



Building an End-to-End E-commerce solution

Designing and Implementing a Full-stack E-commerce website

Md Touhidul Islam

BACHELOR'S THESIS
May 2023

Software Engineering

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Bachelor's Degree Programme in Software Engineering

MD TOUHIDUL ISLAM:
Building an End-to-End E-commerce Solution
Designing and Implementing a Full-stack E-commerce Website

Bachelor's thesis 40 pages
May 2023

E-commerce is becoming a strong and important way of making business. This thesis presents the implementation and technology behind TradeTreeTop, an e-commerce application. This application was created to make use of online shopping's advantages, expand the target market, increase sales, and improve the entire consumer experience. The study then explores the application's technology framework, system design, and implementation process. Features of this application are user registration and login system, authentication process, profile management, product list, and shopping cart functionalities. The application was developed using technologies such as React, NodeJS, MongoDB, Express, and Chakra UI components.

At the end of the thesis will be clearer about the responsiveness of the application.

Key words: e-commerce, react, node.js, responsive, javascript, business, redux, mongodb

CONTENTS

1	INTRODUCTION	5
2	E-COMMERCE	6
2.1	Overview of E-commerce Scenario.....	6
2.1.1	Traditional Commerce vs E-commerce	7
2.2	E-commerce business models	7
2.2.1	B2C vs C2C.....	9
2.3	Technology Trends in E-commerce.	9
2.3.1	Social Media Commerce	9
2.3.2	Livestream Commerce	10
2.3.3	Buy and Pickup from Store	10
2.3.4	Buy Now, Pay Later.....	10
2.4	E-commerce Platforms and Frameworks	10
3	TECHNOLOGY FRAMEWORK	12
3.1	MERN Stack Overview.....	12
3.2	Advantages of MERN Stack for e-commerce development	13
3.3	Overview of MongoDB	14
3.4	Overview of Express	15
3.5	Overview of React.....	15
3.6	Overview of Node.....	17
3.7	Difference of MARN Stack with other stacks.....	17
4	SYSTEM DESIGN	18
4.1	Functional Requirements and use cases	18
4.1.1	Functional Requirements.....	18
4.1.2	Use Case.....	18
4.2	Architecture Design.....	19
4.3	Database Design.....	19
5	IMPLEMENTATION	23
5.1	Environment setup configuration.....	23
5.1.1	Installing React Application	23
5.1.1	Installing Node.js and Express.js.....	23
5.2	Client-side implementation.....	24
5.2.1	Folder and Structure.....	24
5.2.2	Redux.....	25
5.3	Third-party library selection.....	28
5.3.1	Chakra UI	28

5.3.2 Axios.....	29
5.3.3 React Router Dom.....	30
5.3.4 React Icon	32
5.4 Challenges faced during implementation	33
5.5 Deployment with Vercel	33
6 Responsive	34
6.1 Mobile vs desktop	34
7 DISCUSSION	38
REFERENCES	39

1 INTRODUCTION

The advancement of technology and the widespread use of the internet have brought about significant changes in the business landscape of modern society. Most businesses are increasingly shifting their items to e-commerce platforms. E-commerce means purchasing and selling goods and services over the Internet. Increasing business conversion to e-commerce to increase customers, enhance efficiency, and cut costs. Another aspect of e-commerce growth is the increased use of mobile phones and other gadgets, as well as high-speed internet. These developments make it easier for organizations to reach clients at any time and from any location, while also offering new opportunities and saving time. Furthermore, the expansion of e-commerce presents other issues, such as the necessity for data protection and fraud prevention.

The purpose of this thesis is to develop a typical e-commerce platform TradeTreeTop. The platform should be user friendly and responsive. It should offer a variety of products that clients may buy online, and it should be created using the recent mainstream web development technologies. The platform should be accessible from any device, including mobile, laptop, and desktop.

The TradeTreTop e-commerce platform, including its features, design, and functionality, is examined in this thesis.

2 E-COMMERCE

2.1 Overview of E-commerce Scenario

The e-commerce solutions have become increasingly popular in the modern business world. The methodology is designed to help businesses, organizations, and customers to reduce operational costs while delivering high quality services. By adopting e-commerce techniques companies can eliminate paperwork.

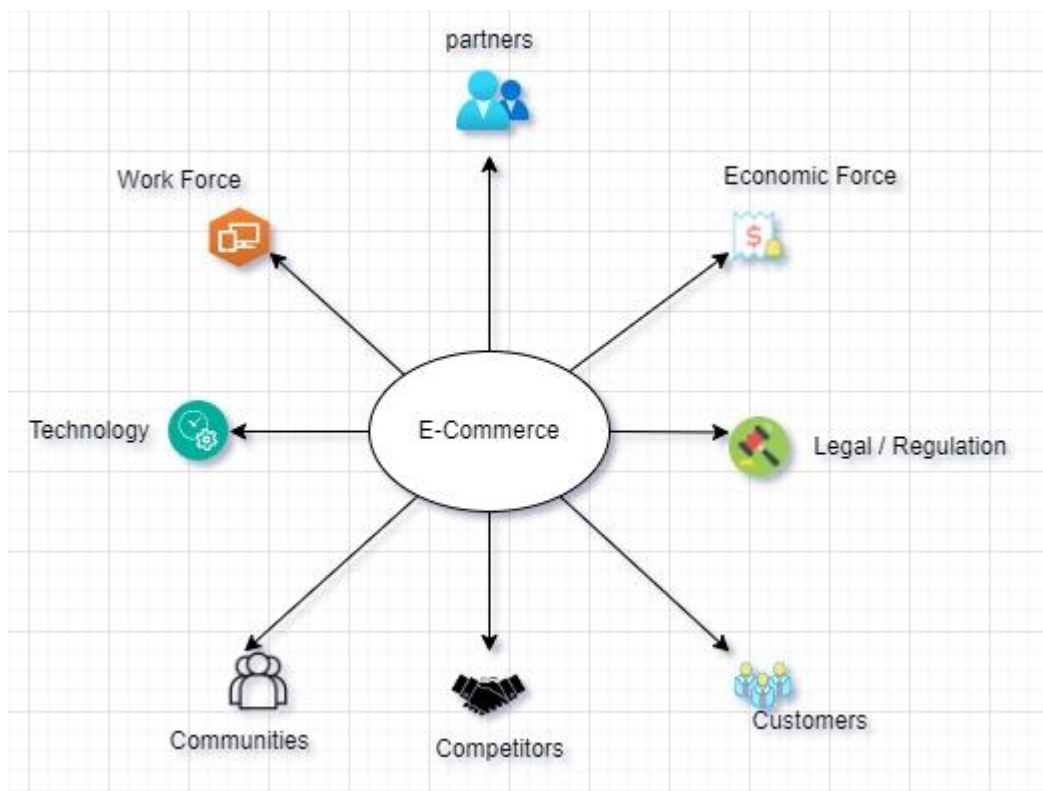


FIGURE 1. E-commerce Environment

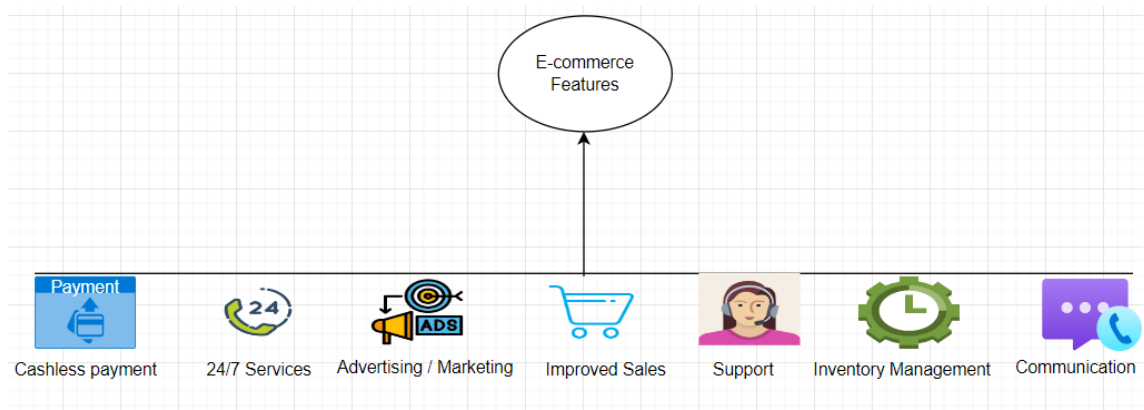


FIGURE 2. E-commerce Features

2.1.1 Traditional Commerce vs E-commerce

TABLE 1. Traditional Commerce vs E-Commerce (Jane, S. n.d.a)

Traditional	E-commerce
The transaction was done manually.	The transaction was done automatically.
Difficult to maintain standards.	Easy to establish and maintain.
Depends on personal communication.	Websites provide all information in one place.
Information exchange from person to person is dependent on it.	Very little dependency on the person. Easy to share information via electronic communication.

2.2 E-commerce business models

There are several types of business models. Each of the models has different challenges and many companies use several models. B2C (Business to consumer) is a business model that sells products directly to the end user. B2B (Business to business) that sells products as well as services to another business. B2B2C (Business to business to consumer) (Mcfarland, C. 2022a). This model sells products in partnership with another organization. B2G (Business to government) e-commerce model sells products to public administrations (Big

Ecommerce. n.d.a). C2B (Consumer to Business) model sells products and services to companies (Mcfarland, C. 2022a). D2C (Direct to consumer) model sell product directly to the customer without any other partners. C2C (Consumer to consumer) model like online marketplace tori. fi (Mcfarland, C. 2022a).

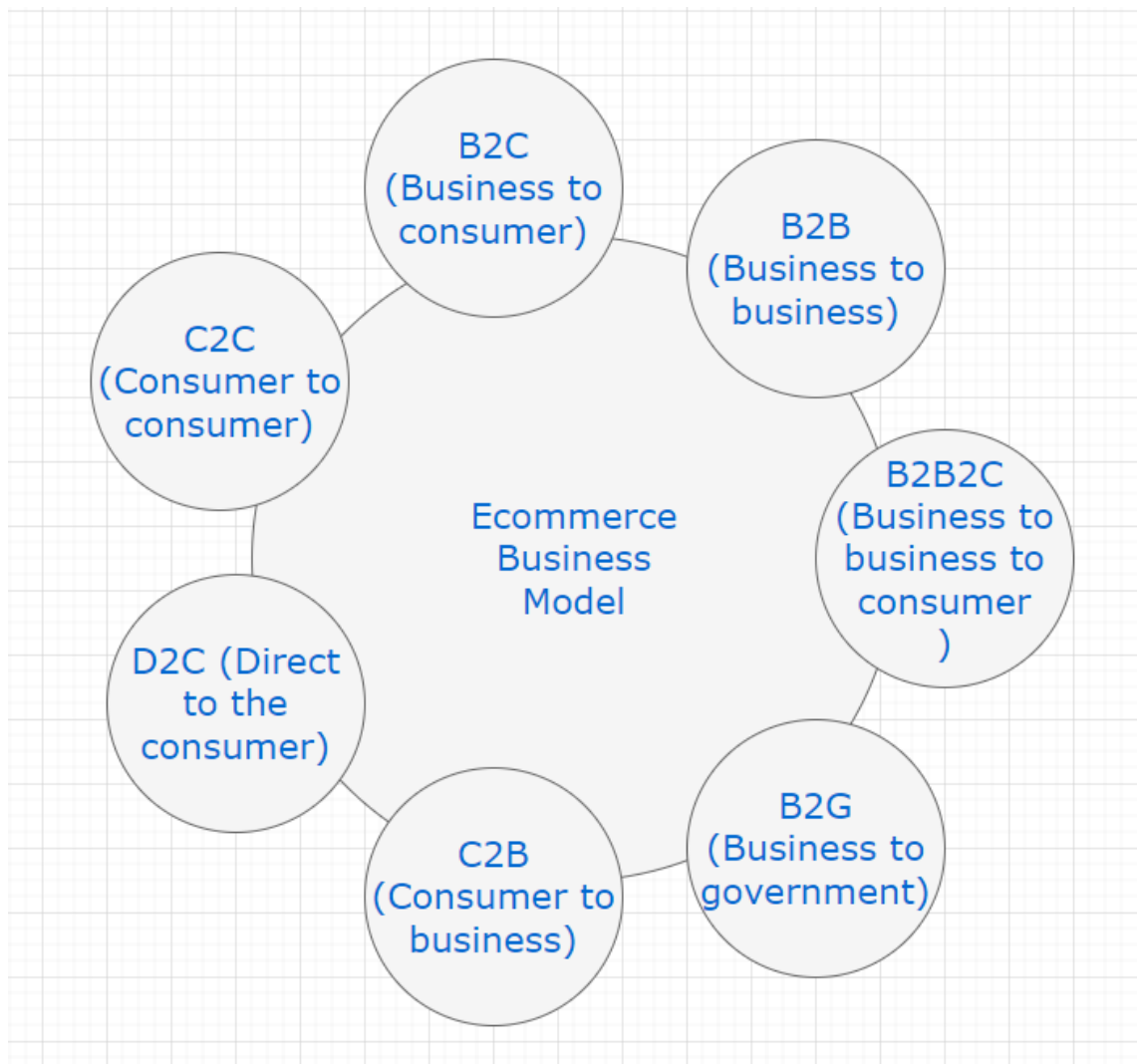


FIGURE 3. E-commerce Business Model (Learn e-commerce, n.d.)

2.2.1 B2C vs C2C

TABLE 2. B2C vs C2C (Mcfarland, C. 2022b)

B2C	C2C
Business to consumer.	Consumer to consumer.
Sell products directly to the end user.	Sell products between consumer to consumer.
More complex and need research.	Easy to manage without involving middlemen.
Provide products and services.	Provide only services.
Examples: Spotify, Netflix, and Amazon.	Example: Fiverr, Upwork.

2.3 Technology Trends in E-commerce.

The advent of new e-commerce technologies has given rise to a growing preference for purchasing products through online channels among consumers worldwide. E-commerce platforms have become very incredibly popular over time compared to traditional brick-and-mortar stores. Hence businesses big or small are embracing the trend with a combination of conventional and digital marketing techniques to boost sales.

2.3.1 Social Media Commerce

Today's world thrives on social media as one of the most prominent platforms for business expansion through enhanced communication avenues. Entrepreneurs can leverage this network by seamlessly marketing brands while providing prompt customer service - ultimately motivating consumers to purchase online. Through social media e-commerce tactics that focus on direct product sales or website-based traffic generation; businesses can cover all aspects essential in fostering successful transactions within a social commerce ecosystem.

2.3.2 Livestream Commerce

It is a commercial platform where the host shows viewers different goods and takes orders by sending messages via Messenger or WhatsApp. Also, viewers can buy products directly from the shop's website. This trend has been increasing day by day. This approach helps to present items according to the customer's feedback. Facebook is a prominent platform for Livestream e-commerce.

2.3.3 Buy and Pickup from Store

E-commerce technology has given rise to an innovative shopping experience whereby customers can purchase items online and collect them from stores at their convenience. Providing flexibility and cost-saving benefits by allowing shoppers to choose their preferred product quickly and schedule hassle-free pickups from a nearby store without any additional transportation charges.

2.3.4 Buy Now, Pay Later

This trend become more popular during the pandemic when people needed alternative funding and flexibility. Customers have been able to buy products interest-free for several months. This system is provided most of the time by a third party. In Finland, Klarna provides a one-month interest-free pay-later service(Shopping with Klarna, n.d.a). It provides a one-month interest-free pay-later service. Sometimes, the terms of these services may change over time.

2.4 E-commerce Platforms and Frameworks

The e-commerce an online software platform for consumers to buy products. It has different features like add to cart, checkout, find specific products, order management, and payment. There are three types of commerce platforms SaaS (software-as-service), PaaS (platform-as-service), and On-premises (E-commerce framework. n.d.a). SaaS and PaaS both deliver e-commerce solutions for businesses. Shopify is an example of a SaaS platform (Ecommerce framework. n.d.a).

The E-commerce frameworks prebuild the solution for e-commerce and can be customized according to the business needs to provide everything needed for an e-commerce store (Ecommerce framework. n.d.a).

3 TECHNOLOGY FRAMEWORK

3.1 MERN Stack Overview

MERN is an abbreviation from MongoDB, ExpressJS, ReactJS, and NodeJS. The MERN stack is a popular web development stack. It is a front-end and back-end JavaScript framework that can be used to build web applications. The MERN stack developer can develop a client and server-side applications (Stack difference. n.d.a).

- a. MongoDB: It is a NoSQL database. NoSQL database store data and manage data in JSON format (Stack difference. n.d.a).
- b. ExpressJS: Framework for the backend used to build a web app and API (Stack difference. n.d.a).
- c. ReactJS: JavaScript library, is used to build the user interface (Stack difference. n.d.a).
- d. NodeJS: Runtime environment for the backend that enables running JavaScript outside of a web browser (Stack difference. n.d.a).

3.2 Advantages of MERN Stack for e-commerce development

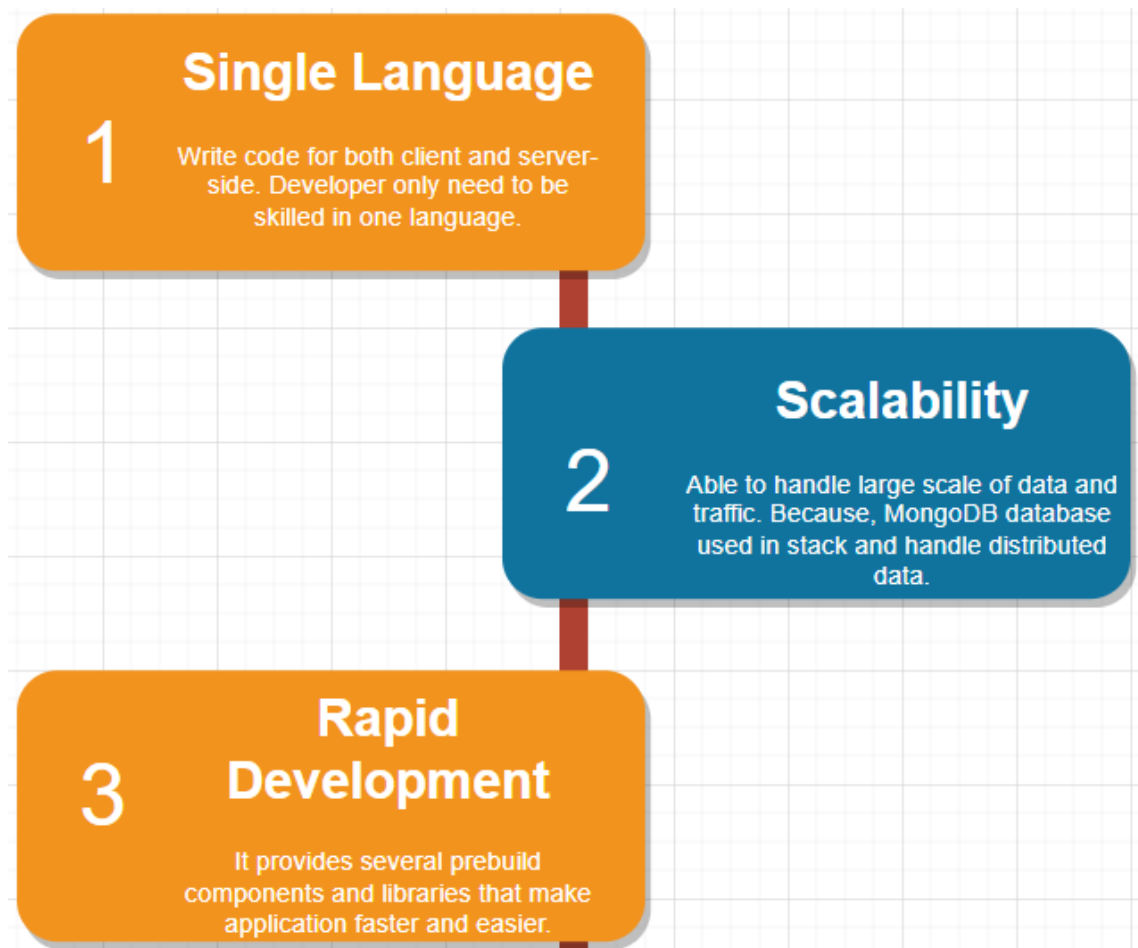


FIGURE 4. Advantages of MERN stack

3.3 Overview of MongoDB

MongoDB is a popular NoSQL database created by MongoDB Inc and first release February 11, 2009 (MongoDB. n.d.a). Unlike traditional SQL databases that rely on strict schemas for organizing entities, Mongo allows developers a great deal of flexibility when it comes to structuring data. This makes it an ideal solution for those managing complex datasets that may not fit neatly into predefined tables.

MongoDB can be scaled effortlessly without sacrificing performance. MongoDB information sharing capabilities ensure that the system will remain reliable and performant under heavy loads. MongoDB also offers support for memory computing - enabling fast query responses that are essential in today's high-speed application ecosystem. The ACID transaction support ensures that multi-document operations always maintain their integrity throughout the process.

MongoDB is a good solution manage large amounts of unstructured or semi-structured data. It is open source by nature and active community also makes it a wise choice for businesses of all sizes looking to drive innovation and get the most out of data.

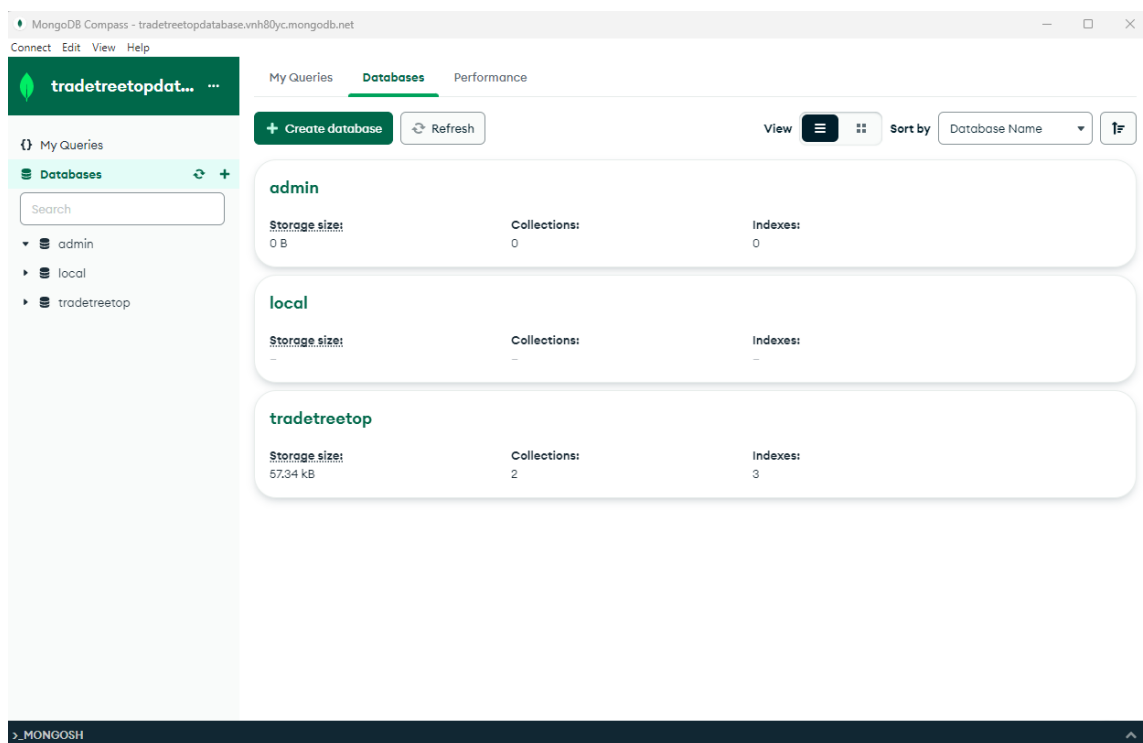


FIGURE 4. MongoDB database for e-commerce application

3.4 Overview of Express

Node.js Express framework provides developers with a lightweight and flexible solution for building web applications. Middleware functions in Express.js are combined into a pipeline that handles HTTP requests, such as GET, POST, PUT, and DELETE (HTTP methods. n.d.a). Middleware and factories are commonly used in e-commerce applications, such as AI chatbots.

```

1 import express from 'express';
2 ..
3 const app = express();

```

FIGURE 5. Express application

Figure 6 shows an example of using Middleware function in Express application. The request passes through different Middleware functions one by one, each of which is responsible for independent well-defined responsibility like logging or setting headers. The modular approach of middleware functions makes the application easily expandable. Benefit of this pattern is flexibility distribution among objects.

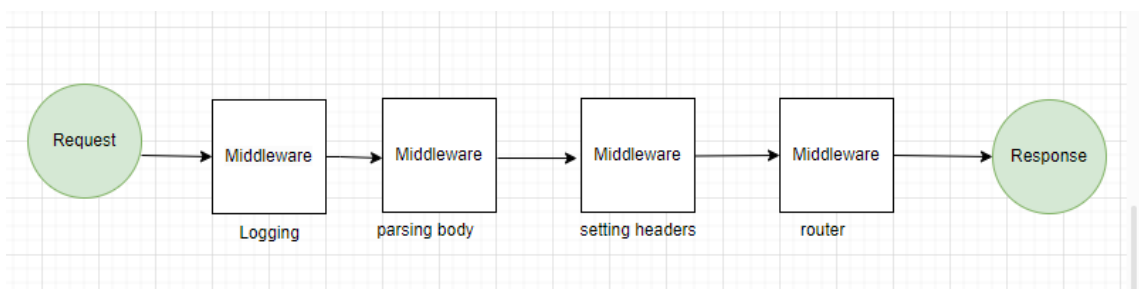


FIGURE 6. Middleware request-response lifecycle (Ho, C. 2016a).

3.5 Overview of React

A JavaScript package called React is used to create reusable user interface elements. JSX, components, unidirectional data flow, and license are its four primary features. HTML may be used in React apps thanks to JSX, which is mainly JavaScript XML. Data can be given from one component to another as a prop in React, and components can be reused. Additionally, react uses one-way data

flow and is Facebook-licensed. React makes use of virtual DOM, a JavaScript object that is quicker than traditional DOM. It can be used with other frameworks for both server and client applications. Large apps can be easier to manage with React components (Stack difference, n.d.d).

```
import React from "react";

function App() {
  return (
    <div>
      <h1>Hello World!</h1>
    </div>
  );
}
export default App;
```

FIGURE 7. Printing Hello World in React App

```
import React from 'react';
import ReactDOM from 'react-dom';
import HelloWorld from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

FIGURE 8. Rendering the main component onto the root element

3.6 Overview of Node

Node.js is a server-side platform that offers a runtime environment constructed on the JavaScript engine of Google Chrome. It was created in 2009 by Ryan Dahl and provides a variety of libraries that make it easier to design web applications. Node.js is renowned for being MIT-licensed, speedier, single-threaded, buffer-free, and asynchronous. Numerous businesses, including Microsoft, PayPal, Uber, and Yahoo!, utilize Node.js. (NodeJS history, n.d.a.) It is frequently used for backend development in single-page applications, chat apps, data streaming, and IoT apps.

3.7 Difference of MARN Stack with other stacks

MERN	MEAN
MERN uses MongoDB, Express, React, and Node.	MEAN uses MongoDB, Express, Angular, and Node.
JS library.	JavaScript framework.
Lower productivity.	Better productivity.
Data flow is bidirectional.	Data flow is unidirectional.

TABLE 2. Difference between the MARN and MEAN stack (Stack difference. n.d.e)

4 SYSTEM DESIGN

4.1 Functional Requirements and use cases

Functional requirements mean the functions included in the e-commerce web application. Use case means how the user will interact with the system or software application to achieve their goal (Dilmegani, C. 2022a). It also clarifies, identifies, and organizes the requirements of the software. Use cases can be used in business, project management, and system design (OpenAI ChatGPT 2023a).

4.1.1 Functional Requirements

The functional requirements for the TradeTreeTop Application are follows:

1. Shopping cart functionality.
2. Browse and search products.
3. Payments can be processed securely.
4. Order and shipment track capabilities.
5. Managing capabilities of stock levels and managing inventory.
6. Manage customer reviews and feedback.
7. Manage promotions and discounts.
8. Customer behavior and generate a report and analyze it.
9. User account creation and management.

4.1.2 Use Case

The primary use cases for the TradeTreeTop application and for any typical e-commerce application are as follows:

1. Customers can browse the web app and able to add products to their cart before checking out and completing purchases.
2. Customers can search for products and filter products with price.
3. Customers leaving items in the cart can be notified for completing tasks.
4. Customers can leave reviews for purchasing products.
5. Order tracking (Dilmegani, C. 2022b).

4.2 Architecture Design

The architecture design of an application refers to the fundamental approach utilized to build it. Although there are various forms of architecture, the most prevalent are two-tier, and three-tier, architectures. The application for this project employs a three-tier design comprised of presentation, logic, and data layers. The user interface is handled by the presentation layer, whereas the logic layer permits communication between the presentation and database layers. The data is stored in the database layer. This architecture design was chosen to ensure a clear separation of concerns and to enable the application's scalability and maintainability.

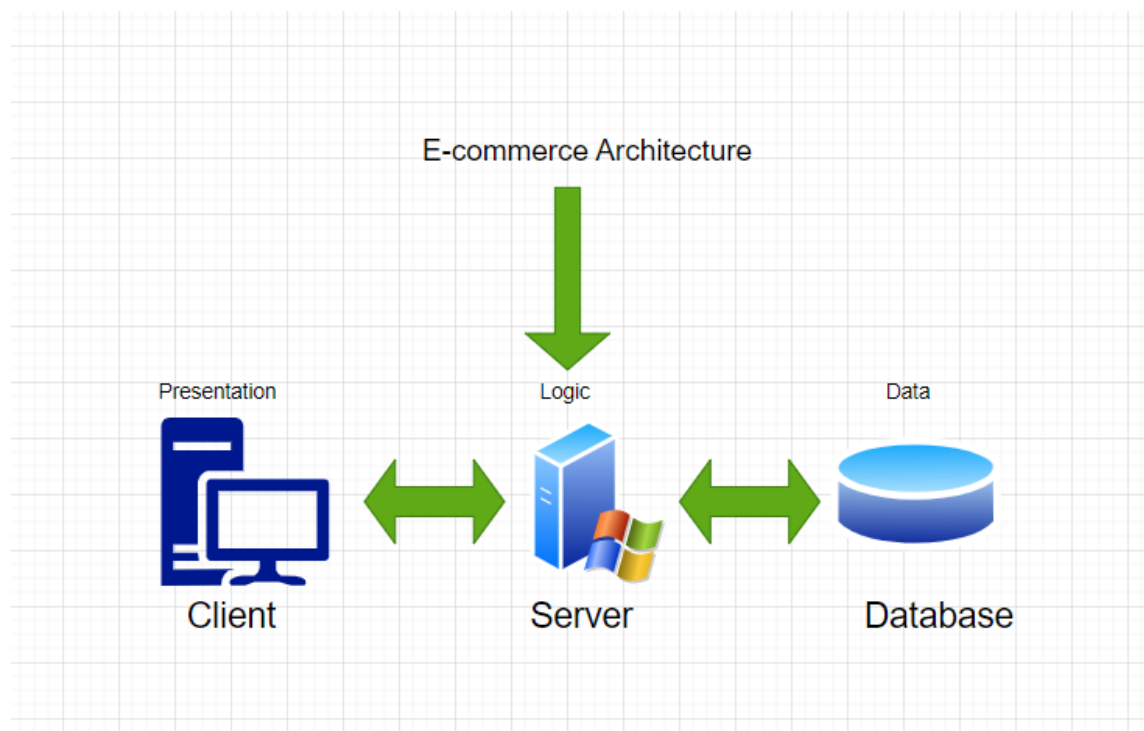


FIGURE 9. Three-tier e-commerce Architecture

4.3 Database Design

The database is an important part of data storage. The database design provides the system's structure and clarity, as well as identifies the e-commerce platform's operations, such as user management, product management, inventory management, shopping cart management, and payment administration. These features are critical for an e-commerce platform's day-to-day operations to ensure that

data is kept effectively, precisely, and securely, the database design must be carefully constructed (Lim, D. 2020a).

The Entity Relationship (ER) diagram to show the database design of the TradeTreeTop application are shown in FIGURE 10 to 14.

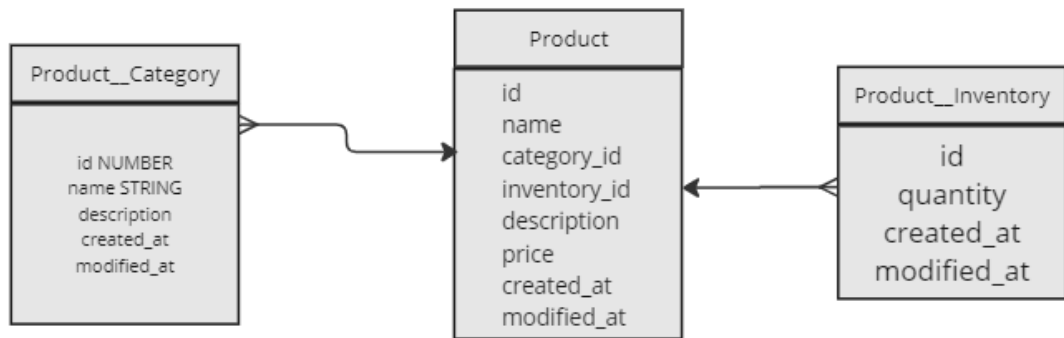


FIGURE 10. Product design for database

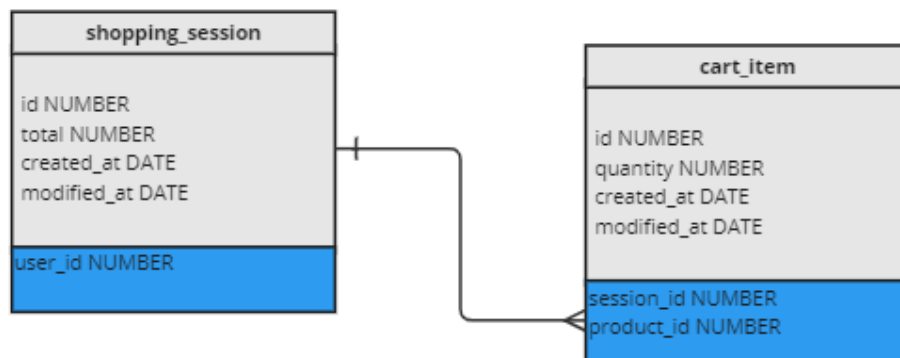


FIGURE 11. Shopping cart design

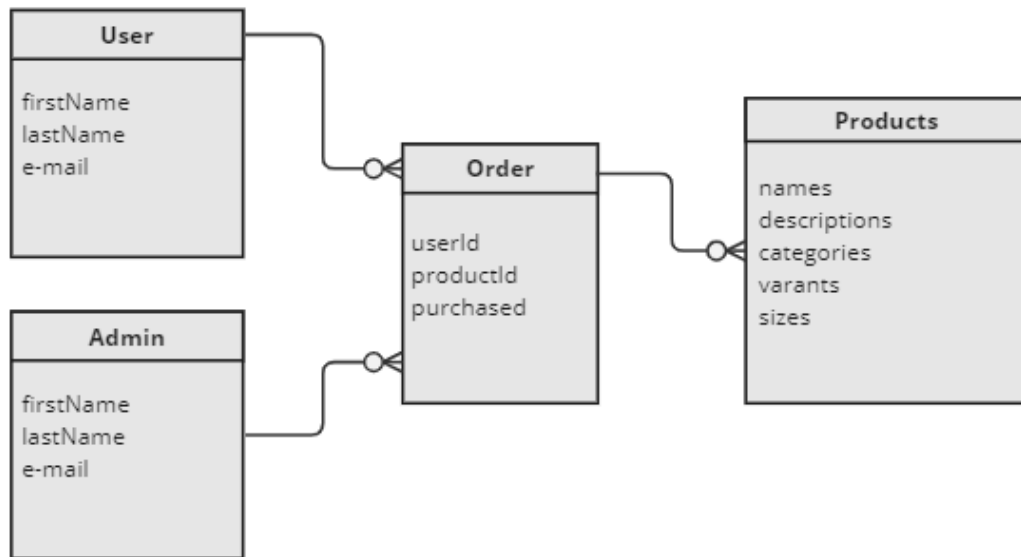


FIGURE 12. User order design

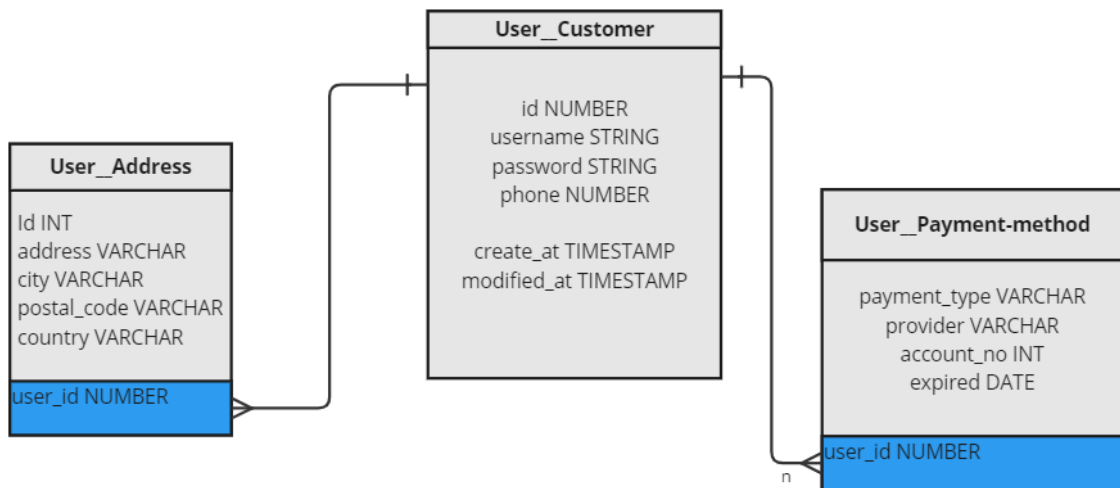


FIGURE 13. User customer design

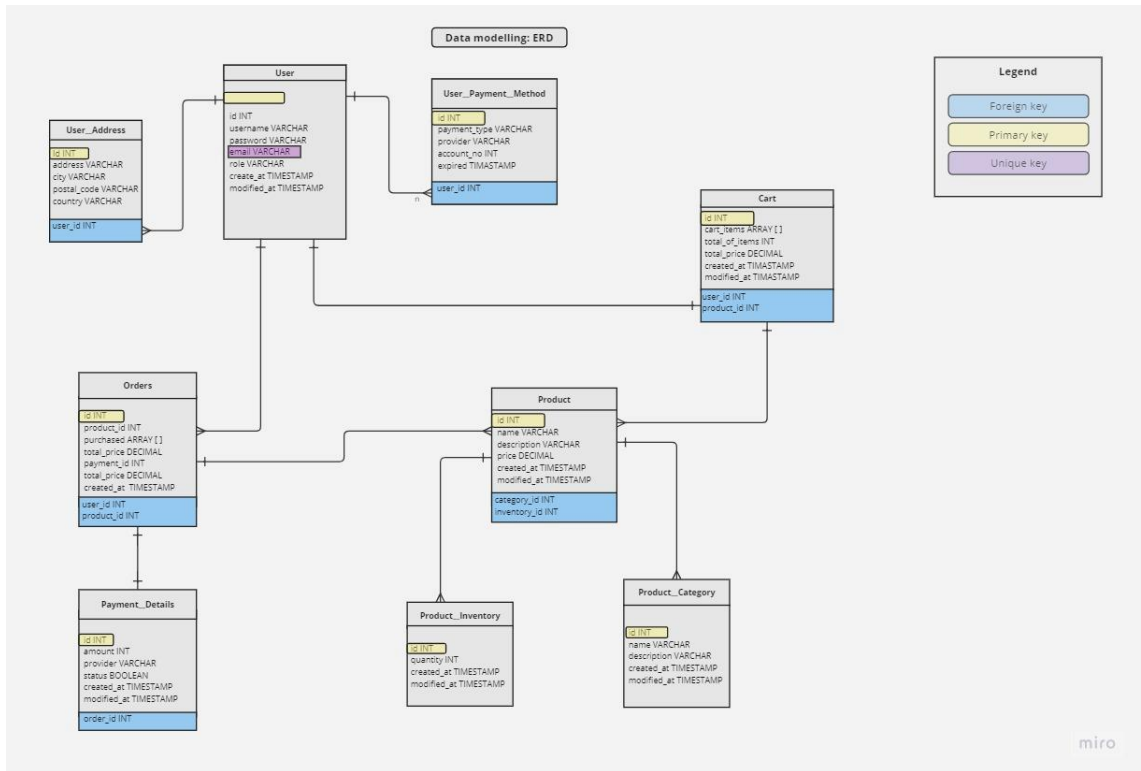


FIGURE 14. Data modelling EDR for e-commerce app (Lim, D. 2020b)

5 IMPLEMENTATION

5.1 Environment setup configuration

Node.js environment is necessary to develop code in JavaScript. Node.js version 16.16.0 is used in this application. Using the V8 JavaScript engine, Node.js allows code to be executed outside of the browser (Sufiyan, T. 2023a). MongoDB is used for data storage and maintenance, and writing code requires the use of an editor. For design and responsiveness, Chakra UI is used. GitHub is used to store code and maintain version control online.

5.1.1 Installing React Application

Following the installation of node.js, the react application is installed with the command **"npx create-react-app tradetreetop"** (React create app. n.d.a). This command generates a react app in the folder "tradetreetop." Then, using the command **"cd tradetreetop"** (React create app. n.d.a), move to the application directory. The application may be started with the command **"npm start"** which displays the react application's UI with the logo and the phrase "Hello World."

5.1.1 Installing Node.js and Express.js

Node.js is a runtime environment that allows programs to be run outside of the browser (n.d.a. Node.Js installation). Express is a popular online application framework that includes a variety of tools for developing web apps. To begin utilizing Express, execute the **"npm install express"** command or yarn, depending on the package management being used. Following installation, Express may be imported into the index.js file, and a route with a client that replies to a request to the root path can be defined. Finally, the program can be launched by specifying a port number, such as 3000.

5.2 Client-side implementation.

The client side refers to the user interface with which end users interact. The client side of this application was built with React.js and designed with Chakra UI. Chakra UI is a framework that allows developers to create simple, modular, and accessible React apps. Furthermore, Redux was used in this section to store data and make it easier to access that data from any component of the application by using state.

5.2.1 Folder and Structure.

For the creation of an application, a precise folder structure is essential. A well-organized folder structure makes it simpler for other developers to access the project code. The folder structure in this project has been thoughtfully created. Different files with either the .js or .jsx extension can be found in each folder. Three types of .js file folders and two types of .jsx file folders can be found in the client folder. The server folder, however, only contains files with the .js extension.



FIGURE 15. TradeTreeTop application folder structure

5.2.2 Redux.

Redux is an open-source JavaScript library for managing and centralizing states in React or Angular user interfaces. On June 2, 2015, it was released (Redux, n.d.a). Using state, Redux offers access to data stored everywhere in the component (De Roy, S. 2022.a). The store must be inserted as the provider in the Index.js file. Redux adopts a pattern that ensures all state transitions are explicit and monitored (De Roy, S. 2022.a).

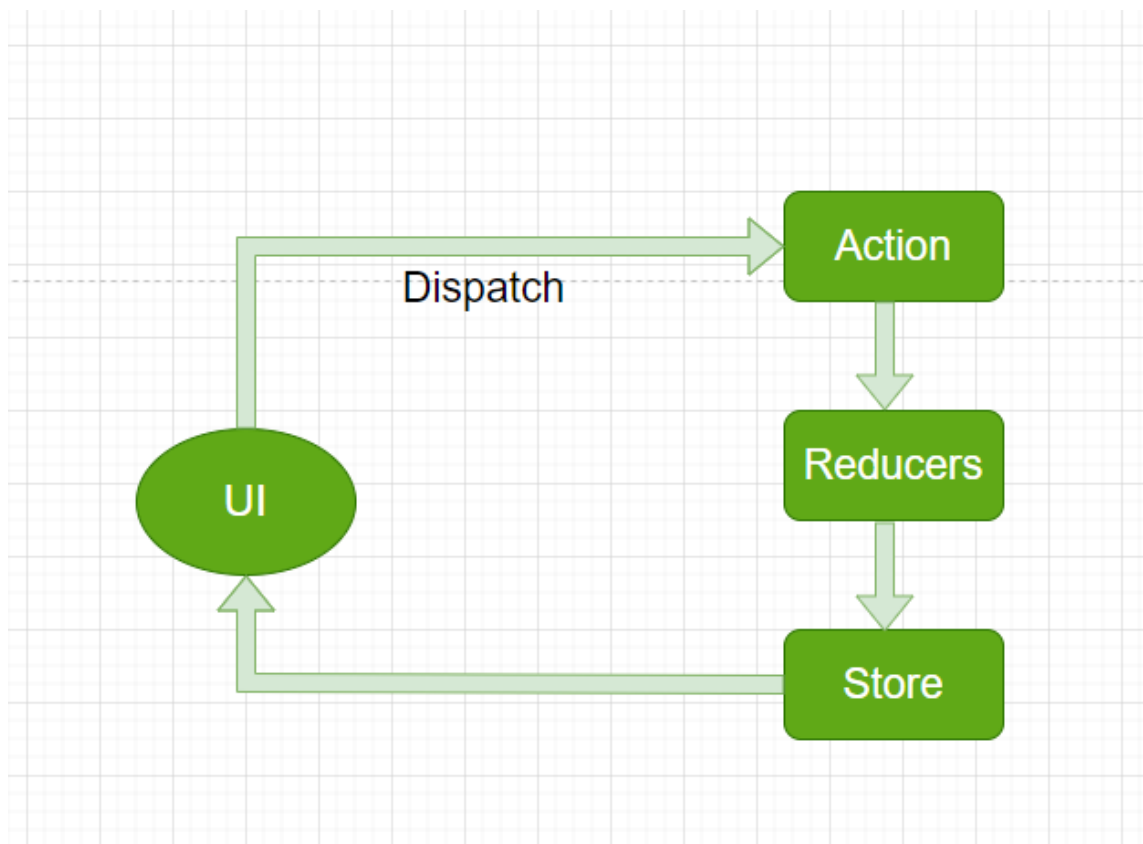



FIGURE 16. The basic concept of redux (De Roy, S. 2022. b)

An action in Redux is a JavaScript object that describes what will happen in the application. In an e-commerce project, there are many types of actions. Here is an example of an `addCartItem` action. This action handles the functionality of adding items to the cart.



```

// put your import axios from "axios";
import { setLoading, setError, cartItemAdd, cartItemRemoval } from "../slices/cart";

export const addCartItem = (id, quantity) => async (dispatch) => {
  dispatch(setLoading(true));
  try {
    const { data } = await axios.get(`/api/products/${id}`);
    const itemToAdd = {
      id: data._id,
      name: data.name,
      image: data.image,
      price: data.price,
      stock: data.stock,
      quantity,
    };
    dispatch(cartItemAdd(itemToAdd));
  } catch (error) {
    dispatch(
      setError(
        error.response && error.response.data.message
          ? error.response.data.message
          : error.message
          ? error.message
          : "An unexpected error, Please try it later."
        )
    );
  }
}

export const removeCartItem = (id) => async (dispatch) => {
  dispatch(setLoading(true));
  dispatch(cartItemRemoval(id));
} code here

```

snappify.com

FIGURE 17. Redux action example for cart

The Redux reducer sets the state object as an initial value. In the example below, the initial values are loading, error, cart, expressShipping, and subtotal as state objects. The cart reducer handles set loading, setError, cartItemAdd, and cartItemRemoval.

```

import { createSlice } from "@reduxjs/toolkit";

const calculateSubtotal = (cartState) => {
  let result = 0;
  cartState.map((item) => (result += item.quantity * item.price));

  return Number(result).toFixed(2);
};

export const initialState = {
  loading: false,
  error: null,
  cart: JSON.parse(localStorage.getItem("cartItems")) ?? [],
  expressShipping: false,
  subtotal: localStorage.getItem("cartItems")
    ? calculateSubtotal(JSON.parse(localStorage.getItem("cartItems")))
    : 0,
};

const updateLocalStorage = (cart) => {
  localStorage.setItem("cartItems", JSON.stringify(cart));
  localStorage.setItem("subtotal", JSON.stringify(calculateSubtotal(cart)));
};

export const cartSlice = createSlice({
  name: "cart",
  initialState,
  reducers: {
    setLoading: (state) => {
      state.loading = true;
    },
    cartItemAdd: (state, { payload }) => {
      const existingItem = state.cart.find((item) => item.id === payload.id);
      if (existingItem) {
        state.cart = state.cart.map((item) =>
          item.id === existingItem.id ? payload : item
        );
      } else {
        state.cart = [...state.cart, payload];
      }

      state.loading = false;
      state.error = null;
      updateLocalStorage(state.cart);
      state.subtotal = calculateSubtotal(state.cart);
    },
    setError: (state, { payload }) => {
      state.error = payload;
      state.loading = false;
    },
    cartItemRemoval: (state, { payload }) => {
      state.cart = [...state.cart].filter((item) => item.id !== payload);
      updateLocalStorage(state.cart);
      state.subtotal = calculateSubtotal(state.cart);
      state.loading = false;
      state.error = null;
    },
  },
});

export const { setLoading, setError, cartItemAdd, cartItemRemoval } =
  cartSlice.actions;
export default cartSlice.reducer;
export const cartSelector = (state) => state.cart;

```

FIGURE 18. Redux reducers example for cart

The individual reducers for products, carts, and users are combined and exported to the store using the combined Redux and configureStore redirection toolkit packages.

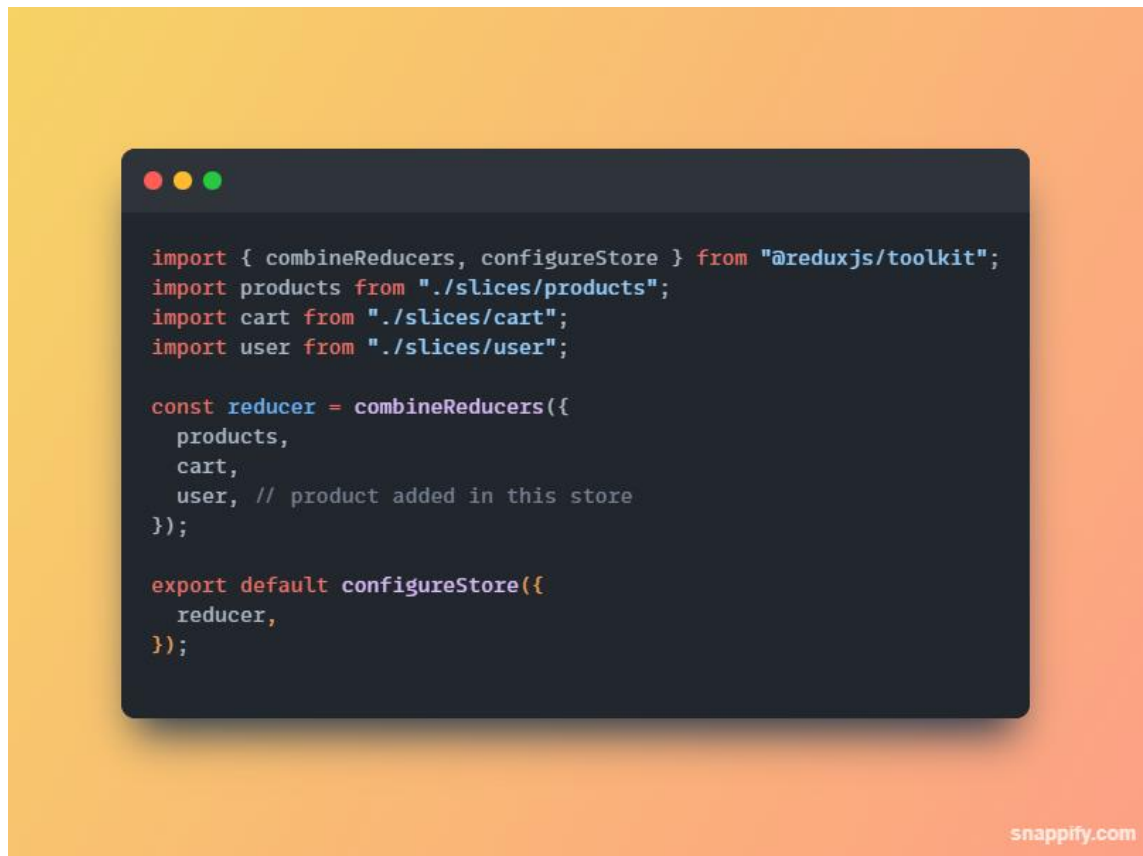


FIGURE 19. Redux store

5.3 Third-party library selection

Choosing the libraries to use with a modern full stack application can be challenging as there are so many possibilities. There are several possible approaches and requirements for choosing the libraries.

5.3.1 Chakra UI

Chakra UI is a popular component library for building user interfaces in React applications. It has a lot of design components for building responsive and accessible applications. Buttons, forms, navigate, box, and footer are examples of Chakra UI components. Chakra UI is installed to the client project with the command **"npm add @chakra-ui/react @emotion/react @emotion/styled framer-motion"** (Chakra UI components. n.d.a).

In figure 20 is shown how the Chakra library is used in code.



```
import * as React from 'react'

// 1. import `ChakraProvider` component
import { ChakraProvider } from '@chakra-ui/react'

function App() {
  // 2. Wrap ChakraProvider at the root of your app
  return (
    <ChakraProvider>
      <TheRestOfYourApplication />
    </ChakraProvider>
  )
}
```

snappify.com

FIGURE 20. How to set up Chakra Provider (Chakra UI components. n.d.b).

5.3.2 Axios

Axios is a Javascript library to make HTTP requests in a javascript application. It provides features like request, response, error handling, interceptors, and automatic data conversion. Axios library have to be imported as shown in Figure 21 (Axios library. n.d.a).



FIGURE 21. How to use Axios

5.3.3 React Router Dom

React Router is a well-known library for handling client-side routing in React applications. This library contains components for defining routes and navigating between various views and pages. The packages Router, Route, and Switch are imported from react-router-dom in Figure 22. The application's views include the home page, product screen, single product, cart screen, login screen, registration screen, and profile screen. The application component is wrapped with the Router component to enable client-side routing, and each route is created using the Route component. The landing page's URL path is set to ("/")(AI chatbot, n.d.b).

```

import { ChakraProvider } from "@chakra-ui/react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import "./App.css";
import Navbar from "./components/Navbar";
import ProductScreen from "./screens/ProductScreen";
import CartScreen from "./screens/CartScreen";
import SingleProductScreen from "./screens/SingleProductScreen";
import Footer from "./components/Footer";
import LandingScreen from "./screens/LandingScreen";
import LoginScreen from "./screens/LoginScreen";
import RegistrationScreen from "./screens/RegistrationScreen";
import ProfileScreen from "./screens/ProfileScreen";

function App() {
  return (
    <ChakraProvider>
      <Router>
        <Navbar />
        <main>
          <Routes>
            <Route path="/" element={<LandingScreen />}></Route>
            <Route path="/products" element={<ProductScreen />}></Route>
            <Route
              path="/product/:id"
              element={<SingleProductScreen />}
            ></Route>
            { /* <Route
              path="/product/:id"
              element={<SingleProductScreen />}
            ></Route> */ }
            <Route path="/cart" element={<CartScreen />}></Route>
            <Route path="/login" element={<LoginScreen />}></Route>
            <Route path="/registration" element={<RegistrationScreen />}></Route>
            <Route path="/profile" element={<ProfileScreen />}></Route>
          </Routes>
        </main>
        <Footer />
      </Router>
    </ChakraProvider>
  );
}

export default App;

```

FIGURE 22. How to use React Router DOM

5.3.4 React Icon

React Icon is a widely used library in React applications. It provides icon elements that must be imported for use in the application. In the Figure 23 example, the FaGithub library is imported from react-icons/fa to show up the GitHub icon inside a link.

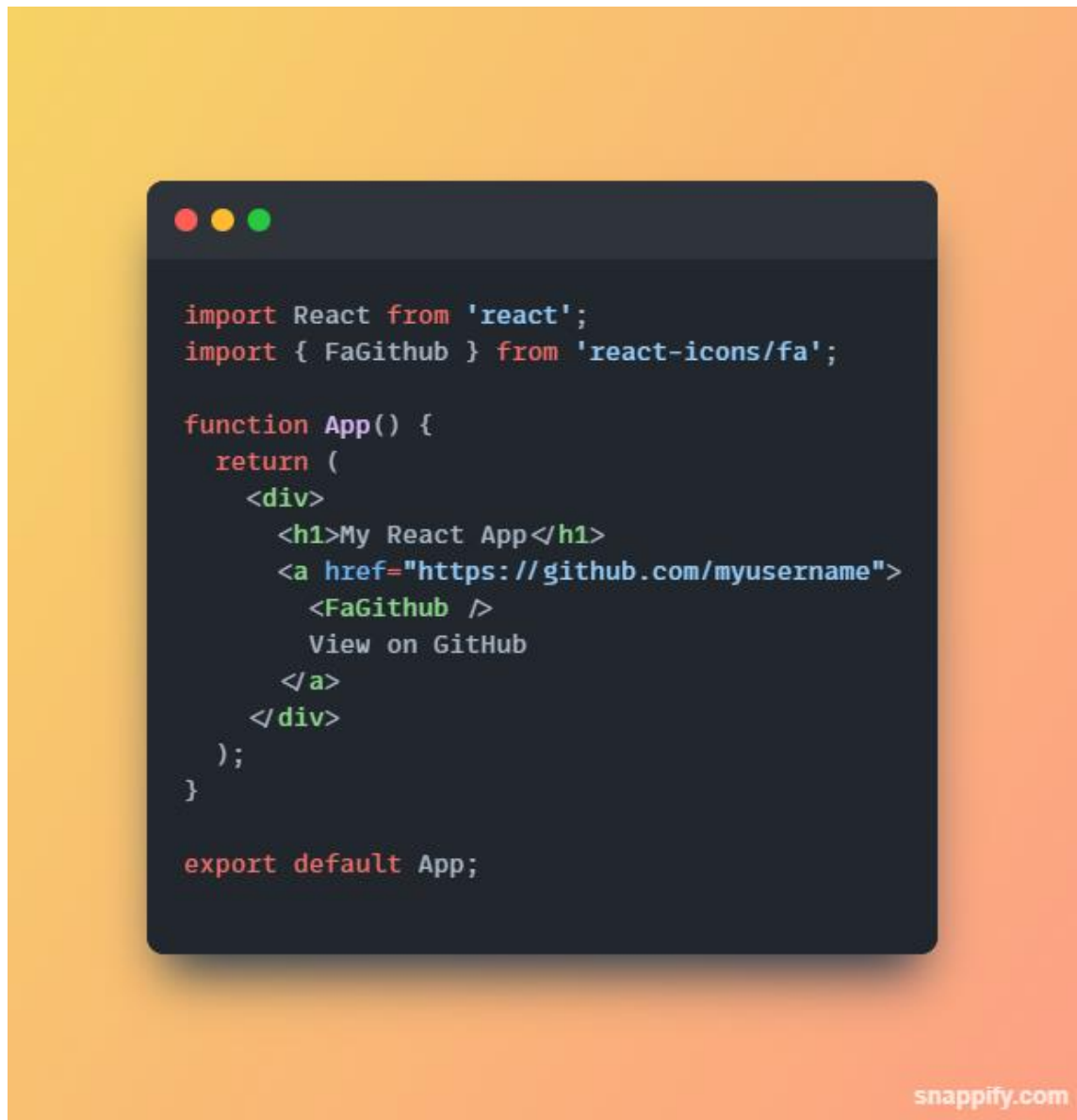


FIGURE 23. How to use React Icon

5.4 Challenges faced during implementation

During the implementation process, several challenges were encountered. The most difficult challenge was the 431 large request error, which occurred when an HTTP request was sent to retrieve data from the database, and the response indicated that the request was too large and could not be loaded. It took more than a day to resolve this error. Another major challenge was making the web application responsive.

5.5 Deployment with Vercel

Deployment is a critical step that enables a website to be used online. Vercel is the chosen service provider for deploying the website, and there are two systems for deploying a project. The first method involves importing the project folder directly to Vercel, while the second involves connecting GitLab or GitHub with Vercel and deploying it from there. When using GitHub, deployment occurs automatically when code is pushed to the repository. Each push initiates an automatic deployment.

6 Responsive

Responsive web design means adjusting layout to the size and orientation of the device. A responsive web application ensures that the application is user-friendly and easily readable according to the device size. There are a lot of ways to make a website responsive. The most common responsive way for beginners is through media queries. In this application, responsiveness is handled by Charka UI components. Responsive design is a default basic requirement for modern web applications.

6.1 Mobile vs desktop

A large desktop screen should be 1200px or more, while medium displays should be 992px to 1199px and tiny screens should be 768px to 991px. Common screen sizes for mobile devices are 414px and higher for large, 375px to 413px for medium and 320px to 374px for small. The Tradetreetop program is mobile and desktop responsive, making it usable on any device. A menu icon is produced in the mobile view that, when clicked, allows the user to access the sites and sign-up button.

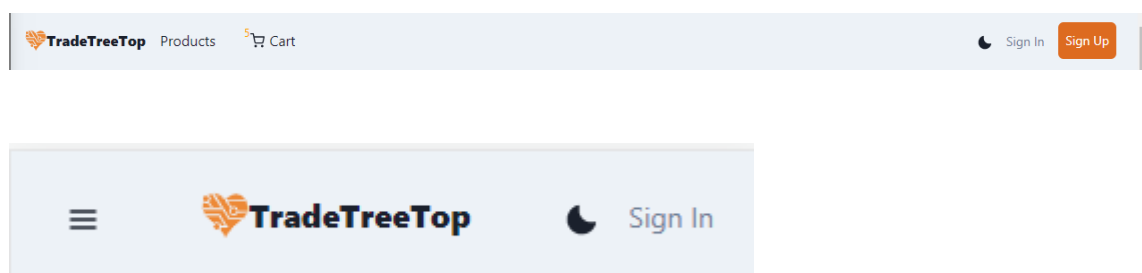


FIGURE 24. Navbar view for desktop and mobile

On desktop five products are shown in one column but on mobile, only one product is showed on the screen, and the others below. Both product views are user-friendly.

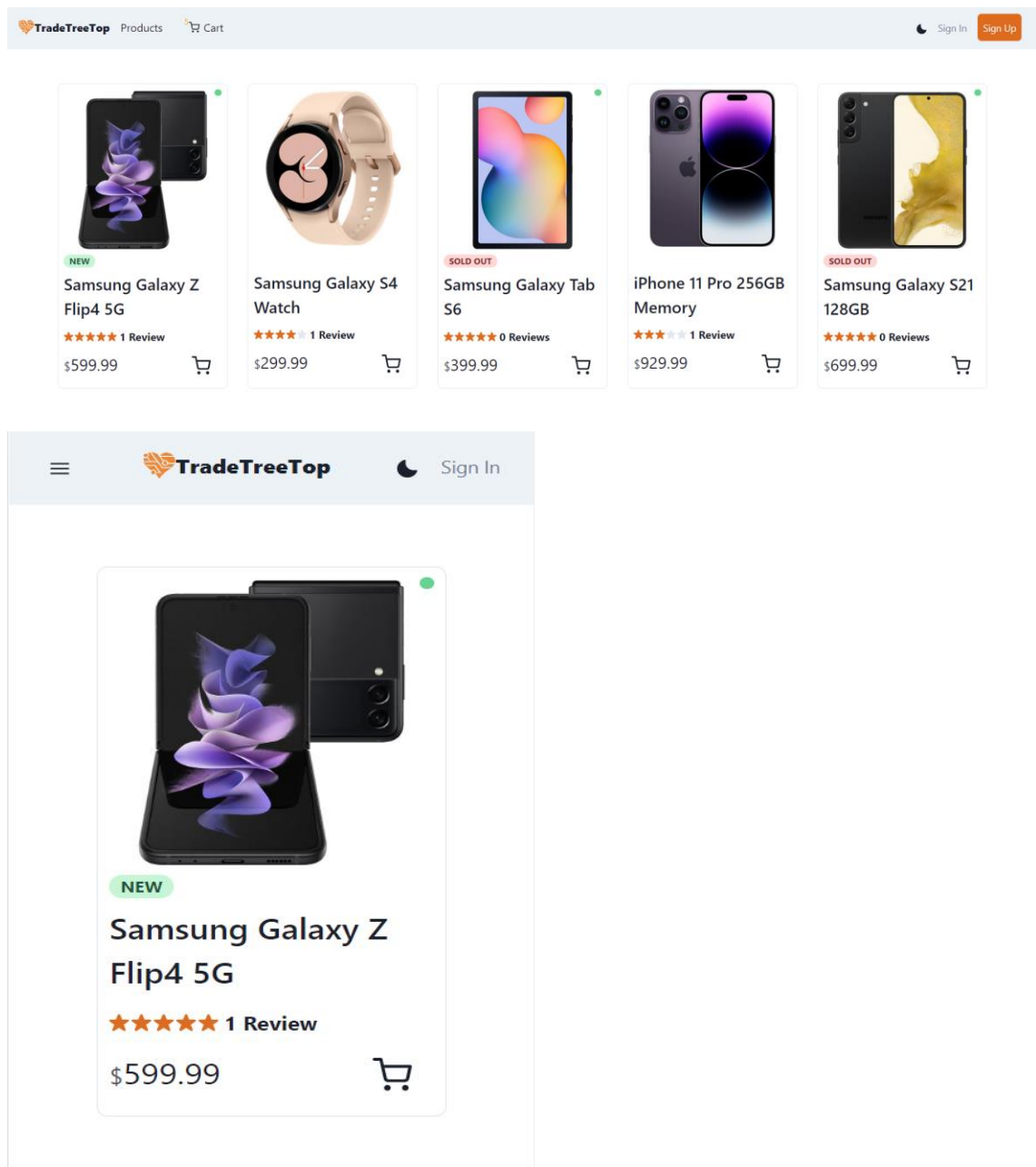





FIGURE 25. Product page views for desktop and mobile

In the cart page for the desktop view cart item and order summary are shown in different columns. From the mobile view, both cart items and order summary are in the same column. Both views are responsive.

Shopping Cart (5 Items)

	Samsung Galaxy Z Flip4 5G	1	\$599.99	X
	Samsung Galaxy Tab S6		\$399.99	X
	iPhone 11 Pro 256GB Memory	1	\$929.99	X

Order Summary

Subtotal \$3329.95

Shipping \$ FREE

Total \$ 3329.95

[Checkout →](#)

or [Continue Shopping](#)

Shopping Cart (1 item)



Apple Watch

1

\$699.99

X

Order Summary

Subtotal \$699.99

Shipping \$ 4.99

Total \$ 704.98

[Checkout →](#)

or [Continue Shopping](#)

FIGURE 26. Cart page views for desktop and mobile

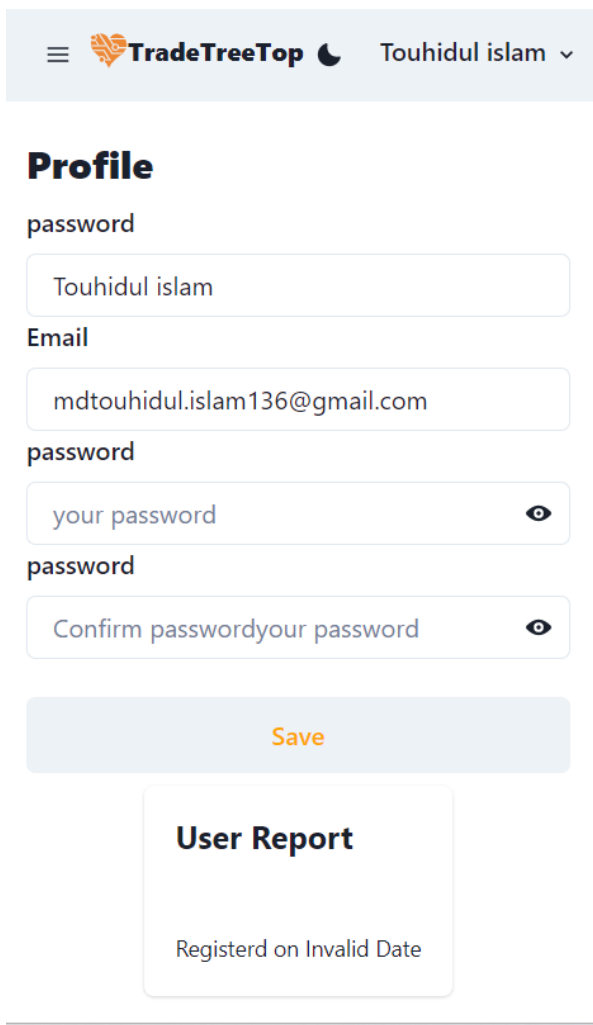
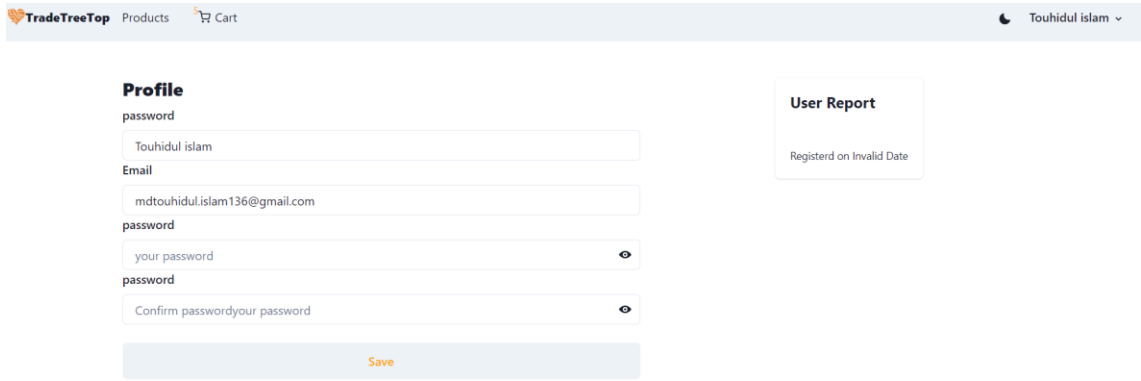


FIGURE 26. User profile view desktop and mobile

7 DISCUSSION

The TradeTreTop is an e-commerce platform that offers online shoppers an easy and user-friendly experience. The platform provides a diverse range of products in a variety of categories to meet the demands of varied users. To ensure that users can visit the website conveniently from any device, it is developed with a responsive and mobile-friendly layout. To develop this application MERN stack technology is used. The advantage of this technology approach is that the e-commerce business has full freedom to tailor the site exactly as they wish. Downside is the maintenance “burden”. It has to be updated, at least security updates have to be applied regularly.

Using a SaaS solution on the other hand is easy and probably more affordable for small company but limits tailoring and tuning possibilities. Instead of React for instance Vue would have been used. Instead of implementing own backend, Firebase could have been employed as a pre-built backend solution. There are always many alternatives and they each have different, benefits and tradeoffs.

This project's implementation has gone through several phases, including planning, designing, developing, and testing the application. The usage of current technologies such as React, Chakra UI, and MongoDB has resulted in a solid, scalable, and efficient e-commerce platform. The project has met its initial goals, and the result is a fully functional application that can meet the demands of online buyers.

REFERENCES

Axios library. n.d. Promised-based HTTP request using Axios. Read on 19.04.2023. <https://axios-http.com/docs/intro>

Big Ecommerce. n.d. Type of e-commerce business models. Read on 14.03.2023. <https://www.bigcommerce.com/articles/ecommerce/types-of-business-models/>

Chakra UI components. n.d. Chakra UI components for design and responsive web applications. Read on 21.03.2023 <https://chakra-ui.com/docs/components>

Dilmegani, C. 2022. E-commerce technologies. Published 20.09.2022. Read on 14.03.2023. <https://research.aimultiple.com/ecommerce-technology/>

De Roy, S. 2022. Redux store, actions, and reducers activities in the application. Published 27.07.2022. Read on 17.04.2023. <https://www.freecodecamp.org/news/what-is-redux-store-actions-reducers-explained/>

E-commerce theory. 2021. Details of the e-commerce web application. Understanding e-commerce applications stem and building theory. Published 13.05.2021. Read on 20.03.2023. <https://mailchimp.com/marketing-glossary/ecommerce-website/>

Ecommerce framework. n.d. Selecting an e-commerce platform and framework. Read on 21.03.2023. <https://www.bigcommerce.com/articles/ecommerce-website-development/ecommerce-frameworks/#-best-ecommerce-frameworks>

Ho, C. 2016. Middleware pattern. Published 20.06.2016. Read on 17.04.2023. <https://dzone.com/articles/understanding-middleware-pattern-in-expressjs>

HTTP methods. n.d. Express.js http request. Read on 25.04.2023. https://en.wikipedia.org/wiki/HTTP#Request_methods

Jane, S. n.d. Traditional e-commerce and ecommerce. Read on 17.04.2023. <https://www.geeksforgeeks.org/difference-between-traditional-commerce-and-ecommerce/>

Lim, D. 2020. Design an e-commerce database example. Published 19.11.2020. Read on 25.03.2023. <https://fabric.inc/blog/ecommerce-database-design-example/>

Learn e-commerce, n.d. E-commerce overview. Read on 30.04.2023. <https://www.tutorialspoint.com/e-commerce/e-commerce-overview.htm>

Mcfarland, C. 2022. E-commerce profit community. Difference between c2c and b2c e-commerce. Published 04.02.2022. Read on 14.03.2023.

<https://beproofit.co/a/community/business-management/what-is-the-difference-between-c2c-and-b2c-e-commerce>

MongoDB. n.d. Release date of MongoDB. Read on 12.05.2023.

<https://en.wikipedia.org/wiki/MongoDB>

NodeJS installation. n.d. React app setup and create a project. Read on 22.03.2023. <https://www.pragimtech.com/blog/reactjs/install-reactjs/>

NodeJS history. n.d. Introduce node.js and overview of node.js. Read on 15.04.2023.

https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm

OpenAI ChatGPT 2023a. Response to a question about functional requirements and use cases. Used 14.03.2023. <https://chat.openai.com/chat>

OpenAI ChatGPT 2023b. Response to a question about use case of e-commerce application. Used 14.03.2023. <https://chat.openai.com/chat>

React create app. n.d. How to create a react application. Read on 22.03.2023.

<https://legacy.reactjs.org/docs/create-a-new-react-app.html>

Redux, n.d. Redux published date. Read on 17.04.2023.

[https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library))

Sufiyan, T. 2023. Details of node.js, use of node.js, and guide. Published 0.03.2023. Read on 14.03.2023.

<https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>

Stack difference. n.d. About different stacks like MERN and MEAN. Read on 15.04.2023.

<https://www.geeksforgeeks.org/difference-between-mean-stack-and-mern-stack/>

Shopping with Klarna, n.d. Buy now pay later service. Read on 24.04.2023.

<https://www.klarna.com/uk/smooth/>