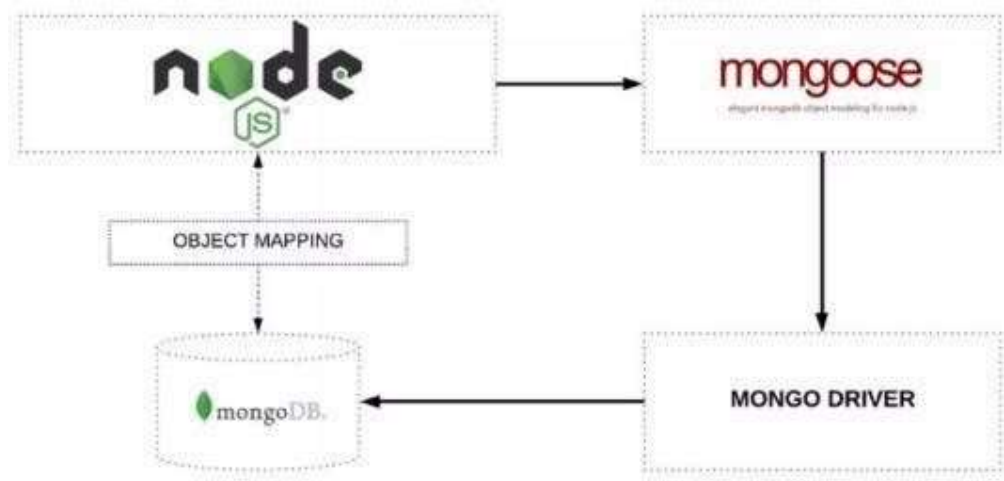


Fig 1.6 The relationship between Mongoose, NodeJS and MongoDB

## 1.4 HARDWARE & SOFTWARE REQUIREMENTS

### 1.4.1 Hardware Specification (Minimum requirement)

- Processor: Any processor above 500 MHz
- RAM: 1GB
- Hard Disk: 80GB
- System: Pentium IV 2.4 GHz
- Internet Connection: Active

### 1.4.2 Software Specification

- Operating System: Any operating system
- Web Browser: Any web browser

Any system with above or higher configuration is compatible for this project.

## CHAPTER 2

### LITERATURE SURVEY

MERN stack is a commonly used combination of technologies for building modern web applications, comprising MongoDB, Express.js, React, and Node.js. E-commerce websites are increasingly important due to the rise of online shopping. This literature review discusses the use of the MERN stack for e-commerce website development, covering its advantages, technical aspects, security considerations, case studies, best practices, and future developments.

Scalability is a key advantage of using the MERN stack for e-commerce website development. The modular architecture allows easy scaling of the application, while the use of MongoDB as a NoSQL database enables flexible data modelling for complex and dynamic data structures. Another advantage is the stack's performance, which is boosted by its ability to handle both client-side and server-side components.

Developing an e-commerce website using the MERN stack involves creating an API with Express.js for server-side communication. MongoDB stores and retrieves data while React builds dynamic and responsive user interfaces. Node.js is used for server-side scripting to enable real-time communication between client-side and server-side components.

Security is a critical aspect of e-commerce website development, and the MERN stack provides several features to ensure the security of the application. MongoDB provides authentication and access control, and SSL/TLS encryption for secure communication. Express.js handles authentication and authorization and can be used to prevent common security vulnerabilities. React and Node.js support SSL/TLS encryption and XSS protection.

Several successful e-commerce websites have been built using the MERN stack, such as Grofers in India and Ubuy in the Middle East.

Best practices for e-commerce website development using the MERN stack include optimizing performance, testing thoroughly for reliability and security, deploying to a reliable and scalable infrastructure, and maintaining the application through regular updates and bug fixes. Additionally, it is important to adhere to industry standards and regulations such as PCI DSS compliance for payment processing.

One of the potential drawbacks of using the MERN stack for e-commerce website development is its learning curve. The technology stack requires expertise in multiple technologies, which can be challenging for novice developers. Additionally, the modularity of the MERN stack can result in a large number of dependencies, which can

increase the complexity of the application and make maintenance more difficult.

Finally, the MERN stack may not be the most appropriate choice for all e-commerce websites, particularly those with simpler requirements.

The future of e-commerce website development using the MERN stack will see the integration of emerging technologies like machine learning, blockchain, and augmented reality. Machine learning can provide personalized product recommendations, blockchain can facilitate secure payment processing, and augmented reality can provide a more immersive shopping experience.

MERN stack offers a flexible and powerful solution for e-commerce website development, with advantages in scalability and performance. Its security features and successful case studies make it a reliable and popular choice for e-commerce developers. Best practices and future developments provide guidance and opportunities for continued innovation

## **2.1 EXISTING SYSTEM**

The present scenario for shopping is to visit the shops and market manually and then from the available product list one needs to choose the item he or she wants and then pay for the same item mainly in cash mode is done, as not every society is well educated and aware to use net banking or card modes or wallets etc.

This system is not very user-friendly as one needs to go to the market physically and then select items only from the available list. So mostly it is difficult to get the product as per our desire. Description About the products is less available and are mostly verbal only. For this type of shopping, one needs to have an ample amount of free time.

Also, not good markets exist everywhere, so often good markets become out of reach for certain people. In the proposed system customers need not go to the shops for purchasing the products.

The current system for shopping is to visit the shop manually and from the available product wants from the item customer wants and buy the item by payment of the price of the item.

- It is less user-friendly.
- User must go to the shop and select products.
- It is difficult to identify the required product.
- Description of the product limited.
- It is a time-consuming process.
- Not in reach of distant users.

## 2.2 PROPOSED SYSTEM

The proposed system for an e-commerce website using the MERN stack will leverage the latest technologies to provide a user-friendly, secure, and scalable e-commerce platform. The MERN stack, comprising MongoDB, Express.js, React, and Node.js, will be used to develop a robust and dynamic e-commerce website that will enable businesses to sell products online.

The proposed system will have a client-server architecture, with the server-side component built using the MERN stack. MongoDB will be used as the database to store and retrieve data, and Express.js will handle the server-side communication and routing. Node.js will be used for server-side scripting, allowing real-time communication between the client-side and server-side components.

The client-side component will be developed using React, which will enable the creation of dynamic and responsive user interfaces. React will also provide an efficient rendering of components, resulting in reduced load times and enhanced user experience. Additionally, the proposed system will be developed using a modular approach, allowing for easy scaling of the application.

The proposed system will have a range of features to provide a seamless e-commerce experience for users. These features will include

- **Product Catalog:** The product catalogue will allow users to browse through a wide range of products and filter them based on their preferences.
- **Shopping Cart:** The shopping cart will enable users to add products to their cart, review their orders, and checkout.
- **Payment Gateway Integration:** The payment gateway integration will allow users to make secure payments using a variety of payment methods, such as credit cards, debit cards, and digital wallets.
- **Order Management:** The order management system will enable businesses to manage orders, track shipments, and generate invoices.
- **User Management:** The user management system will enable businesses to manage user accounts, including registration, login, and authentication.
- **Search Engine Optimization:** The proposed system will be optimized for search engines to improve the visibility of the e-commerce website and increase traffic.

- **SSL/TLS Encryption:** The system will use SSL/TLS encryption to secure all communication between the client-side and server-side components.
- **Access Control:** Access control will be implemented using role-based authentication to ensure that only authorized users have access to sensitive data.
- **PCI DSS Compliance:** The system will be compliant with the Payment Card Industry Data Security Standard (PCI DSS) to ensure the secure processing of credit card payments.
- **Regular Security Audits:** The proposed system will undergo regular security audits to identify and address any vulnerabilities or security issues.

The proposed system for e-commerce websites using the MERN stack will provide businesses with a powerful and flexible platform for selling products online. The system will be developed using the latest technologies and security features to ensure a seamless and secure e-commerce experience for users. The modular architecture of the MERN stack will enable easy scaling of the application, allowing businesses to expand their e-commerce operations as they grow.

## 2.3 FEASIBILITY STUDY

The feasibility study for an e-commerce website using the MERN stack is essential to evaluate the practicality and viability of the proposed system. This study will assess the technical, operational, economic, and legal aspects of the project and provide insights into the feasibility of developing an e-commerce website using the MERN stack.

### 2.3.1 Economic Feasibility:

The economic feasibility of the proposed system is determined by assessing the cost of development, deployment, and maintenance of the e-commerce website. The MERN stack is an open-source technology, and the initial cost of development is relatively low. However, ongoing maintenance and updates will require a dedicated team, which may result in recurring expenses. The project's profitability will depend on the volume of sales generated, and the e-commerce website's scalability and flexibility will enable businesses to expand their operations and generate more significant profits.

### 2.3.2 Technical Feasibility:

The technical feasibility of the proposed system is determined by evaluating the availability of resources and the compatibility of the MERN stack components with the project requirements. The MERN stack has a vast developer community, and there are

numerous resources available for learning and development. Moreover, MongoDB, Express.js, React, and Node.js are compatible with each other and can be integrated seamlessly to create a robust e-commerce website.

### **2.3.3 Operational Feasibility:**

The operational feasibility of the proposed system is evaluated based on the availability of resources, skills, and expertise required for development, deployment, and maintenance. The development team must possess the necessary skills and expertise in the MERN stack to ensure the successful development and deployment of the e-commerce website. Additionally, the team must have experience in managing e-commerce websites, including order management, user management, and inventory management.

### **2.3.4 Legal Feasibility:**

The legal feasibility of the proposed system is evaluated by assessing compliance with data protection laws, intellectual property laws, and other relevant regulations. The e-commerce website must comply with the General Data Protection Regulation (GDPR) and other data protection laws to ensure the security and confidentiality of user data. Additionally, the website must comply with intellectual property laws, such as trademark and copyright laws, to avoid legal disputes.

The feasibility study demonstrates that developing an e-commerce website using the MERN stack is technically and operationally feasible. However, the project's profitability will depend on the volume of sales generated, and businesses must be prepared for recurring expenses associated with maintenance and updates. The legal feasibility of the project will depend on compliance with data protection and intellectual property laws, and businesses must ensure compliance to avoid legal disputes. Overall, the proposed system for an e-commerce website using the MERN stack is a viable and practical solution for businesses looking to sell products online.

## CHAPTER 3

### SYSTEM ANALYSIS DESIGN

#### 3.1 PROJECT SPECIFICATION

The project specification for an e-commerce website using the MERN stack is a comprehensive document that outlines the requirements, functionality, and features of the proposed system. This specification will provide a detailed description of the e-commerce website, including its design, user interface, functionality, security, and performance.

##### 3.1.1 General Requirements:

The e-commerce website must be developed using the MERN stack, which includes MongoDB, Express.js, React, and Node.js. The website should be responsive, intuitive, and easy to navigate.

It should be designed with a clean and modern interface, consistent with the branding and image of the business. The website must support multiple languages and currencies to facilitate global sales.

##### 3.1.2 Functional Requirements:

The e-commerce website must have the following features and functionalities:

- **User Account Management:** The website should allow users to create and manage their accounts, including personal information, shipping addresses, and payment options.
- **Product Catalog:** The website should provide a comprehensive catalogue of products, including product descriptions, images, prices, and reviews. Users should be able to search and filter products based on different criteria, such as price, category, and brand.
- **Shopping Cart:** The website should allow users to add products to their shopping cart, view the contents of their cart, and proceed to checkout.
- **Payment Gateway Integration:** The website should integrate with a secure and reliable payment gateway to facilitate secure online transactions.
- **Order Management:** The website should allow businesses to manage orders, including order tracking, order processing, and order fulfillment.

- **Customer Service:** The website should provide customer service options, such as live chat, email support, and FAQ section.

### 3.1.3 Security Requirements:

The e-commerce website must have the following security features:

- **SSL Encryption:** The website should use SSL encryption to secure user data and prevent unauthorized access.
- **User Authentication:** The website should require users to create and use strong passwords to protect their accounts.
- **Payment Security:** The website should use a secure payment gateway to ensure secure and safe online transactions.
- **Fast Load Time:** The website should load quickly to provide a seamless user experience.
- **Scalability:** The website should be scalable to accommodate growing user traffic and transactions.
- **Reliability:** The website should be reliable and available 24/7 to facilitate sales and transactions.

The project specification for an e-commerce website using the MERN stack provides a detailed overview of the proposed system's requirements, functionalities, and features. The specification highlights the importance of a responsive design, intuitive user interface, secure payment gateway, and fast load time to provide a seamless and satisfactory user experience. By following this specification, businesses can develop a robust e-commerce website using the MERN stack and expand their operations to reach a global audience.



## 3.2 SYSTEM ARCHITECTURE

- **Architecture design**

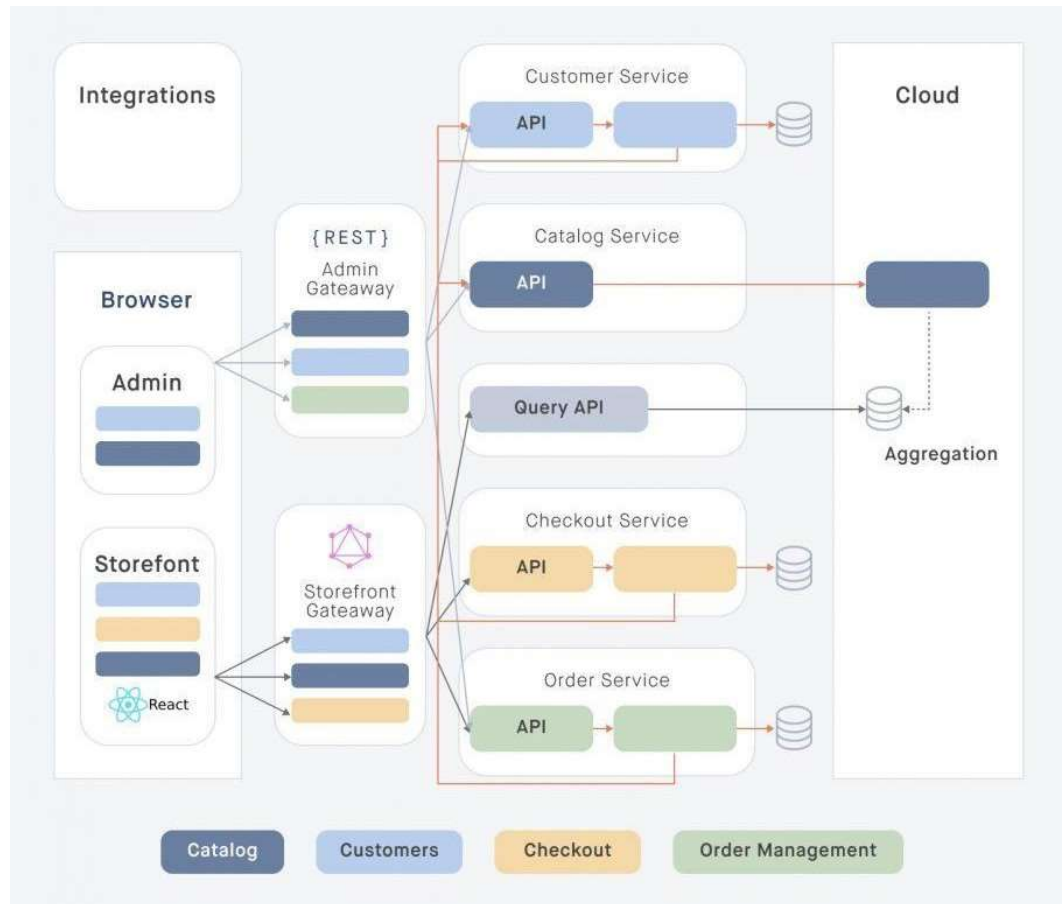


Fig 3.1 Architecture design

- **3-Layer Architectural Patter**

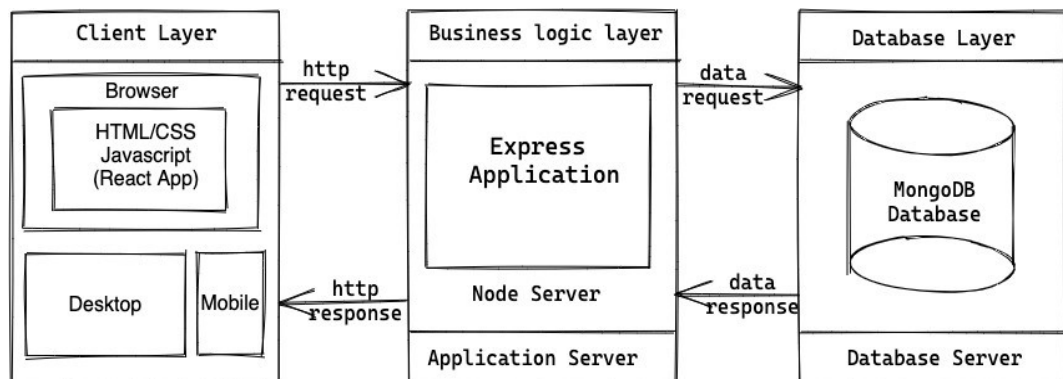


Fig 3.2 3-Layer Architectural Pattern

### 3.3 DESIGN/DIAGRAMS AND FLOWCHARTS

- **Use Case Diagram**

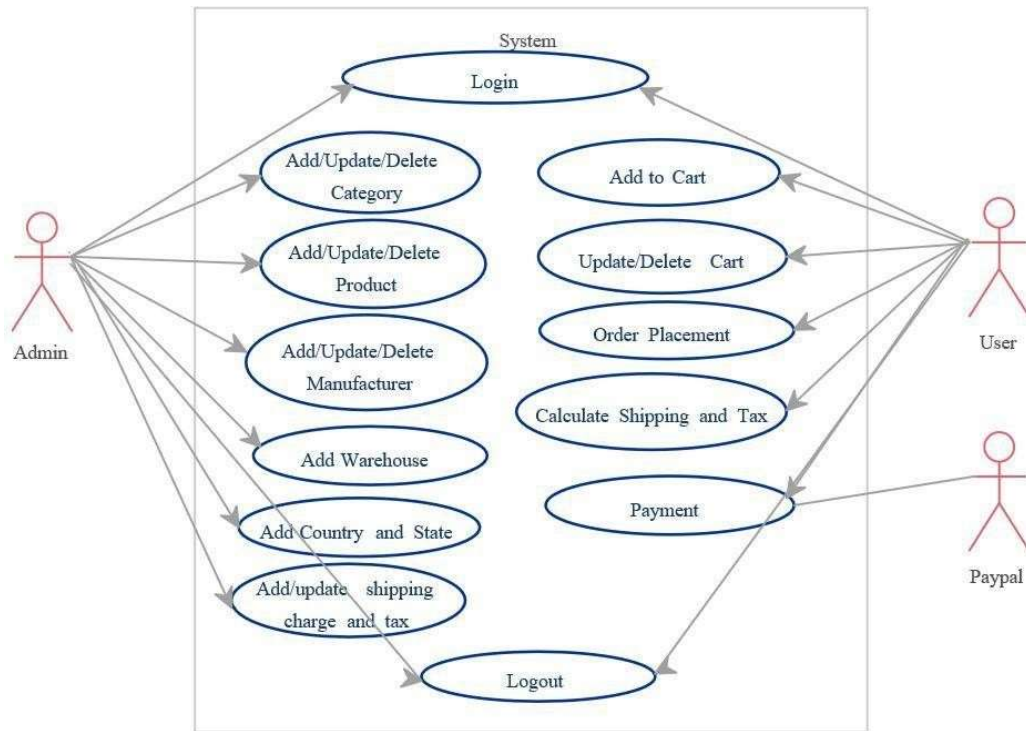


Fig 3.3 Use Case Diagram

- **Block Diagram**

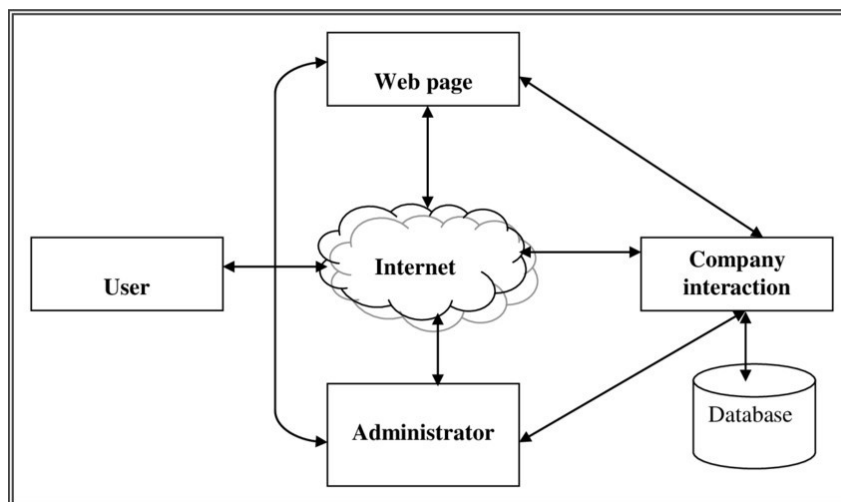


Fig 3.4 Block Diagram

- **Sequence Diagram**

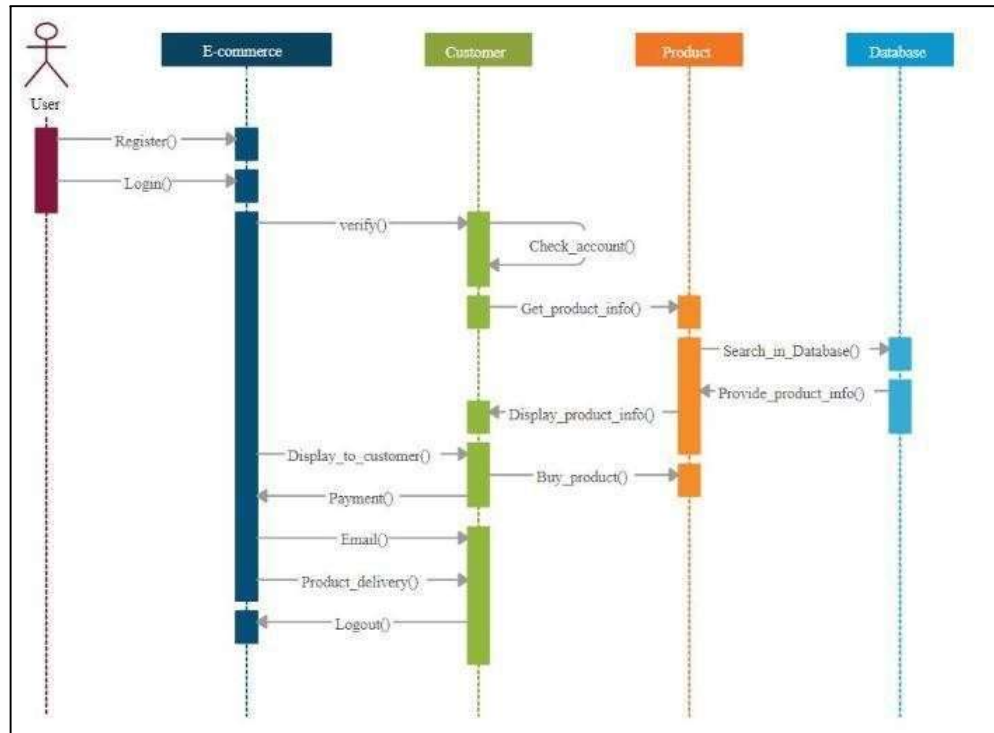


Fig 3.5 Sequence Diagram

- **Client-Side Activity Diagram**

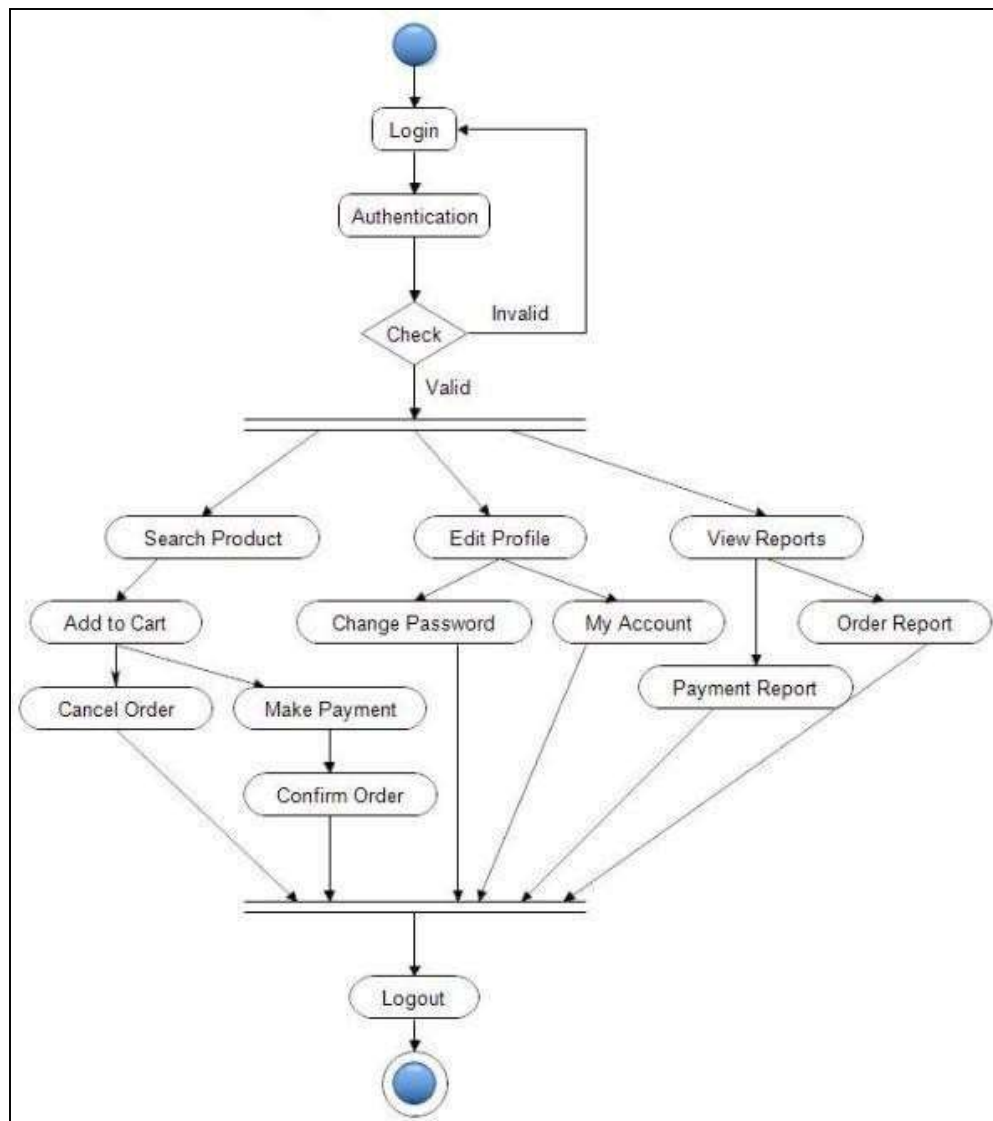


Fig 3.6 Client-Side Activity Diagram

- **Admin-Side Activity Diagram**

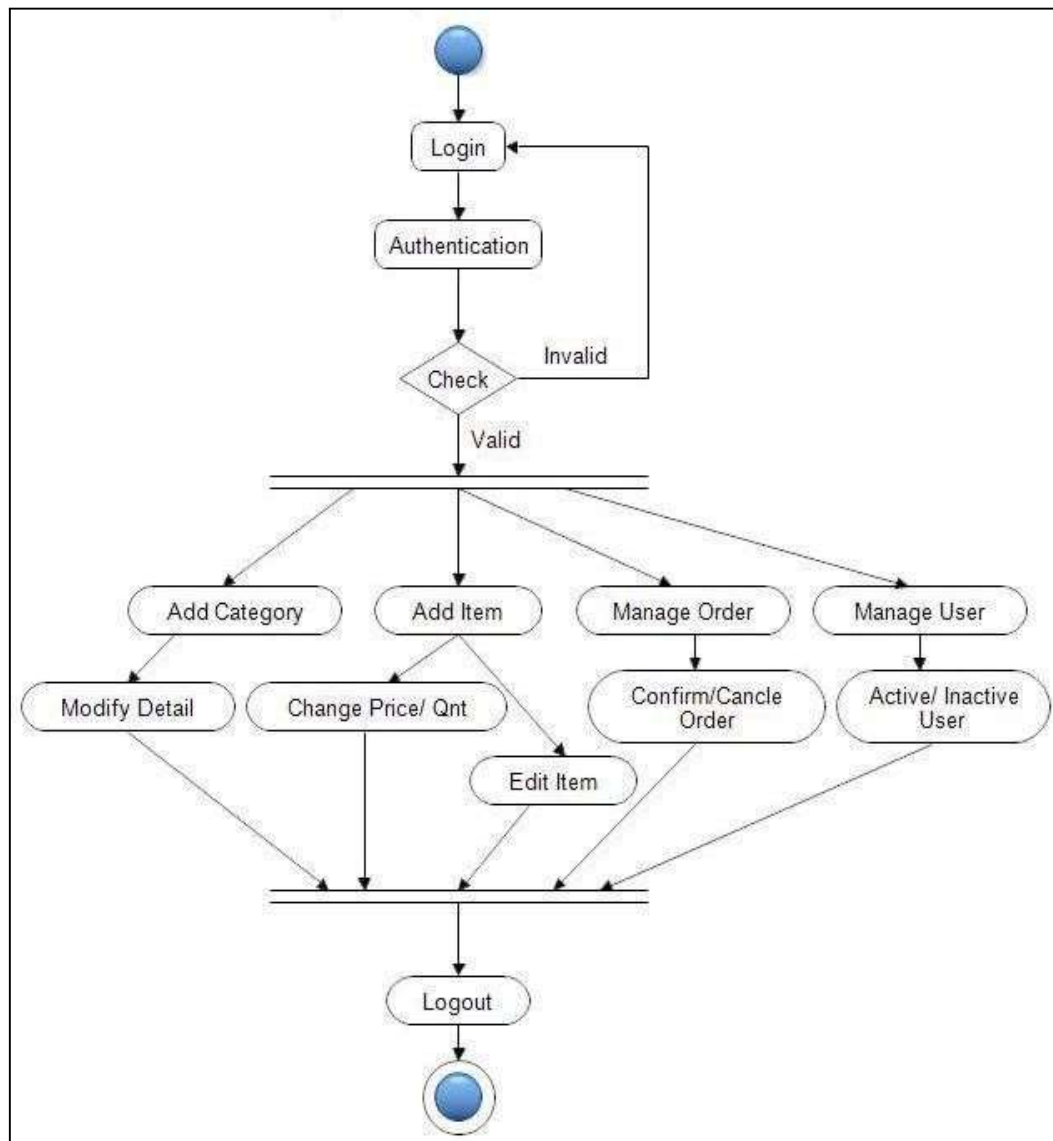


Fig 3.7 Admin Side Activity Diagram

- **Dataflow Diagram**

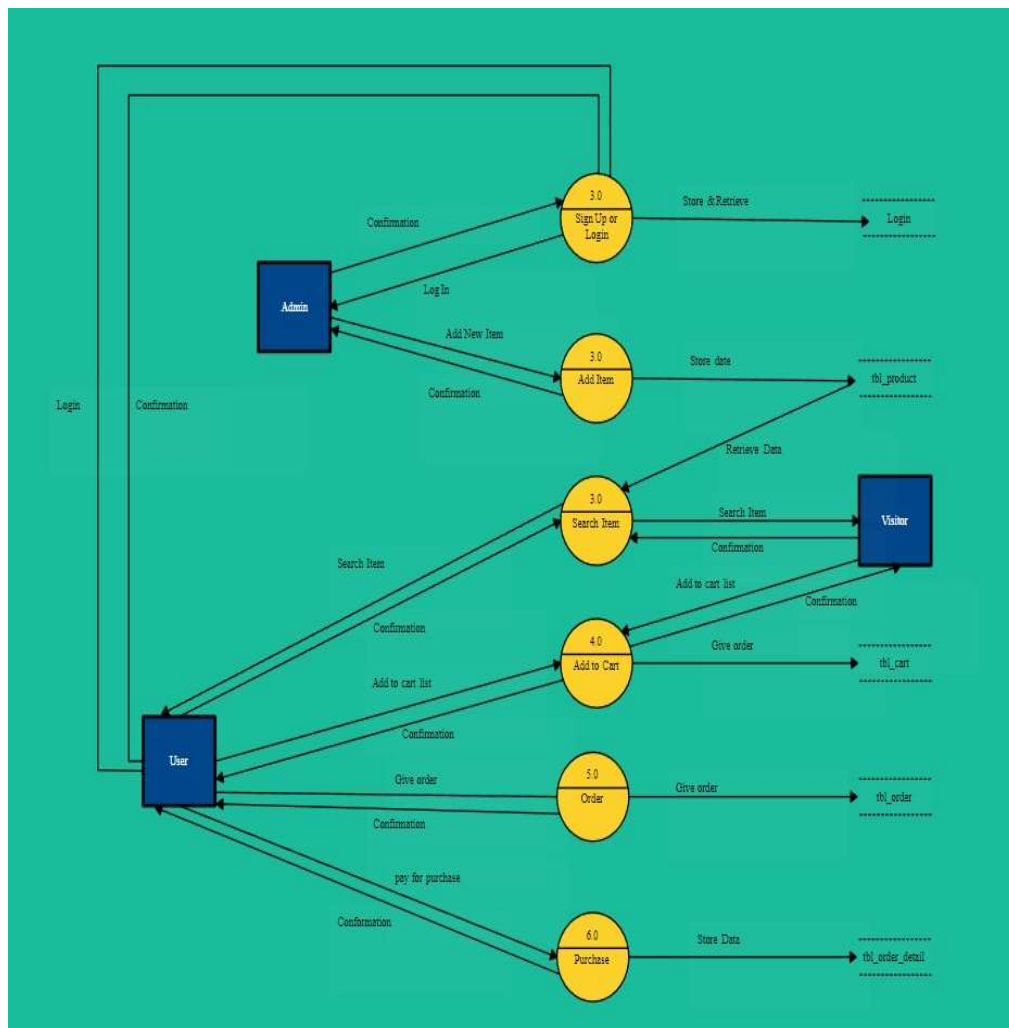


Fig 3.8 Dataflow Diagram

- Flowchart

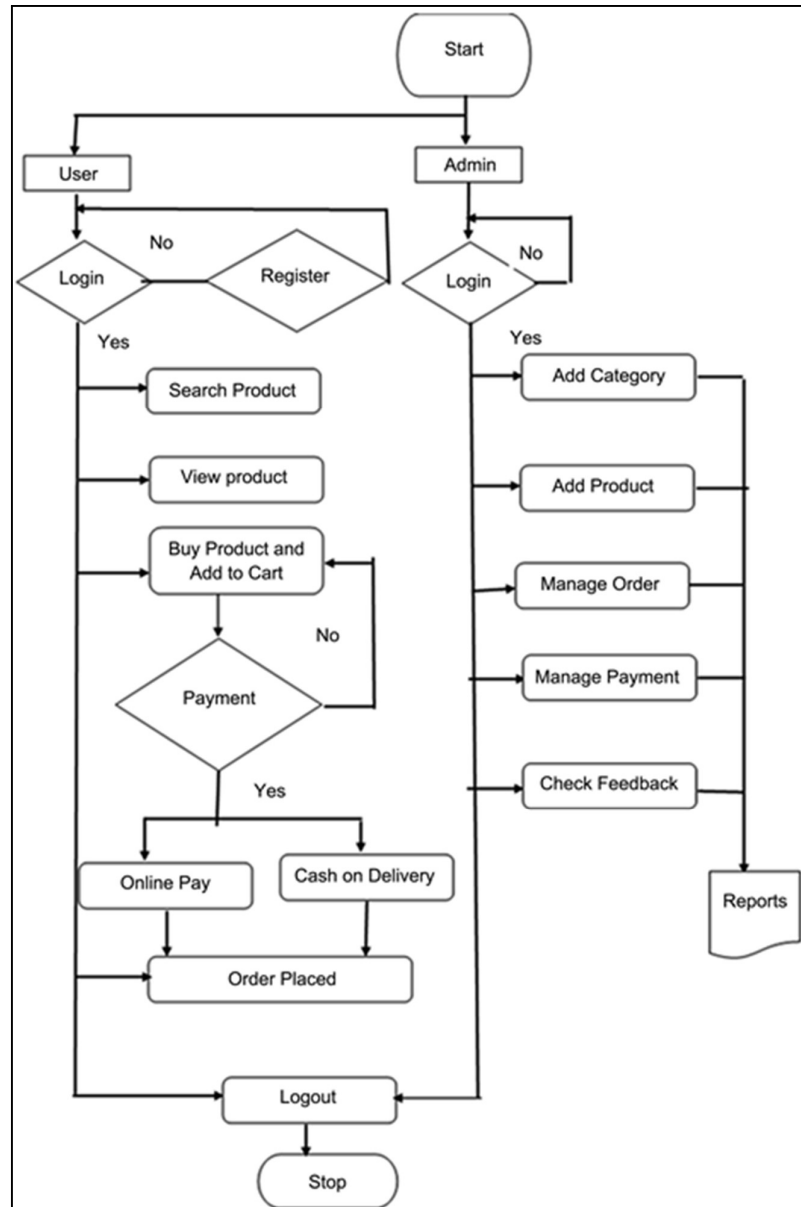


Fig 3.9 Flowchart

## CHAPTER 4

### OUTPUTS & RESULT

The initial version of an electronic commerce application that reproduces an online store was successfully built by leveraging all four key technologies that comprise the MERN stack and making use of numerous Node modules. This particular programme is made to be effective, simple to use, and run smoothly.

#### 4.1 HOME PAGE

The home page of an e-commerce site is the main landing page that customers first see when they visit the website. It serves as the storefront or the main entrance to the online store. The home page provides a snapshot of the website's products, services, and overall brand identity. The home page typically includes several key elements, including a Navigation menu, Featured products, Promotional banners, Search bar, and About Us section.

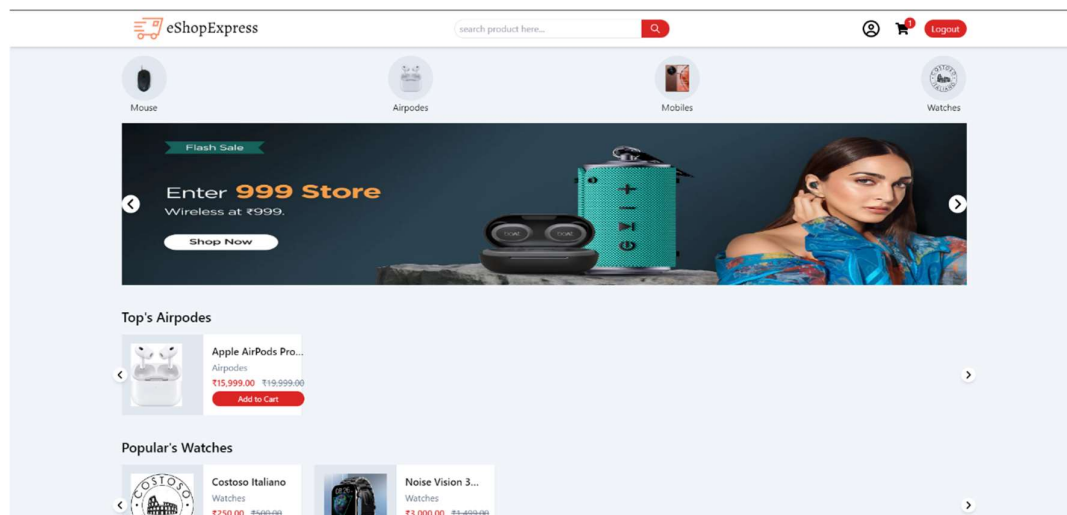


Fig 4.1 Home Page

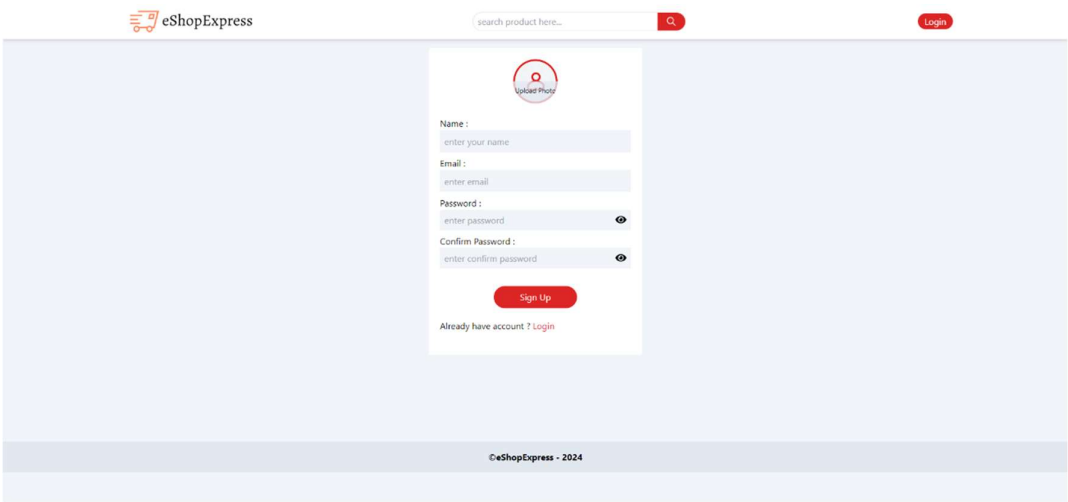
#### 4.2 SIGNUP PAGE

A signup page is a webpage that allows users to create an account with a website or online service. It is typically the first step in the onboarding process for new users. The signup page collects important information from the user, such as their name, email address, and password, and may also ask for additional details such as their date of birth or location.

Overall, the signup page is an important part of the user onboarding process, and it's essential that it's designed to be user-friendly, easy to navigate, and secure. A well-



designed signup page can help to build trust with users and encourage them to continue using the website.

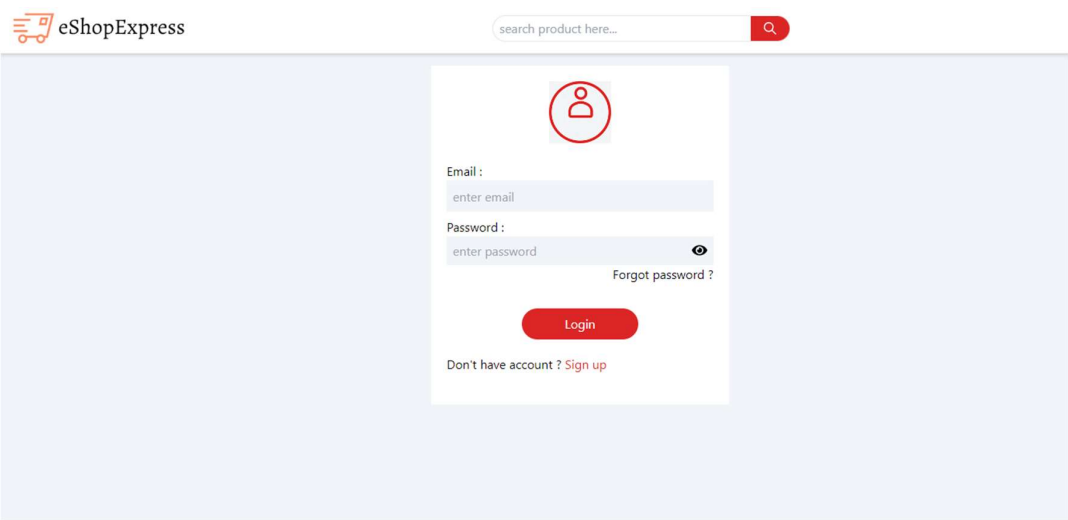


The image shows a web browser displaying the eShopExpress sign-up page. The header includes the eShopExpress logo, a search bar with the placeholder text "search product here...", and a "Login" button. The main content area features a central white box with a red circular icon containing a person silhouette. Below the icon are input fields for "Name", "Email", "Password", and "Confirm Password", each with a placeholder text. A red "Sign Up" button is positioned below the "Confirm Password" field. At the bottom of the white box, there is a link that says "Already have account ? Login". The footer of the page displays "eShopExpress - 2024".

Fig 4.2 Sign-Up Page

### 4.3 SIGN IN PAGE

The sign-in option on a website or application allows users who have already created an account to log back into their account. This allows users to access their personal information, settings, and account features. Typically, the sign-in option is located on the website or application's homepage and can be accessed by clicking on a "sign in" or "log in" button. Once clicked, the user is directed to a sign-in page, where they are prompted to enter their username or email address and password.



The image shows a web browser displaying the eShopExpress sign-in page. The header includes the eShopExpress logo, a search bar with the placeholder text "search product here...", and a "Login" button. The main content area features a central white box with a red circular icon containing a person silhouette. Below the icon are input fields for "Email" and "Password", each with a placeholder text. A red "Login" button is positioned below the "Password" field. To the right of the "Password" field, there is a link that says "Forgot password ?". At the bottom of the white box, there is a link that says "Don't have account ? Sign up".

Fig 4.3 Sign-In Page

## 4.4 PRODUCT PAGE

The product page is a webpage on an e-commerce site that provides information about a specific product. It typically includes the product name, image, price, and a brief description of its features and benefits. The goal of the product page is to provide customers with enough information to make an informed purchasing decision. Other common elements on a product page may include customer reviews and ratings, related products, and an option to add the product to a shopping cart or wishlist. Overall, a well-designed product page is crucial for driving conversions and boosting sales on an e-commerce site.

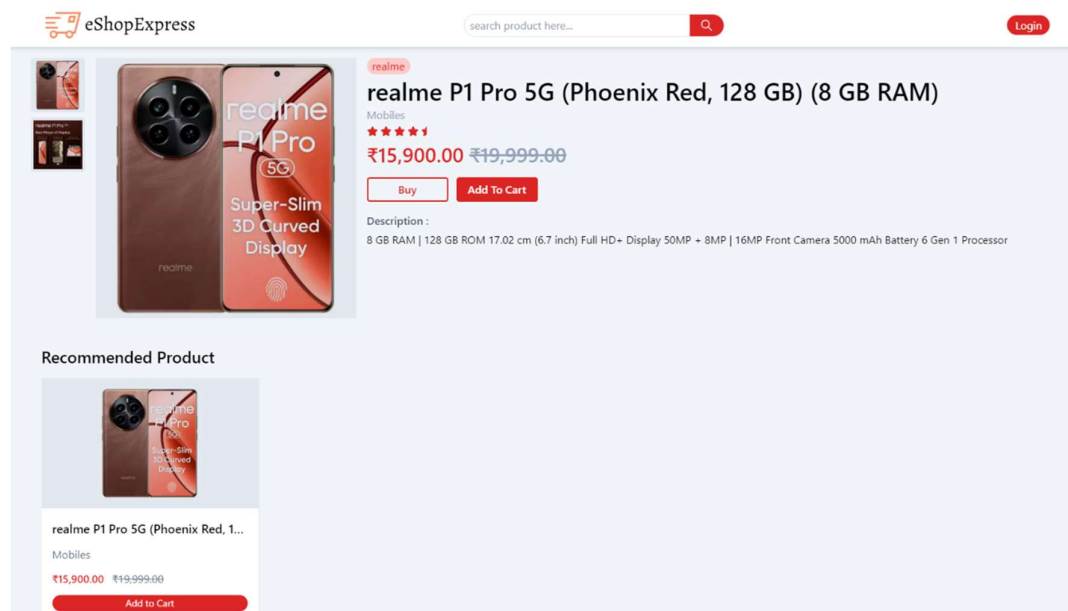


Fig 4.4 Product Page

## 4.5 CART PAGE

The cart page (also known as a shopping cart or basket) is a webpage on our e-commerce site that displays a customer's current order. The cart page typically shows a list of the products the customer has added to their cart, along with the quantities and prices. Customers can typically edit the quantities of each item or remove items from their cart altogether. The cart page may also display the subtotal, taxes, shipping costs, and total order costs. From the cart page, customers can proceed to checkout to complete their orders or continue shopping to add more items to their cart. A well-designed cart page is important for providing customers with a clear overview of their order, and for encouraging them to complete their purchase.

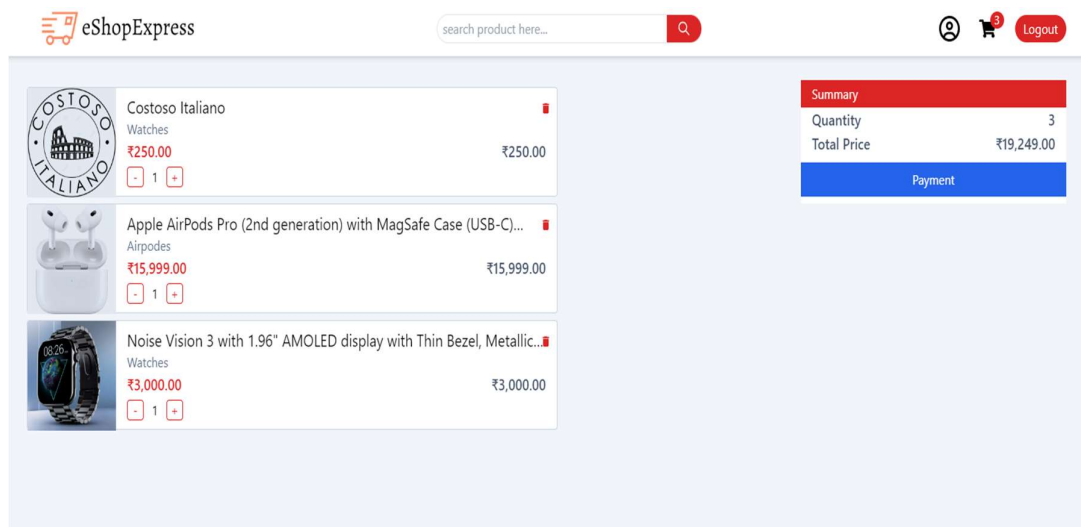


Fig 4.5 Cart Page

## CHAPTER 5

### TESTING

#### 5.1 INTRODUCTION

Testing is the bedrock of software development, especially in complex applications like eShopExpress. It serves as a vital phase in the development lifecycle, ensuring the application's functionality, performance, and reliability. This document delves into the multifaceted testing strategies employed to scrutinize eShopExpress thoroughly. By expanding on two pivotal test cases - User Registration and Adding Items to Cart - this document aims to provide a comprehensive understanding of the rigorous testing methodologies employed in guaranteeing a seamless user experience.

##### 5.1.1 Understanding Testing Methodologies

The success of any testing endeavor hinges on a robust framework of methodologies. Unit testing dissects individual components to validate their functionality in isolation. Integration testing examines the cohesion between these components, ensuring they interact seamlessly. User acceptance testing validates the application's alignment with user expectations and specified requirements. In the context of eShopExpress, the amalgamation of these methodologies forms a comprehensive testing framework, essential for ensuring its reliability and effectiveness.

#### 5.2 TEST CASE-1: USER REGISTRATION

**Objective:** This test case aims to validate the user registration process within eShopExpress.

##### 5.2.1 Preconditions:

- eShopExpress must be accessible and operational.
- The user must navigate to the registration page.

##### 5.2.2 Test Steps:

1. Input valid information, including a unique email address, full name, and secure password, into the registration form.
2. Click on the "Submit" button to commence the registration process.

### **5.2.3 Expected Results:**

A confirmation message should be promptly displayed upon successful registration.

The user's information, particularly their email, should be securely stored in the database for future authentication.

### **5.2.4 Postconditions:**

The user should seamlessly log in using the registered credentials, gaining access to eShopExpress's features and functionalities without hindrance.

## **5.3 TEST CASE-2: ADDING ITEMS TO CART**

**Objective:** This test case validates the core functionality of adding items to the cart within eShopExpress.

### **5.3.1 Preconditions:**

- The user must be logged into eShopExpress.
- The user must navigate to the product listing or detail page.

### **5.3.2 Test Steps:**

1. Navigate to a specific product's page.
2. Select product options (if applicable, such as size or color).
3. Click on the "Add to Cart" button.

### **5.3.3 Expected Results:**

The product should be promptly added to the user's cart and a confirmation message displayed.

The cart should reflect the added item, including details like product name, price, and selected options.

### **5.3.4 Postconditions:**

The added item must be readily accessible within the cart, ensuring that the user can proceed to checkout or continue shopping seamlessly.

## CONCLUSION

E-Commerce is a flexible answer for consumers and businesses. In this competitive and convenience prominent period, only some people have the time and patience to spend time in markets to buy their products and services. An online store is open all hours a day, every day, which means all your consumers can visit your store whenever they want, regardless of their schedule. E-Commerce is the requirement of the current period, which is being well-served. This project is based on a technology known as MERN stack, which makes it possible to buy and sell products on this online store. This project has features for searching and sorting products based on price and popularity. And it also can store and save user data like profile information, orders, and past searches. The online store is simple to use for sellers, they can easily add products and establish new categories. Consumers will find it quite attractive to look at the products while sitting at home or at the office.

An e-commerce website built using the MERN stack (MongoDB, Express.js, React, and Node.js) is a powerful solution for businesses looking to sell products online. The MERN stack provides a robust and scalable architecture, making it an excellent choice for building complex web applications like e-commerce websites. With MongoDB as the database, Express.js as the web application framework, React for building the user interface, and Node.js for server-side scripting, the MERN stack offers a comprehensive and cohesive development environment. This full-stack JavaScript approach allows developers to build efficient and responsive e-commerce websites that can handle high volumes of traffic and provide a seamless user experience. An e-commerce website built using the MERN stack can offer a range of features and functionalities, such as product listings, search and filtering options, shopping carts, payment gateways, and order tracking. These features can be tailored to the specific needs of the business, ensuring that the website meets the unique requirements of the e-commerce industry. Overall, an e-commerce website built using the MERN stack is a powerful tool for businesses looking to expand their online presence and sell products to a wider audience. By leveraging the capabilities of the MERN stack, developers can create robust and reliable e-commerce websites that provide a seamless and enjoyable shopping experience for customers.

## REFERENCES

- [1] Thi Thu Hien Tran. (2022). THE DEVELOPMENT OF AN ECOMMERCE WEB APPLICATION USING MERN STACK.
- [2] Monika Mehra, Manish Kumar, Anjali Maurya, Charu Sharma, Shanu. (2021). MERN Stack Web Development. *Annals of the Romanian Society for Cell Biology*, 25(6), 11756–11761.
- [3] Naidu, N. D., Adarsh, P., Reddy, S., Raju, G., Kiran, U. S., Sharma, V., ... & Sharma, V. (2021). E-Commerce web Application by using MERN Technology. *International Journal for Modern Trends in Science and Technology*, 7, 1-5.
- [4] Mai, N. (2020). E-commerce Application using MERN stack. Subramanian, V. (2019). MongoDB. In: Pro MERN Stack. Apress, Berkeley, CA.
- [5] Hirenkumar Pravinbhai Vacchani. (2018). A Critical Study of E-Commerce Market of India. *Vidhyayana - An International n Multidisciplinary Peer-Reviewed E-Journal - ISSN 2454-8596*, 4(2).
- [6] Ullah, S. E., Alauddin, T., & Zaman, H. U. (2016, January). Developing an E-commerce website. In 2016 International Conference on Microelectronics, Computing and Communications (MicroCom) (pp. 1-4). IEEE.
- [7] Niranjnamurthy, M., Kavyashree, N., Jagannath, S., & Chahar, D. (2013). Analysis of e-commerce and m-commerce: advantages, limitations and security issues. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(6), 2360-2370.
- [8] Chanana, N., & Goele, S. (2012). Future of e-commerce in India. *International Journal of Computing & Business Research*.