OULU UNIVERSITY OF
APPLIED SCIENCES

Thinh Nguyen

ONLINE T-SHIRTS STORE: E-COMMERCE WEB APPLICATION

ONLINE T-SHIRTS STORE: E-COMMERCE WEB APPLICATION

Thinh Nguyen
Bachelor's Thesis
Autumn 2022
Degree in Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Bachelor's Degree in Information Technology

Author(s): Thinh Nguyen
Title of the bachelor's thesis: Online T-shirts Store: E-Commerce Web Application
Supervisor(s): Meija Lohiniva
Term and year of completion: Autumn 2022 Number of pages: 58

In this thesis, the T-shirt E-commerce site was built to provide customers with various T-shirts for purchase. In order to facilitate online purchases, a shopping cart was provided for users. The T-shirt web application did not have the back-end and front-end parts. This thesis project aimed to improve the user experience so customers can do basic operations for T-shirt e-commerce stores via a UI.

The theoretical section discussed a construction method for a web application based on the combination of front-end, back-end and database. The knowledge of construction that creates a web application will be applied to create a T-shirt-related application that will be presented in the empirical part.

Technologies used in this project included React.js, Tailwind CSS, Node.js, Express.js, MongoDB, Postman and Visual Studio Code. Firstly, to learn more about the Development Environment, the repository was cloned, extended needed extensions in Visual Studio Code and set up the database MongoDB. In addition, the application functions were built for the client. Last but not least, the UI communicated with the back-end to transfer data from the database

This project aimed to develop a simple website that provides a consumer with a shopping cart application and to learn about the technologies necessary to create such an application. In order to develop and deploy an e-commerce website, many underlying technologies will be covered in this article.

Keywords: E-Commerce, T-shirts, React.js, Node.js, Web Application.

PREFACE

First, I would like to express my utmost and most heartfelt appreciation to Mrs Meija Lohiniva, the teaching professor and thesis adviser, for guiding and counselling me during this work. I could not have done it without her. The teacher's tough academic approach, vast knowledge, and unselfish devotion motivated me to work harder in school. Her kind assistance and resources were invaluable to me throughout the process of writing my thesis. It is possible that the thesis would not have been completed adequately if she had not provided the directions.

I want to thank my project manager at the firm, Mrs Suvi Tolonen, for giving me a chance, the most excellent advice, the wealth of information, and the creative ideas. The thesis might not have been finished without his guidance.

Additionally, I want to thank my team at the Keyloop Company, which includes Ha Quyen, and Dmitri Tkachev, for the countless fruitful conversations and meetings that provided me with invaluable guidance and support. It was a pleasure working with the crew.
Last but not least, I would like to sincerely thank my parents for their unending love, unwavering support, inspiration, and direction in all aspects of life. They provided me with several opportunities to engage with current cutting-edge technologies.

Oulu, 11.10.2022
Thinh Nguyen

CONTENTS

VOCABULARY

API	Application Data Processing
B2C	Business to Consumer
CLI	Command-Line Interface
CSS	Cascading Style Sheet
DB	Database
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identification
I/O	Input/Output
IP	Internet Protocol addresses
JS	JavaScript
JSX	JavaScript XML
JWT	JSON Web Tokens
NPM	Node Package Manager
ODM	Object Data Modeling
PaaS	Platform as a service
REST	Representational state transfer
UI	User Interface
URL	Uniform Resource Locators
VS	Visual Studio

1 INTRODUCTION

1.1 Introduction

The Industrial Revolution 4.0 has become the driving force for the world's e-commerce, especially cross-border e-commerce activities, attracting the participation of many people. Cross-border e-commerce is rapidly becoming a core element of the global economy and an inevitable trend no country can stand aside. In recent years, the e-commerce market has expanded and has become a popular business method known to businesses and people. The diversity of operating models, participants, operational processes and supply chains of goods and services, with the support of Internet infrastructure and modern technology, has brought e-commerce to the world and become an essential pillar in developing the country's digital economy. (1)

The word "e-commerce," commonly known as "electronic commerce" or "internet commerce," is used to exchange for online transactions, money and data needed. The term "e-commerce" is frequently used to refer to the online sale of tangible goods, but it may also apply to any business deal made possible by the Internet. Through online merchants and marketplaces, e-commerce has developed to simplify finding and buying things. E-commerce has benefitted independent contractors, small enterprises, and big businesses by allowing them to offer their products and services on a scale that was not feasible with conventional offline shopping. By 2020, global retail e-commerce revenues were anticipated to total $27 trillion. As a result, there have been many e-commerce web applications on the internet recently, especially e-commerce models called Business to Consumer (B2C), i.e. when a company offers a product or service to a single customer (e.g. You buy clothes from an online retailer) (2). Therefore this thesis aims to build an e-commerce web application with the type B2C where customers can see diverse, trendy local brand clothes and then choose the product they want to buy.

The reasons why E-commerce is more preferred over other typical shopping ways are as follows:

Shopping at Home: Customers will not have to spend their weekends or nights going to several places to conduct errands if they purchase online. From the comfort of their sofa, customers may purchase goods from any location in the nation and occasionally abroad.

Shop at Any Time: Customers may still shop online day or night if they have a flexible schedule or if they are just really busy. Even though most establishments close at night, a website may be accessible 24/7.

Product Information: If customers buy something online, they may read user reviews, look at comparable items on other websites to see if there are less expensive choices, read the product description, and look at any warranty information. (3)

With the significant growth of e-commerce web applications with the type B2C in recent years, fashion is a major item in the most popular things. I noticed my enthusiasm for the fashion-related subject, and I decided to investigate more to learn how it operates and how to construct a T-shirt web application. The more I researched this subject, the more I realized that various technology languages could be used to build a website for t-shirt sales. I found that each approach offers its advantages, which made selecting one approach challenging. After careful consideration of the topic of this thesis, ReactJS and TailwindCSS were selected to develop the front-end, NodeJS and ExpressJS section for the back-end section, and last but no less critical is the selected database as the database MongoDB.

2 THEORETICAL BACKGROUND

2.1 Web Application

Web Application (Web App) or Web Application is a computer application that often operates with the help of a web browser and web technologies to perform various internet-based functions. The Web Application is typically hosted on a remote server and may be accessed by the user using web browser software. Web Applications may be built for various purposes and can be used by any person or business. Web Applications may be accessible from any location through a web browser such as Microsoft Explorer, Google Chrome, or Apple Safari. (4)

Web applications are usually coded in a language supported by the browser, such as JavaScript and HTML, because these languages rely on the browser to render the executable. Some dynamic applications require server-side processing, while static applications will not need server-side processing. Web applications require a web server to manage client requests, an application server to perform the requested tasks, and sometimes a database to store information. Application server technologies range from ASP.NET, ASP and ColdFusion to PHP and JSP. (4)

2.2 Front-end

Client-side development is a crucial process element, but it encompasses various technologies. Front-end development creates the visible portion of a website that interacts directly with the user. The aim was to create an application that works on the website and can work flexibly on smaller devices, so ReactJs and Tailwind CSS were chosen for the front-end.

This project will use ReactJs since it is a highly demanding framework among developers and is well-known for its effectiveness in creating user interfaces for application software. ReactJS allows developers to break down complex UI structures into independent components. JSX allows embedding HTML and

Javascript code. ReactJS makes it easier to write Javascript code because it uses a special syntax that is the JSX syntax. (12)

Tailwind CSS is a utility-first CSS framework designed to help people build applications fast and effortlessly. Users do not need to leave their HTML or write a single line of custom CSS to create a unique component design since utility classes may govern the layout, colour, spacing, typography, and other aspects. (14)

2.3 Back-end

Back-end programming is server-side programming used for data processing, storage, and delivery to the front-end. Typically, the back-end consists of a database (database) and programs (application or service) operating on one or more servers and communicating with one another. Today, apps on the back-end comprise the "backbone" of all front-end applications. Since then, a back-end programmer has been creating database-system-communicating applications and services. Back-end applications are often a web server, an API, or a data storage or processing service. (5) The language used for back-end development is Express, a framework built on top of Nodejs. It provides powerful features for web or mobile development. Expressjs supports HTTP and middleware methods creating a potent and easy-to-use API

2.3.1 Web Server

Web servers are programs that are deployed to serve web applications, hold all data, and assume control. A web server, also known as a web server, is interconnected and connected to an extensive computer network. The web server may respond to queries from a web browser using HTTP or another protocol. Today's most popular web servers are Apache, Nginx, and Iis.

A web server is a machine that stores and sends the files that comprise a website (such as HTML, pictures, CSS, and javascript files) to the end user.

The web server is linked to the internet and may be accessed through a domain name such as mozilla.org.

An HTTP server is software that understands URLs (web addresses) and HTTP (the way your browser displays web pages). At the most basic level, any browser needs a host file on a web server, and that browser will request that file via HTTP. The web server consists of several sections that control web user access to the host's file and at least one HTTP server. When a request is sent to the correct web server address, the HTTP server sends back a request via HTTP. (6)

2.3.2 Database

DBMS stands for Database Management System. DBMS is software designed to define, perform operations, retrieve and manage data in a Database. DBMSs can usually manipulate the data itself, data formats, field names, record structures, and file structures. It also defines rules for validating and manipulating data. DBMS allows users to edit/delete/add data without needing framework programs. Query programming languages like SQL often come with DBMSs to make it easy for programmers to interact with the data they need. (7)

MongoDB is the most popular open-source NoSQL(*) database, utilized by millions of individuals. MongoDB is a C++ application. In addition, MongoDB is a cross-platform database based on the Collection and Document ideas and offers high speed, high availability, and simple scaling. (8) This project has two main collections: users and products.

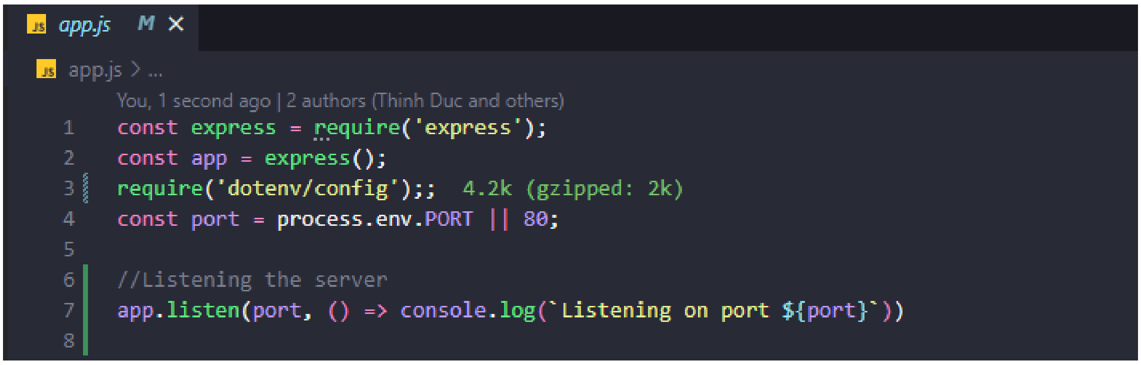
3 TECHNOLOGY

This chapter covers all the information that should be acquired before building a T-shirt e-commerce site.

3.1 Node.js

This project will utilize Node.js since it is a highly popular choice among developers nowadays and is suitable for the original concept. In addition, NodeJs is used for the back-end because it builds a restful API since Node.js is largely JS, which makes it much simpler to deal with JSON. Additionally, Node.js includes a non-blocking mechanism, which makes it an excellent choice for developing web API.

JavaScript code can be executed outside of a web browser using the open-source, cross-platform Node.js back-end runtime environment, which uses a JavaScript Engine (i.e., V8 engine). It was developed to facilitate the creation of scalable network applications. Developers may use JavaScript to create command-line tools and for server-side scripting, which produces dynamic web page content on the server before the page is transmitted to the user's web browser. The" JavaScript far and wide" approach is reflected in Node.js, which unifies web operation development around a single programming language rather than separate languages for garçon- side and customer-side scripts. The event-driven design of Node.js supports asynchronous I/O. These design decisions maximize performance and scalability for real-time web applications with many input/output activities. The OpenJS Foundation, supported by the Linux Foundation's Collaborative Projects program, is now in charge of the Node.js distributed development project, formerly overseen by the Node.js Foundation. (9). An example of a Node.js server that was created using Express can be shown in Figure 1.



```
You, 1 second ago | 2 authors (Thinh Duc and others)
const express = require('express');
const app = express();
require('dotenv/config');;  4.2k (gzipped: 2k)
const port = process.env.PORT || 80;

//Listening the server
app.listen(port, () => console.log(`Listening on port ${port}`))

```

FIGURE 1. Basic Node.js server

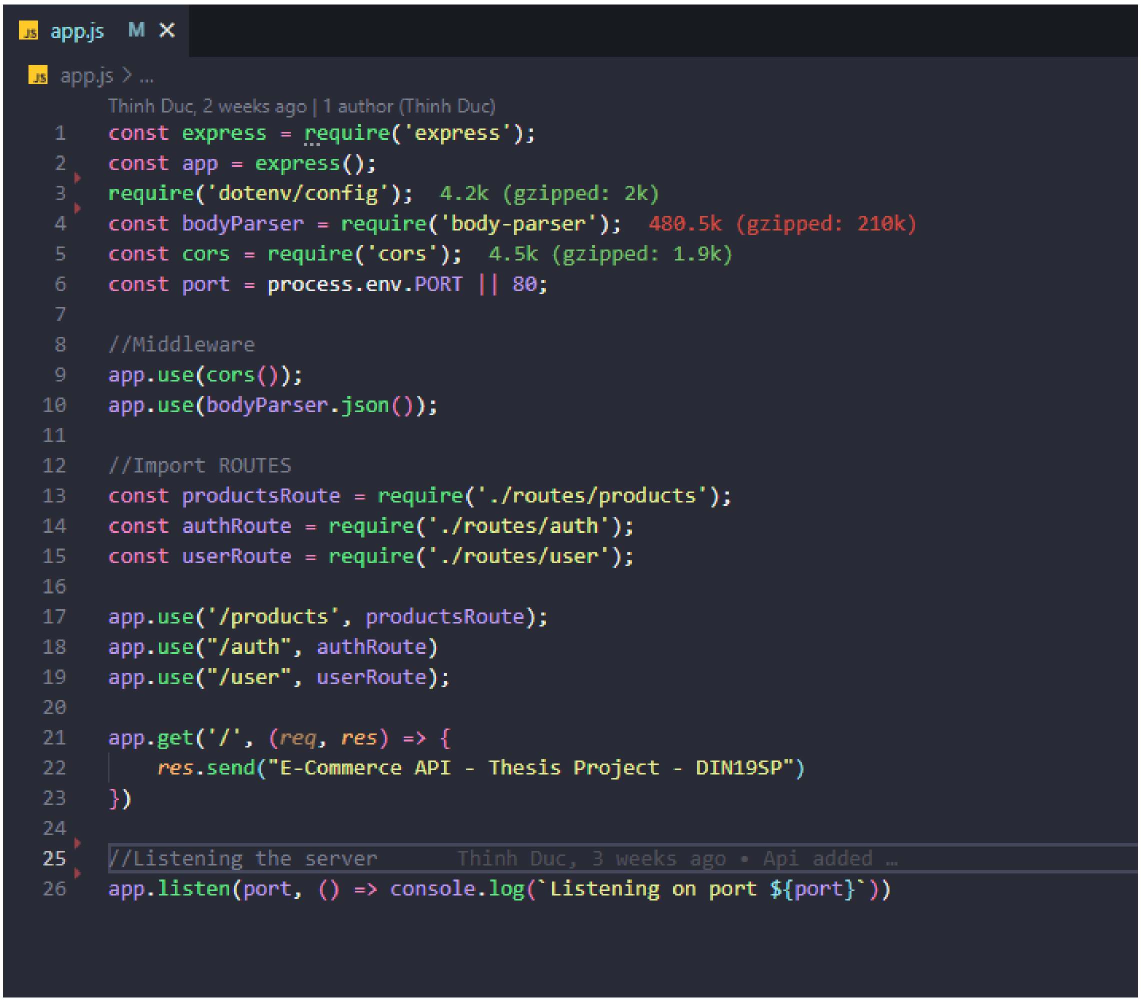
A straightforward Node.js server configuration using the Express framework is shown in Figure 1. Following establishing routes, the server begins listening to a predetermined top-secret port.

3.1.1 Express.js

Express.js is a framework that was developed atop the Node.js platform. It has significant capabilities that may be used for mobile or web development. Express.js is a compelling and user-friendly application programming interface (API) that supports HTTP and middleware methods. It establishes intermediary classes to handle the return of HTTP requests. The usage of the defined router enables the execution of multiple actions depending on the HTTP method and the URL. It provides the ability to return HTML pages depending on the parameters. (10)

Express.js is an appropriate option for this project, given that MongoDB will serve as the database. Express enables users to build the application in whatever manner developers see fit, connect to any database such as MySQL, MongoDB, or PostgreSQL, choose and utilize the view engine of their choices such as pug, ejs, handlebars, mustache, and many more. Moreover, building web servers, connecting to databases, generating dynamic HTML pages, and working with middleware are all made very easy with the help of Express JS.

The example shown in Figure 2 demonstrates how to set up express need statements, routes, and middleware.



```javascript
const express = require('express');
const app = express();
require('dotenv/config');  4.2k (gzipped: 2k)
const bodyParser = require('body-parser');  480.5k (gzipped: 210k)
const cors = require('cors');  4.5k (gzipped: 1.9k)
const port = process.env.PORT || 80;

//Middleware
app.use(cors());
app.use(bodyParser.json());

//Import ROUTES
const productsRoute = require('./routes/products');
const authRoute = require('./routes/auth');
const userRoute = require('./routes/user');

app.use('/products', productsRoute);
app.use("/auth", authRoute)
app.use("/user", userRoute);

app.get('/', (req, res) => {
    res.send("E-Commerce API - Thesis Project - DIN19SP")
})

//Listening the server      Thinh Duc, 3 weeks ago • Api added …
app.listen(port, () => console.log(`Listening on port ${port}`))
```

FIGURE 2. Preparation of an Express.js application by establishing its need statements, routes, and middleware.

First, the npm install module express command was used to load the module, just as the user would for any other NPM package. Next, the need command was run to activate the module.

The middleware's functionalities allow us to perform an action on any request and alter it before sending the answer back.

Routes provide us with the ability to write actions following the path. GET, POST, PUT, and DELETE will be the possibilities.

3.2 MongoDB

The database in this project is MongoDB because a document's organization is pretty straightforward, making it simple to read and comprehend, and there is no need to employ intricate JOIN statements. Easy data extensibility and no need to worry about data types, primary keys, or foreign keys, as is the case with SQL, and saving data on RAM, help access data faster.

MongoDB is a document-oriented NoSQL database that is used for the storing of massive volumes of data. Collections and documents are the primary data structures used by MongoDB as opposed to typical relational database tables and rows. Collections are functionally identical to relational database tables in that they store and act with collections of documents. Key-value pairs are the fundamental building block of data in MongoDB, and documents are its constituent parts.

3.2.1 The Features of MongoDB:

Each database has its collections, and those collections have their documents. Each document has the potential to be unique and may have a varied set of fields. There is room for variety in length and the subject matter of any individual text.

The document format resembles how programmers build their classes and objects using the programming languages in that they are most proficient. It is common for developers to assert that their classes do not consist of rows and columns but instead have a straightforward structure with key-value pairs.

Before they are used, a schema does not need to be specified for the rows (or documents, as they are referred to in MongoDB). Instead, it is possible to generate the fields on the fly.

The user may more easily store arrays, hierarchical relationships, and other more sophisticated structures thanks to the data model accessible inside MongoDB. This data model also enables the client to describe hierarchical relationships.

Scalability – The settings that use MongoDB are very scalable. Clusters have been defined by businesses all over the globe, with some of these businesses operating over 100 nodes and storing millions of records inside the database. (11)

Figure 3 below demonstrates the using MongoDB with Node.js.

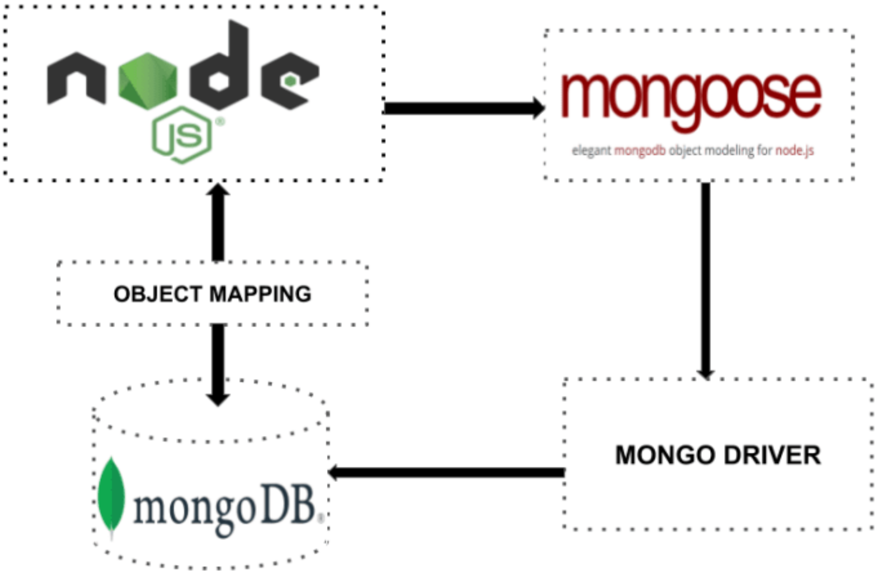


FIGURE 3. Relationships with Mongoose and Node.js (11.)

3.3 React.js

React.js is a free, open-source JavaScript library that is generally referred to as React. Building user interfaces by merging parts of code, often known as components, into whole websites is the most effective method. After being constructed by Facebook, it is currently maintained by Meta and the open-source community. Users are free to make as much or as little use of React as users see fit, which is one of the many benefits of this framework. For instance, users might use React to construct an entire website, or users could only utilize a single React component on a single page.

The programming language JSX, a hybrid of JavaScript and XML, is used to construct React.js. The JSX markup language is used to build the elements, and then JavaScript is used to display them on their website. Although it has a high learning curve for younger developers, React is swiftly becoming one of the most popular and in-demand JavaScript frameworks, regardless of a steep learning curve.

While the other solutions we will discuss today are categorized as frameworks, React is a JavaScript library rather than a framework. A library is a collection of resources that can be used to build anything. It is helpful to conceive of a library as a tool that developers might use in any project and a framework as a comprehensive design.

According to research by State Of JS, the popularity of the programming language known as React.js has experienced rapid expansion in recent years. This is partly attributable to its adaptability and rapid development pace, but it is also assisted by the fact that it is sponsored by Meta, which gives software developers and businesses a sense of security in their choice to adopt React. As a result, there is a significant increase in the need for React developers. Because of this, a diverse selection of employment opportunities is available for software engineers proficient in React. (12)

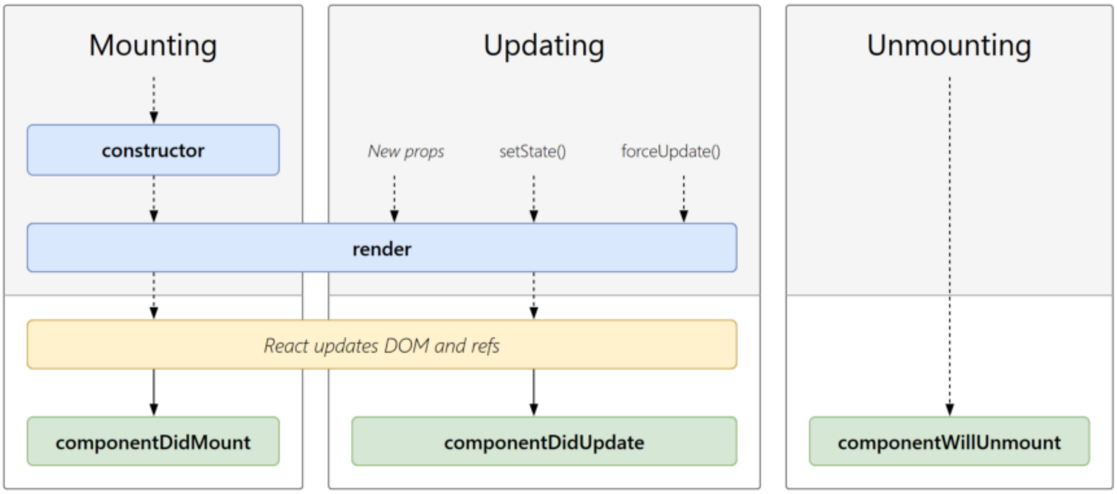


FIGURE 4. Modern diagram for the React component lifecycle. (13.)

The React library has several distinct lifecycle methods that may be called at various stages to carry out the intended tasks. These are known as "lifecycle methods" for the React component. The overarching contemporary lifetime of React components is shown in the above figure along with the proper lifecycle methods for each component. (13)

3.4 Tailwind CSS

Tailwind CSS is a utility-first CSS framework developed to allow users to construct apps more quickly and easily. Users do not have to leave their HTML or write a single line of custom CSS to build a unique component design since users can utilize utility classes to manage the layout, colour, spacing, typography, and more.

Tailwind CSS is a framework with a minimal degree of abstraction. Tailwind does not include fully stylized buttons, dropdowns, and navbars as other CSS frameworks like Materialize and Bootstrap have. Instead, it provides utility classes that let users build their reusable components from scratch.

Because of this, compared to other CSS frameworks, it offers a far greater degree of freedom and control over the appearance of the user application. This might help the developer design a more distinctive webpage. (14) Take a look at figure 5 below to see why Tailwind CSS is popular now.

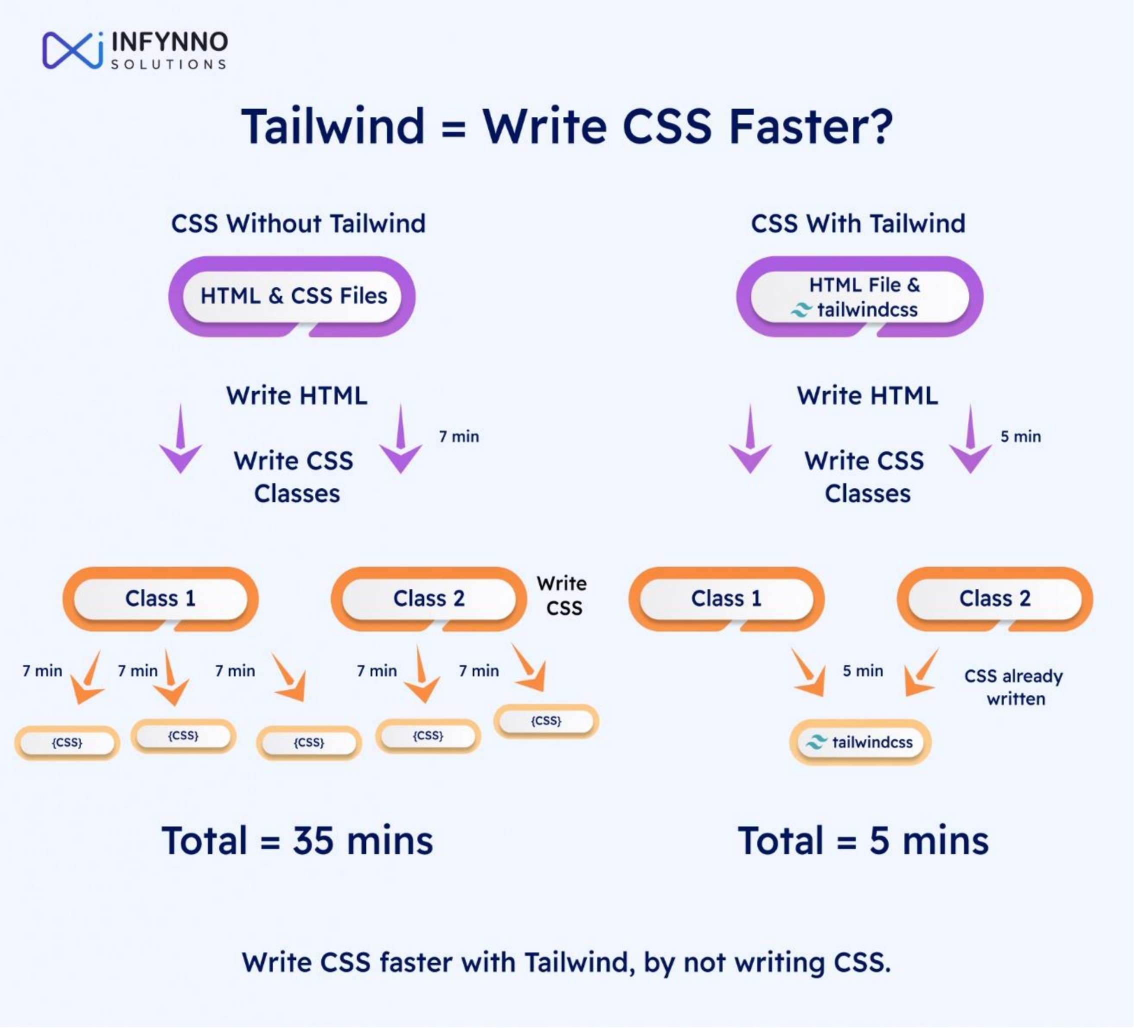


FIGURE 5. Why is Tailwind CSS Trending Now? (15.)

Writing new code and maintaining existing code for users' applications is sped significantly by using Tailwind CSS. Users who utilize this utility-first framework will not have to worry about writing any custom CSS to design for their application. They may instead utilize utility classes to handle their application's padding, margin, colour, font, and shadows. These are just some of the elements users can control.

It has come to our attention that tailwind has recently been gaining more and more popularity. Instead of being a component-based framework like Bootstrap, the Tailwind framework positions itself as a utility-based framework.

In contrast to this utility-based strategy, most designers believe that using a component-based approach is the most productive course of action that can be taken. Conversations are welcomed. Therefore, it is now at the Trending position among all CSS Frameworks. (15)

3.5 Working Environment

3.5.1 Git, GitHub

GitHub was selected as the Git code management platform. GitHub is a well-known site that offers Git source code repositories as part of its offering for software development projects. GitHub has all of Git's capabilities and social features that allow engineers to collaborate and communicate with one another. Therefore, GitHub is a more useable platform because it will not be dependable if just one firm supports the program. GitHub was responsible for managing the sharing of code between authors and companies and providing bug tracking, wiki space, and other applications for "social coding." Later in this thesis, these topic is covered in more detail.

Git is a tool for development and operations teams that manages source code. It is a version management system that is open-source and free that can effectively manage projects ranging in scale from very small to very big. Git is a tool that keeps track of the changes made to the source code. This makes it possible for numerous engineers to collaborate on non-linear development. Git was first developed in 2005 by Linus Torvalds for use in the development of the Linux kernel. (16) Take a look at figure 6 below.

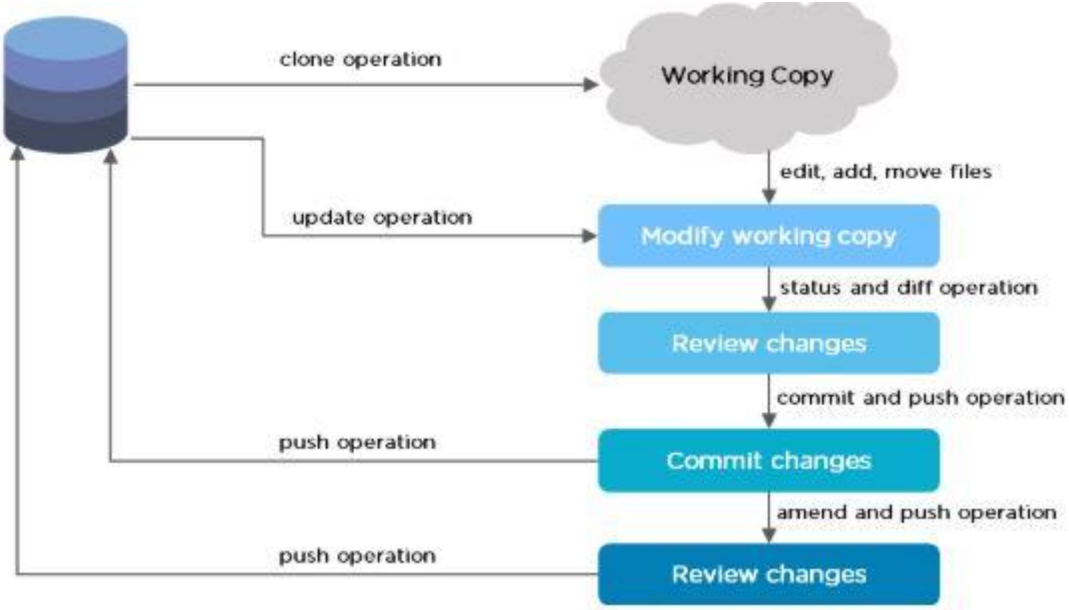


FIGURE 6. Git workflow(16.)

The workflow for Git can be split into three distinct states:

1. Working directory - makes changes to the files located in our working directory.
2. Staging area (Index) - The files will be placed in staging, and you will add snapshots to your staging area.
3. Git directory (Repository) - proceeds out all the commit operations, which will permanently save the snapshots in your Git directory. The developer may start by checking out any existing version, making changes, staging them, and then committing them. (16)

GitHub is a corporation run for profit and provides a service for hosting Git repositories in the cloud. Simply put, it makes it simpler for individuals and teams to utilize Git for collaborative version control and management. Because the interface of GitHub is so simple to use, even inexperienced programmers may benefit from using Git. If the user cannot access GitHub, utilizing Git often calls for additional technical knowledge and command-line experience. (17)

The ability to monitor and manage changes made to the code of a software project is made possible by version control. Version management becomes more important as the scope of a software project expands. Developers can securely go through the branching and merging processes thanks to version control. When using branching, a programmer will make a copy of some of the source code (called the repository). The programmer is then free to make any necessary adjustments to that code section without fear of repercussions for the rest of the project. Then, after the developer has ensured that their portion of the code functions as it should, they can merge their code back into the primary source code to make the change official.

3.5.2 Visual Studio Code

Visual Studio Code was applied to develop and debug the source code for this project**.** Visual Studio Code (VSC) is the most well-known lightweight code editor. The VSC plugins have already achieved widespread use and are indispensable to JavaScript developers. It provides a wide variety of assistance such as code snippets tailored to specific technologies, syntax highlighting, Emmet, and intelligent reminders. In addition, VSC offers a comprehensive set of tools for testing and troubleshooting.

A free source code editor called Visual Studio Code runs on the user's desktop and the web. It is available for Windows, macOS, Linux, and Raspberry Pi OS. Visual Studio Code is lightweight but powerful. It already has JavaScript, TypeScript, and Node.js support built in. Its ecosystem of programming language extensions is broad (such as C++, C#, Java, Python, PHP, and Go), runtimes (such as.NET and Unity), environments (such as Docker and Kubernetes), and clouds (such as Amazon Web Services and Microsoft Azure) (such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform).

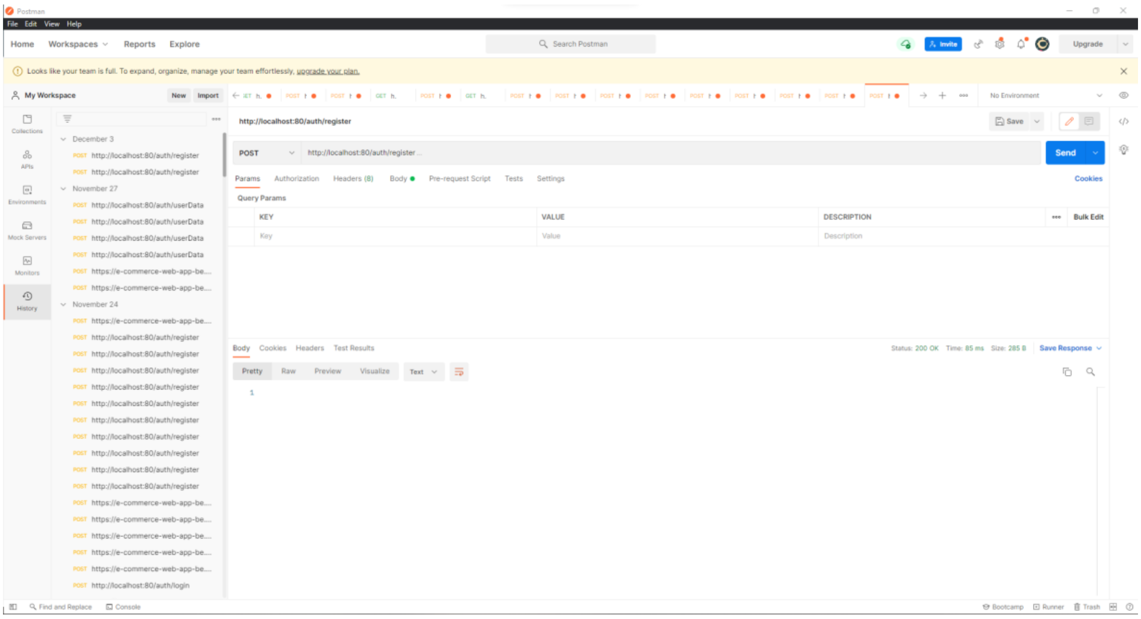
Aside from the overall concept of being lightweight and starting quickly, Visual Studio Code has IntelliSense code completion for variables, methods, and imported modules; graphical debugging; linting, multi-cursor editing, sophisticated editing tools including parameter suggestions, stylish code navigation and refactoring, and integrated source code control with support for Git. (18)

3.5.3 Postman

Postman is the tool chosen for this project to test the API. Utilization of Collections Postman developers has the ability to generate collections for usage with their API requests. Each collection can generate numerous requests and subdirectories. In addition, Debugging - Postman console helps to check what data has been retrieved, making it easy to debug the test.

Postman is now one of the most widely used tools for testing application programming interfaces (APIs). Postman allows us to control application programming interfaces (APIs), most often REST APIs. Users do not have to write a single line of code in order to make calls to the Rest API, thanks to Postman. (19)

Postman supports all HTTP methods (GET, POST, PUT, PATCH, DELETE, ...). In addition, it is possible to preserve the request history, making it extremely handy to reuse if and when required. Take a look at figure 7 below.



Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.

December 3
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
November 27
POST http://localhost:80/auth/userData
POST http://localhost:80/auth/userData
POST http://localhost:80/auth/userData
POST http://localhost:80/auth/userData
POST https://e-commerce-web-app-be....
POST https://e-commerce-web-app-be....
November 24
POST https://e-commerce-web-app-be....
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST http://localhost:80/auth/register
POST https://e-commerce-web-app-be....
POST https://e-commerce-web-app-be....
POST https://e-commerce-web-app-be....
POST https://e-commerce-web-app-be....
POST https://e-commerce-web-app-be....
POST http://localhost:80/auth/login

http://localhost:80/auth/register

POST http://localhost:80/auth/register ...

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 85 ms Size: 285 B

```
1
```

FIGURE 7. POSTMAN interface.

Postman made API testing easy, understandable, and time-saving because Postman can assist with debugging. Accordingly, thanks to Postman's dashboard, users can quickly check the exported data. As a result, the debugging process will become more flexible than ever.

4 IMPLEMENTATION

Currently, the application has progressed through the coding, build, and testing phases. An in-depth examination will be conducted in three finished phases. They are a collection of personal experiences, perspectives, challenges, and research conducted to find a solution.

All essential functionalities are user identification, adding items to a shopping cart, and data transfer and processing. When it comes to the application construction, the process would be as follows: The establishment of the database and connection to the server and application functionality.

4.1 Launch & Manage MongoDB

For MongoDB to function faultlessly and establish a connection to the back-end, many very straightforward processes need to be carried out in an exacting and cautious manner. The steps, with all of their specifics, will be outlined below.

4.1.1 Atlas setting

Through MongoDB Atlas, hosting and managing user data in the cloud, which can be accessed from any location, is a straightforward process. To begin, the user has to sign up for an Atlas account by providing their email address or the information of their Google Account. In addition, they do not have to invest any money to construct and install a cluster. The data may be hosted on the free clusters made available to the client by Atlas for use in a development environment on a smaller scale. Users get access to a fraction of Atlas's capabilities via free clusters, which never run out of available storage space. To access their cluster, users must first establish a database user. After that, they have to add the user's IP address connected to the IP access list on the developer's computer. When accessing a cluster, Atlas requires all clients to authenticate that they are already logged in as MongoDB database users. Take a look at figure 8 below.

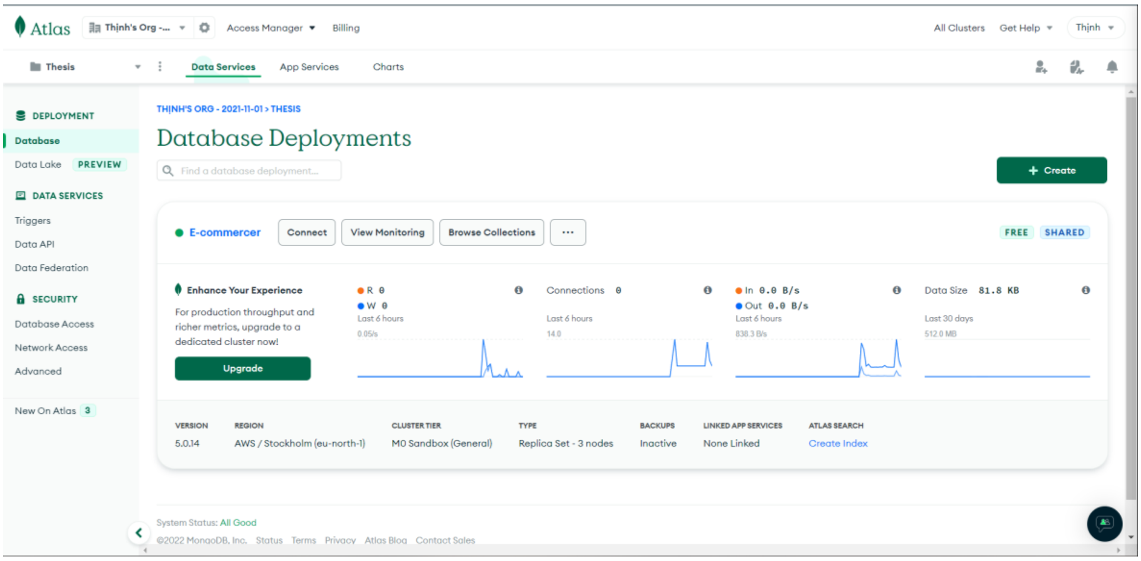


FIGURE 8. Atlas cluster interface.

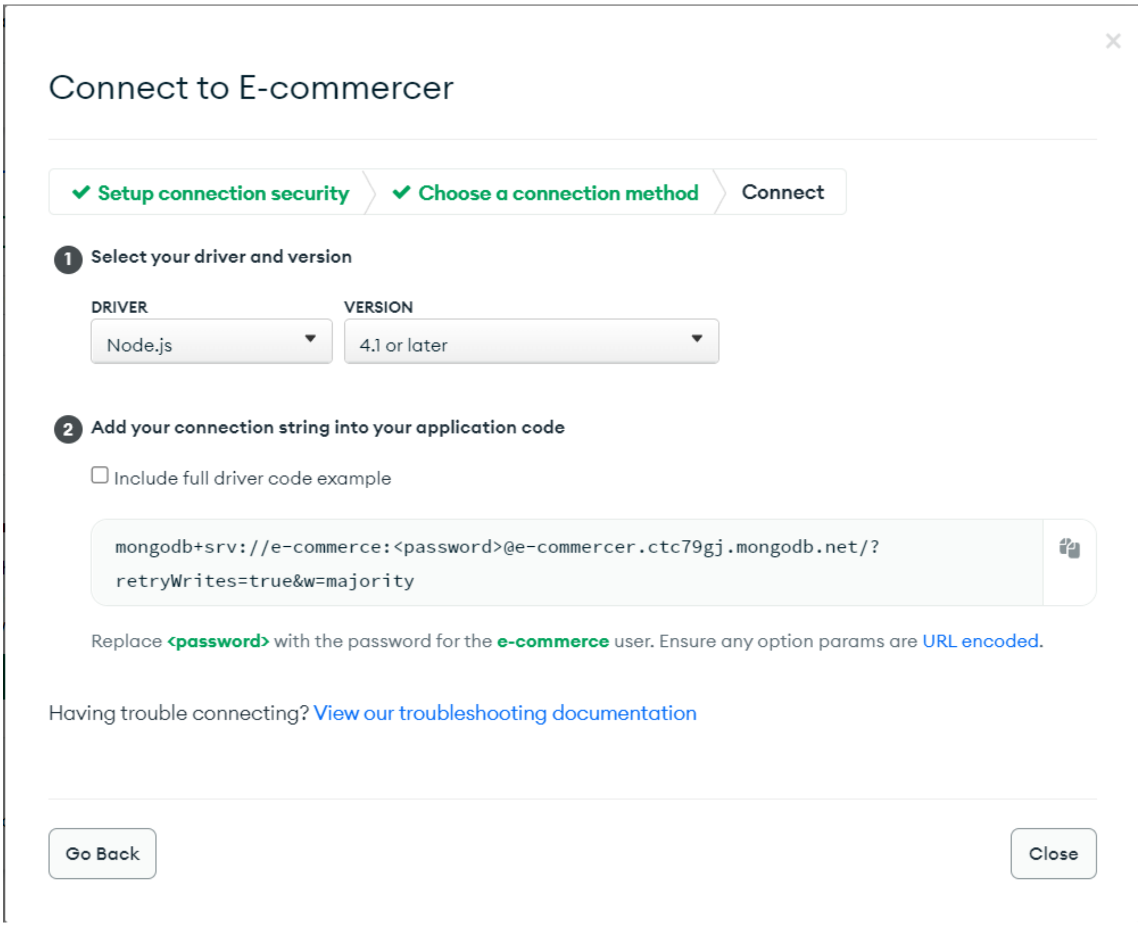
Atlas configuration is a complex step in signing up for an account and building a new database, but it was set up after much online study and tutorials.

4.1.2 Database connection method

There are four different ways to connect to the database, and they are as follows:

1. Connect with the MongoDB Shell: Use MongoDB's interactive Javascript interface to have conversations with other users of the cluster.
2. Connect the stoner operation: Establish a connection between the inventor's operation and the cluster using MongoDB's native motorists. Connect using MongoDB Compass: Utilizing MongoDB's graphical user interface, users can explore, alter, and visualize their data.
3. Connect using VS Code: Visual Studio Code supports establishing connections to MongoDB hosts.

The connect user application's second approach is the most straightforward implementation. Take a look at figure 9 below.



Connect to E-commercer

1 Select your driver and version

DRIVER: Node.js

VERSION: 4.1 or later

2 Add your connection string into your application code

□ Include full driver code example

```
mongodb+srv://e-commerce:<password>@e-commercer.ctc79gj.mongodb.net/?
retryWrites=true&w=majority
```

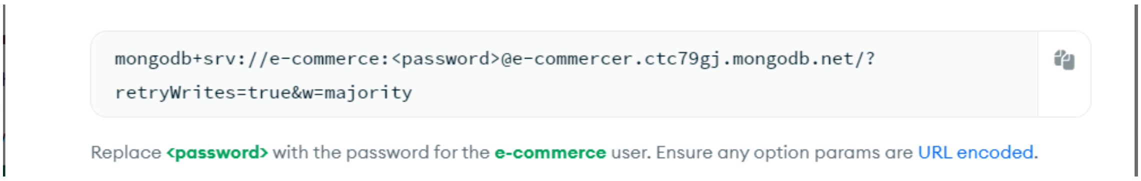
Replace **<password>** with the password for the **e-commerce** user. Ensure any option params are URL encoded.

Having trouble connecting? View our troubleshooting documentation

-

FIGURE 9. Connect application.

In order to improve the program's level of safety, a new file in Visual Studio Code with the extension.env was created and placed in FIGURE 10. In this particular file, I need to replace the password> portion needed to be replaced with the already established password.



```
mongodb+srv://e-commerce:<password>@e-commercer.ctc79gj.mongodb.net/?
retryWrites=true&w=majority
```

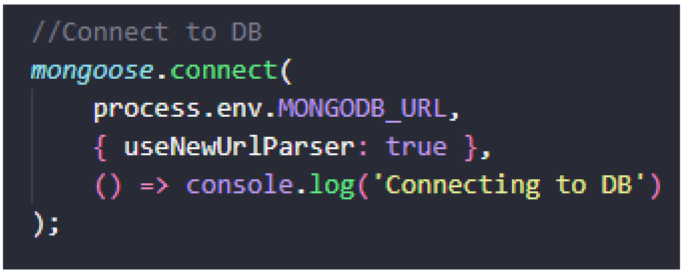
Replace **<password>** with the password for the **e-commerce** user. Ensure any option params are URL encoded.

FIGURE 10. Add the connection string to the application code.

The user's MongoDB clusters or collections are connected to the Node.js application using the Mongoose.js module. Mongoose offers many functionalities when it comes to generating schemas and dealing with them. It allows the developer to construct schemas that can be used for documents.

MongoDB is one of the No-SQL databases that is the most active in the world of software development right now. Instead of sending and retrieving data as SQL objects, No-SQL databases enable developers to transmit and receive data in JSON messages. We can use Mongoose to interact with MongoDB inside a Node.js application.

Then, to ensure that the connection worked flawlessly, I included the statement and drafted the connection method shown in FIGURE 11.



```
//Connect to DB
mongoose.connect(
    process.env.MONGODB_URL,
    { useNewUrlParser: true },
    () => console.log('Connecting to DB')
);
```

FIGURE 11. Connect to the database.

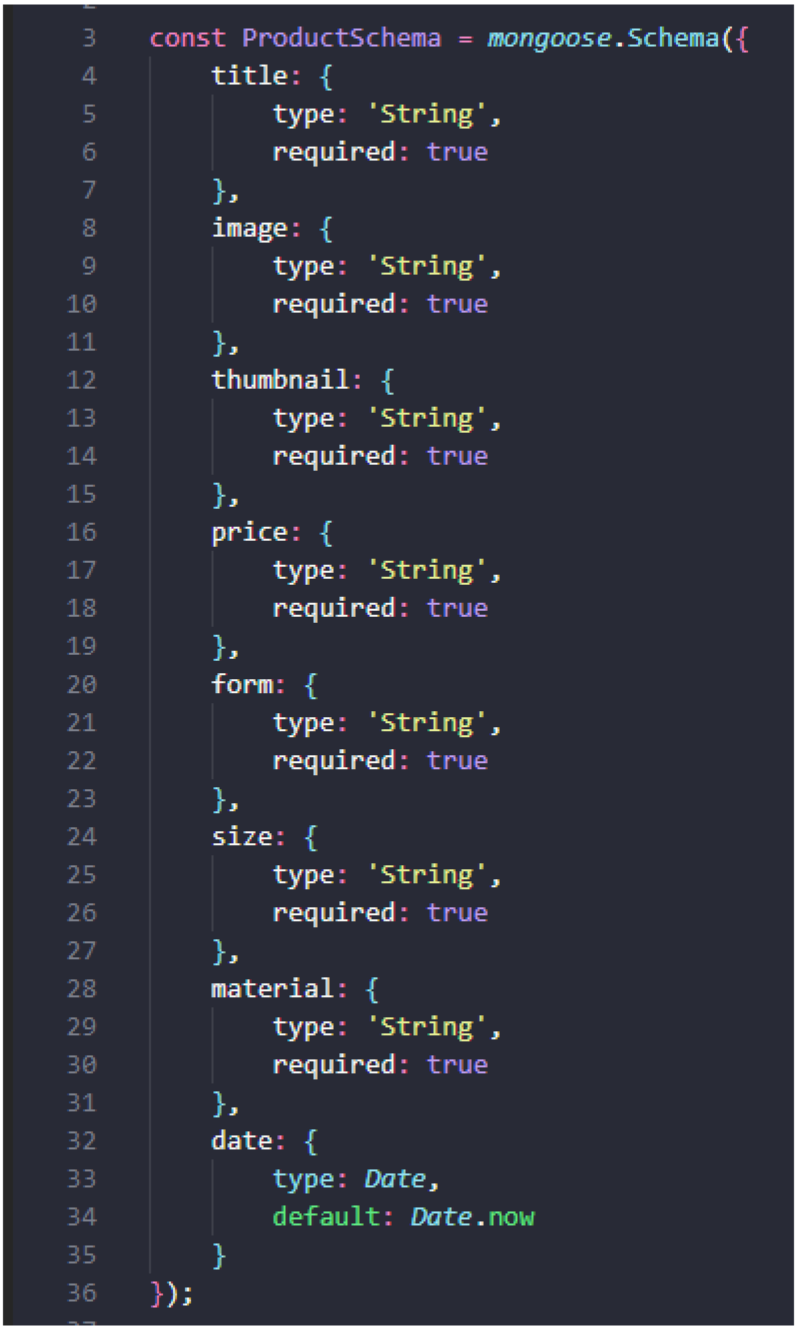
4.2 Build a RESTful API

An application programming interface (API or web API) that complies with the requirements of the REST architectural style and enables interaction with RESTful web services is referred to as a RESTful API (sometimes referred to simply as REST API). In basic terms, a RESTful API is a standard that supports the management of resources and is utilized in the design of APIs for online applications, which is known as the design of Web Services.

4.2.1 Data schema

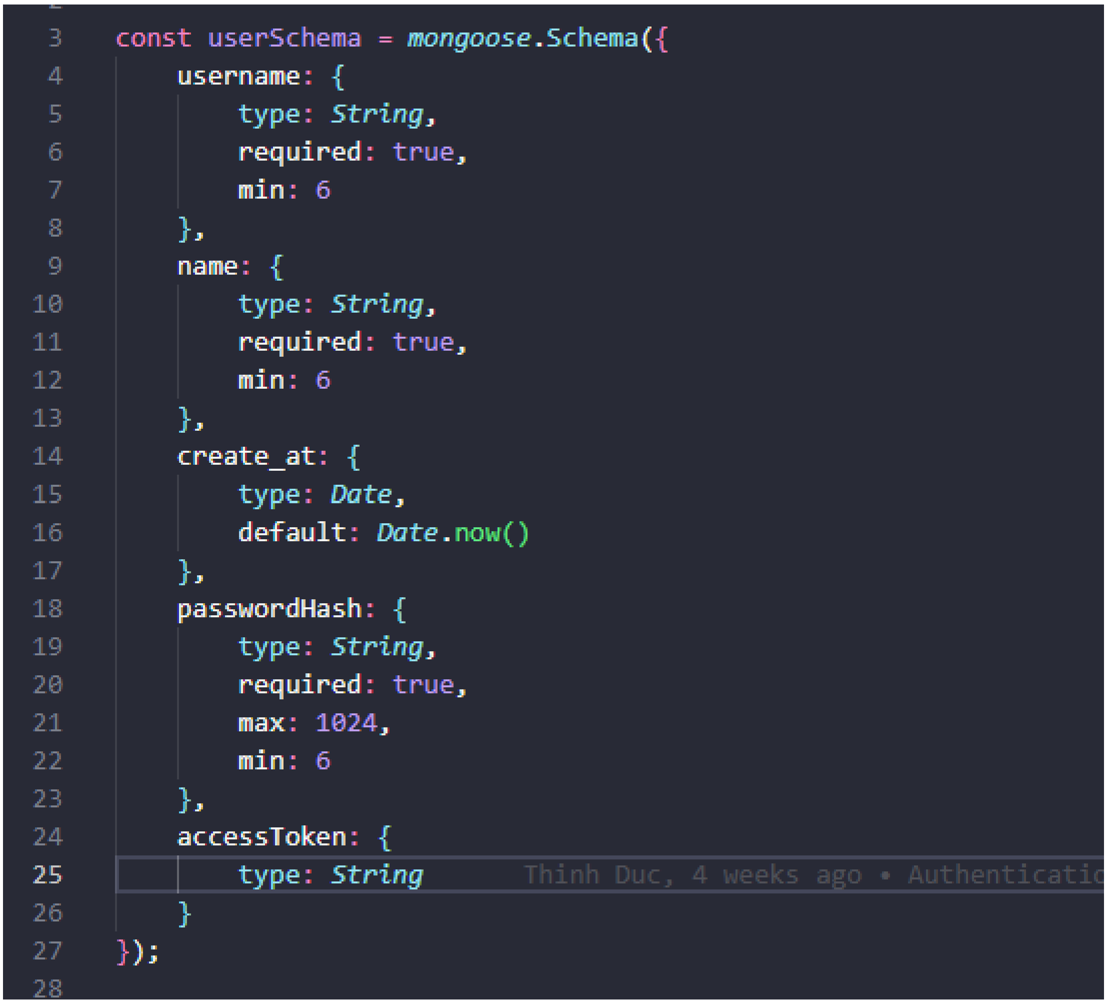
A library for Object Data Modeling (ODM) that works with MongoDB and Node.js is known as Mongoose. It validates the schema, maintains the relationships between data, and translates between objects in code and the representation of those objects in MongoDB. It also offers management of the links between the data.

MongoDB is a document database that uses NoSQL and does not have a schema. This implies that the user can store JSON documents in it, and the format of these documents is not regulated in the same way as SQL databases are. This allows for more flexibility. Creating applications may be completed more quickly using NoSQL, and the complexity of deployments can be simplified. This is one of the benefits of adopting NoSQL. Take a look at figure 12 and figure 13 below.



```
const ProductSchema = mongoose.Schema({
    title: {
        type: 'String',
        required: true
    },
    image: {
        type: 'String',
        required: true
    },
    thumbnail: {
        type: 'String',
        required: true
    },
    price: {
        type: 'String',
        required: true
    },
    form: {
        type: 'String',
        required: true
    },
    size: {
        type: 'String',
        required: true
    },
    material: {
        type: 'String',
        required: true
    },
    date: {
        type: Date,
        default: Date.now
    }
});
```

FIGURE 12. Product schema.



```
const userSchema = mongoose.Schema({
    username: {
        type: String,
        required: true,
        min: 6
    },
    name: {
        type: String,
        required: true,
        min: 6
    },
    create_at: {
        type: Date,
        default: Date.now()
    },
    passwordHash: {
        type: String,
        required: true,
        max: 1024,
        min: 6
    },
    accessToken: {
        type: String        Thinh Duc, 4 weeks ago • Authenticatio
    }
});
```

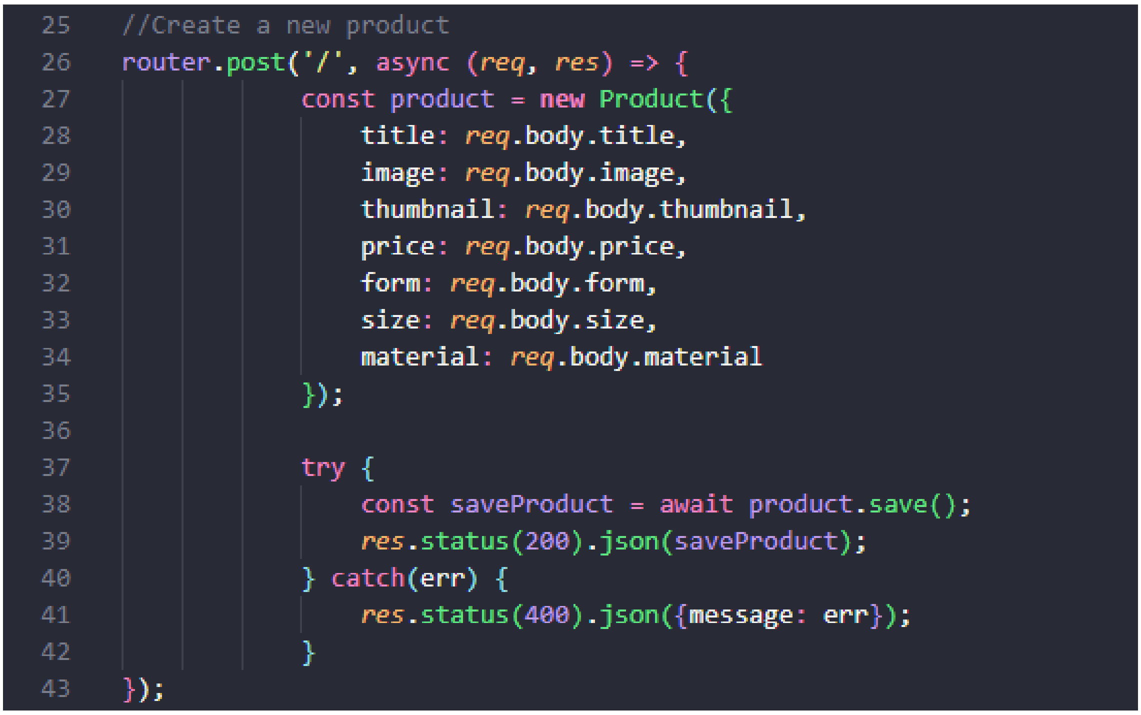
FIGURE 13. Users schema.

Figures 12 and 13 are the outcome of carefully considering the relevant user and product attributes. The data schema was created to be as similar as possible to the one that was provided by the API.

4.2.2 Restful Route

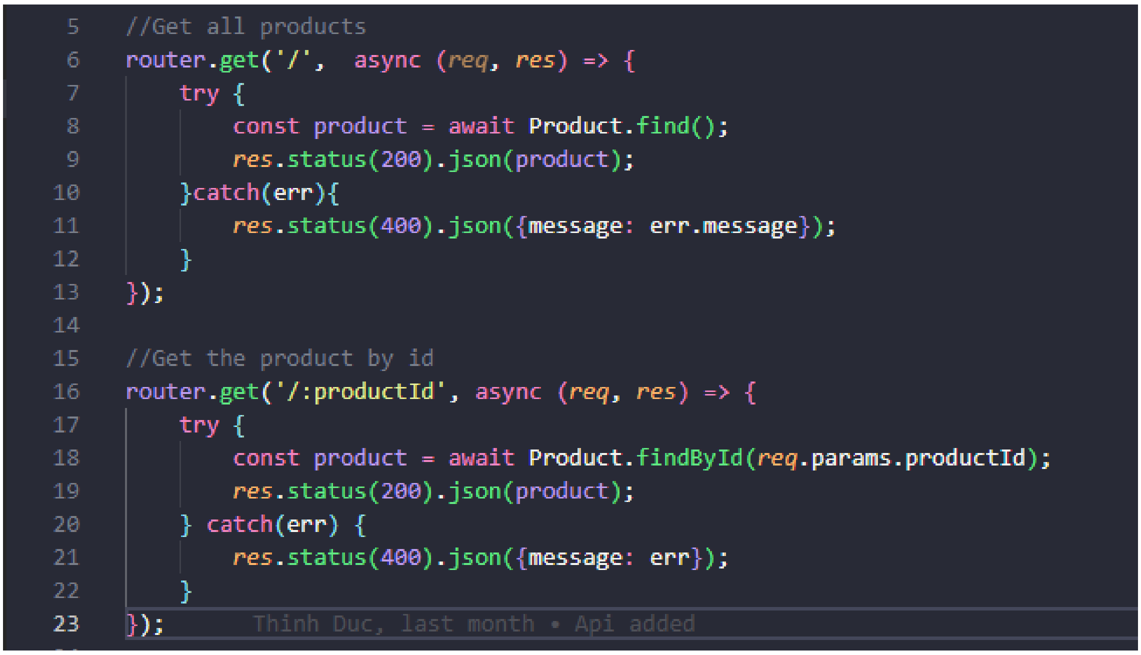
GET and POST requests make up a significant amount of the application's total use of requests. Express.js routes are used for each resource in this particular project. Each particular kind of database object will be assigned a single route URI that corresponds to the CRUD action it performs. The user and the product are the two primary paths that are taken with this project.

Product Route: Since the product is perhaps the most significant aspect of this part, the data schema stipulates that its features must be consistent with those necessary. It will not be able to develop a successful product if its characteristics are insufficient. In addition to this, CRUD activities may be carried out via this product route. Figures 14 and 15 below will make it easier to understand what product can be made and called out.



```
//Create a new product
router.post('/', async (req, res) => {
            const product = new Product({
                title: req.body.title,
                image: req.body.image,
                thumbnail: req.body.thumbnail,
                price: req.body.price,
                form: req.body.form,
                size: req.body.size,
                material: req.body.material
            });

            try {
                const saveProduct = await product.save();
                res.status(200).json(saveProduct);
            } catch(err) {
                res.status(400).json({message: err});
            }
});
```

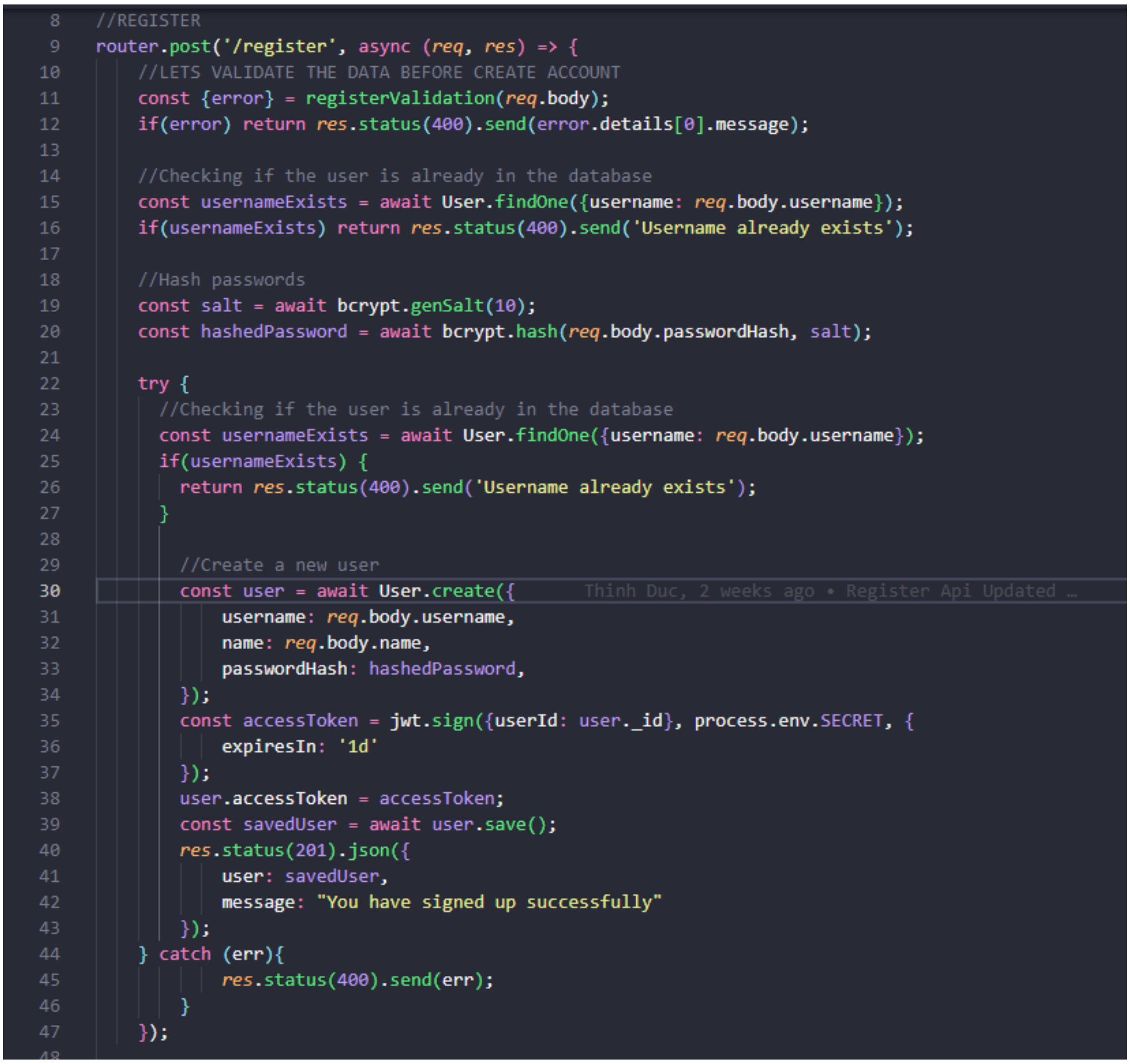
FIGURE 14. Creating a new product.



```
//Get all products
router.get('/',  async (req, res) => {
    try {
        const product = await Product.find();
        res.status(200).json(product);
    }catch(err){
        res.status(400).json({message: err.message});
    }
});

//Get the product by id
router.get('/:productId', async (req, res) => {
    try {
        const product = await Product.findById(req.params.productId);
        res.status(200).json(product);
    } catch(err) {
        res.status(400).json({message: err});
    }
});
```

FIGURE 15. Getting all products and a product by id.

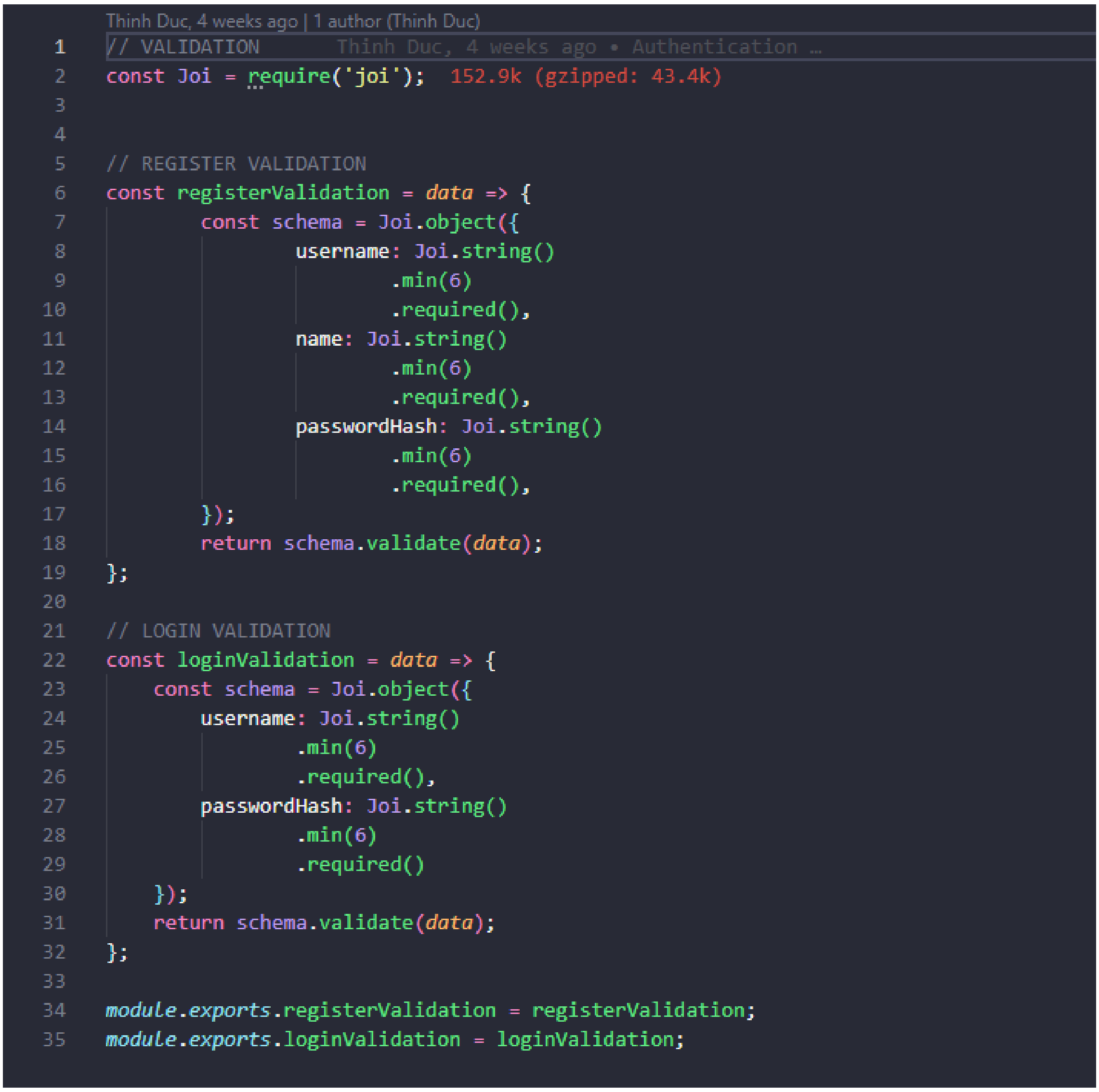
Auth Route: This route is going to create a node.js API that handles Authentication. The first step is to produce a new user, who then has to have their account approved by Joi. When doing tests with the Postman, an error will cause the program to report a status of 400 in addition to an error message. Using the findOne() function, the system can determine whether or not the username already exists. In order to locate a single document that satisfies the criterion, the findOne() method is used. If more than one document satisfies the requirement, it will return the first document to do so. Take a look at figure 16 below.



```
//REGISTER
router.post('/register', async (req, res) => {
    //LETS VALIDATE THE DATA BEFORE CREATE ACCOUNT
    const {error} = registerValidation(req.body);
    if(error) return res.status(400).send(error.details[0].message);

    //Checking if the user is already in the database
    const usernameExists = await User.findOne({username: req.body.username});
    if(usernameExists) return res.status(400).send('Username already exists');

    //Hash passwords
    const salt = await bcrypt.genSalt(10);
    const hashedPassword = await bcrypt.hash(req.body.passwordHash, salt);

    try {
      //Checking if the user is already in the database
      const usernameExists = await User.findOne({username: req.body.username});
      if(usernameExists) {
        return res.status(400).send('Username already exists');
      }

        //Create a new user
        const user = await User.create({
            username: req.body.username,
            name: req.body.name,
            passwordHash: hashedPassword,
        });
        const accessToken = jwt.sign({userId: user._id}, process.env.SECRET, {
            expiresIn: '1d'
        });
        user.accessToken = accessToken;
        const savedUser = await user.save();
        res.status(201).json({
            user: savedUser,
            message: "You have signed up successfully"
        });
    } catch (err){
            res.status(400).send(err);
        }
    });
```

FIGURE 16. Post method for registering.

The realization that the object validation server side is required for public online services led to the development of Joi, which is now the package for object schema descriptions and validation that has gained the most widespread use. Using Joi, we can draft Javascript objects, ensuring that the data we process and eventually accept is correct. Take a look at figure 17 below.



```
Thinh Duc, 4 weeks ago | 1 author (Thinh Duc)
// VALIDATION        Thinh Duc, 4 weeks ago • Authentication …
const Joi = require('joi');  152.9k (gzipped: 43.4k)


// REGISTER VALIDATION
const registerValidation = data => {
        const schema = Joi.object({
                username: Joi.string()
                        .min(6)
                        .required(),
                name: Joi.string()
                        .min(6)
                        .required(),
                passwordHash: Joi.string()
                        .min(6)
                        .required(),
        });
        return schema.validate(data);
};

// LOGIN VALIDATION
const loginValidation = data => {
    const schema = Joi.object({
        username: Joi.string()
                .min(6)
                .required(),
        passwordHash: Joi.string()
                .min(6)
                .required()
    });
    return schema.validate(data);
};

module.exports.registerValidation = registerValidation;
module.exports.loginValidation = loginValidation;
```

FIGURE 17. Validation with Joi.

In addition, The hashing of passwords is an essential step in securing the users on the back-end; nevertheless, it is not foolproof since it hashes consistently. Before feeding a password into a hashing mechanism, the process known as "salting" involves adding a string of random characters to the password. This indicates that it is predictable and vulnerable to assaults using dictionary entries or rainbow tables. Take a look at figure 18 below.

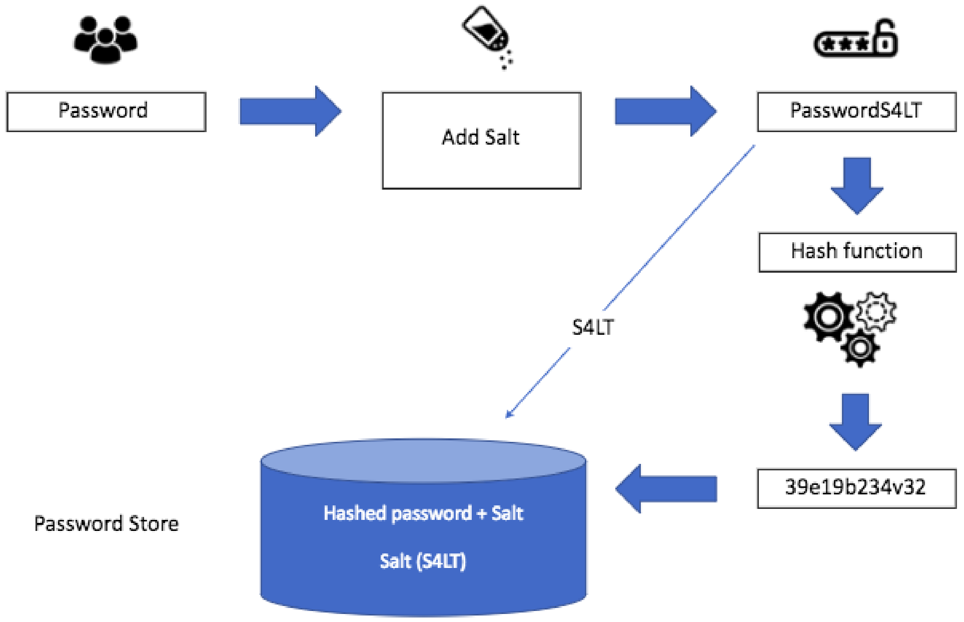
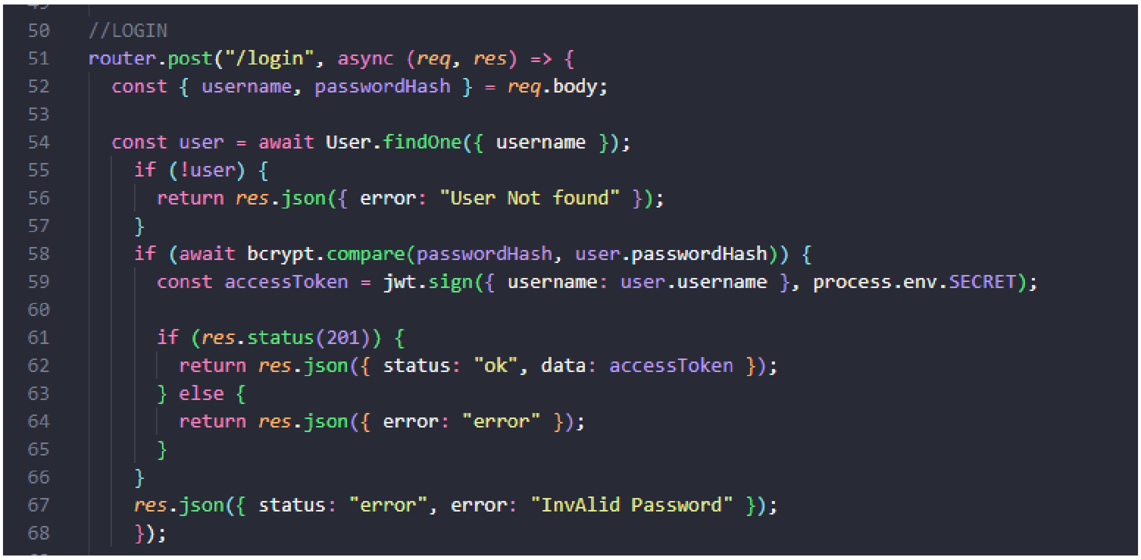


FIGURE 18. Where should salted passwords be stored? (20.)

The next step is configuring the login route, allowing the user to sign in using the account they have just established. Then a JSON web token was added to the auth, also known as JWT, to communicate and exchange sensitive information. Each JWT has its own encoded collection of JSON objects, which includes its claims. JWTs are digitally signed using a cryptographic technique to guarantee that the claims associated with the token cannot be modified after it has been issued. Take a look at figure 19 below.



```
//LOGIN
router.post("/login", async (req, res) => {
  const { username, passwordHash } = req.body;

  const user = await User.findOne({ username });
    if (!user) {
      return res.json({ error: "User Not found" });
    }
    if (await bcrypt.compare(passwordHash, user.passwordHash)) {
      const accessToken = jwt.sign({ username: user.username }, process.env.SECRET);

      if (res.status(201)) {
        return res.json({ status: "ok", data: accessToken });
      } else {
        return res.json({ error: "error" });
      }
    }
    res.json({ status: "error", error: "InvAlid Password" });
    });
```

FIGURE 19. Post method for login, create and assign a token.

User Route: The system makes it possible to see all of the registered users in the database by using this path, and it also makes it possible to view each user's information alongside their identification number. In addition, the program can change and remove people based on their IDs.

4.3 Development and Deployment

When developing software or websites, deployment refers to sending changes or updates from one deployment environment to another. When developers are configuring a website, it will continually have a live website, also known as the live environment or the production environment. Take a look at figure 20 below.

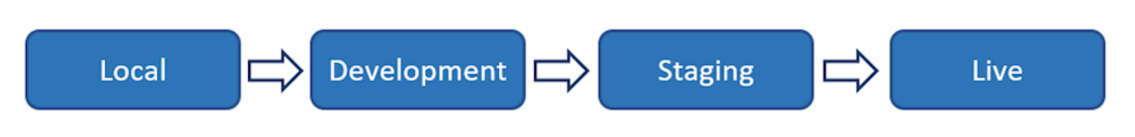
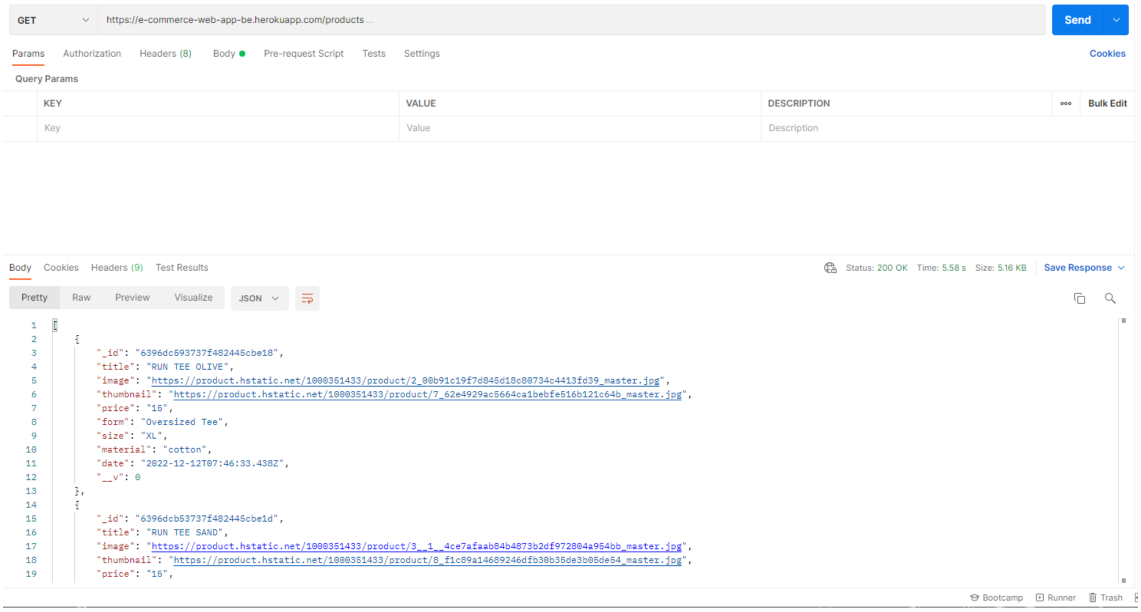


FIGURE 20. A Simplified and classic way of handling deployments when working with websites. (21.)

4.3.1 Development

The application is being created and tested using Postman constantly. Testing of the Postman API is now taking place concurrently with the development process and the bug-fixing phase. After an API route has been developed, the API request is tested using Postman against various use cases to confirm that the algorithm performs as intended. Testing with Postman is quite helpful since it gives the results very fast, allowing the developer to check to see whether the method has been appropriately written or to determine where the mistake lies and where it has to be repaired. The image below is an example of a test API, and it helps to retrieve all products in the database. Take a look at figure 21 below.



GET https://e-commerce-web-app-be.herokuapp.com/products ...

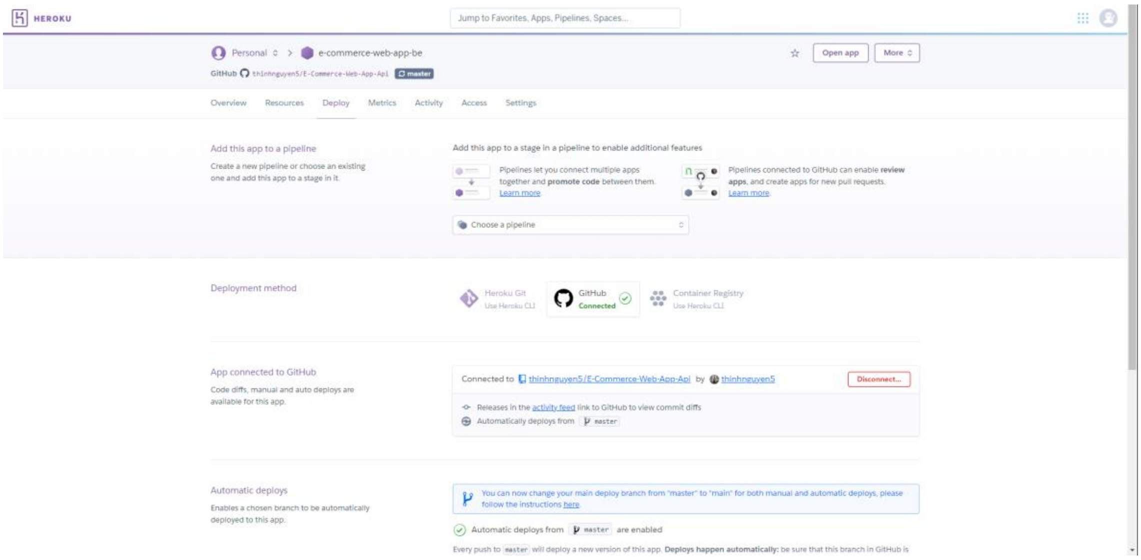
Status: 200 OK Time: 5.58 s Size: 5.16 KB

```
[
    {
        "_id": "6396dc593737f482445cbe18",
        "title": "RUN TEE OLIVE",
        "image": "https://product.hstatic.net/1000351433/product/2_00b91c19f7d845d18c80734c4413fd39_master.jpg",
        "thumbnail": "https://product.hstatic.net/1000351433/product/7_62e4929ac5664ca1bebfe516b121c64b_master.jpg",
        "price": "15",
        "form": "Oversized Tee",
        "size": "XL",
        "material": "cotton",
        "date": "2022-12-12T07:46:33.438Z",
        "__v": 0
    },
    {
        "_id": "6396dcb53737f482445cbe1d",
        "title": "RUN TEE SAND",
        "image": "https://product.hstatic.net/1000351433/product/3__1__4ce7afaab84b4873b2df972804a954bb_master.jpg",
        "thumbnail": "https://product.hstatic.net/1000351433/product/8_f1c89a14689246dfb30b35de3b05de54_master.jpg",
        "price": "15",
```

FIGURE 21. Testing API with Postman.

4.3.2 Deployment

When the API has been tested and functions correctly, it will be uploaded to Heroku. It is a cloud platform that allows developers to design, launch, manage, and grow their applications (PaaS - Platform as a service). It has a high degree of adaptability and is relatively simple to use, making it the most straightforward method for delivering the product to the end customer. It frees up the minds of developers to concentrate on product creation rather than the operation of servers or hardware. Take a look at figure 22 below.



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Deployment method

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to thinhnguyen5/E-Commerce-Web-App-Api by thinhnguyen5

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from `master`

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

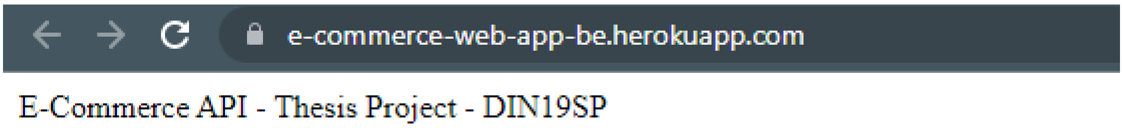
You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

Automatic deploys from `master` are enabled

Every push to `master` will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is

FIGURE 22. Deploy the GitHub repository to Heroku.

This application has a publicly accessible URL, which front-end apps may utilize to access the application. Take a look at figure 23 below.



E-Commerce API - Thesis Project - DIN19SP

FIGURE 23. Public URL.

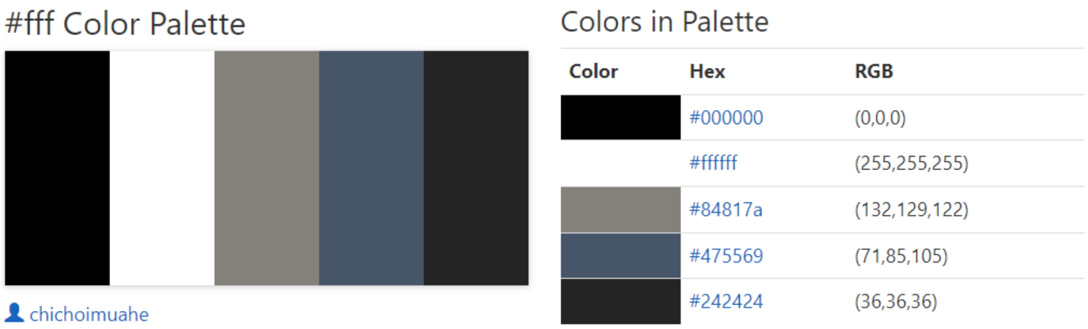
4.4 Web client functionalities

The study recommended in this part how the key features have been implemented to obtain the best possible user experience and performance level. When a technology must be improved or upgraded to a newer version, the solution may be extended to meet these needs.

4.4.1 UI

As this is a website devoted to selling t-shirts, there is a wide variety of colours, and the dark colours will showcase the items. Therefore, the colours and frames have been carefully picked from the beginning, with a preference for deep colours. This decision was made in favour of deep colours. The core colours are White #ffffff, Black #000000, Concord #84817A, Slate #475569, and Grey

#242424. Increasing the contrast and visual power of the composition may be accomplished by using a pair of black and white colours that complement one another. Take a look at figure 24 below.



#fff Color Palette

chichoimuahe

Colors in Palette

Color	Hex	RGB
■	#000000	(0,0,0)
□	#ffffff	(255,255,255)
■	#84817a	(132,129,122)
■	#475569	(71,85,105)
■	#242424	(36,36,36)

FIGURE 24. Main Color Palette

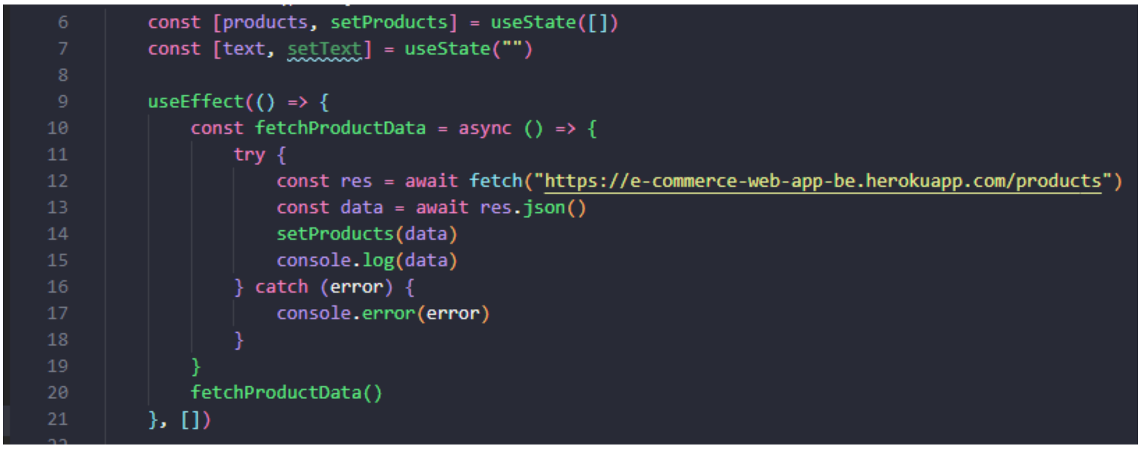
In addition, while creating the interfaces, it is necessary to ensure that they are aligned with the design system for the brand. To increase professionalism and to be aware that colour effects may influence the feelings experienced by the user, in addition to affecting how interactive elements within the application are interpreted.

4.4.2 Navbar

The Navbar includes elements that can lead to the home, store, story, news, login and cart pages. The navigation menu is an essential component of the website since it was designed to make the user's search more efficient. Internal links are included in the navigation bar, connecting to all the pages the website hopes its visitors will read. Above the navigation menu, it enables the owner to direct users to pages ranked lower in importance than the ones they are now viewing. Customers will only be able to connect to some pages of the website's pages from the home page in the absence of a navigation menu. This will limit their ability to get all of the information they need. The navigation bar also helps guarantee that people explore the menu and remain on the site for extended periods, ultimately resulting in the retention of clients for a longer time. In addition to this, navigation contributes to the enhancement of the pageviews index.

4.4.3 Store

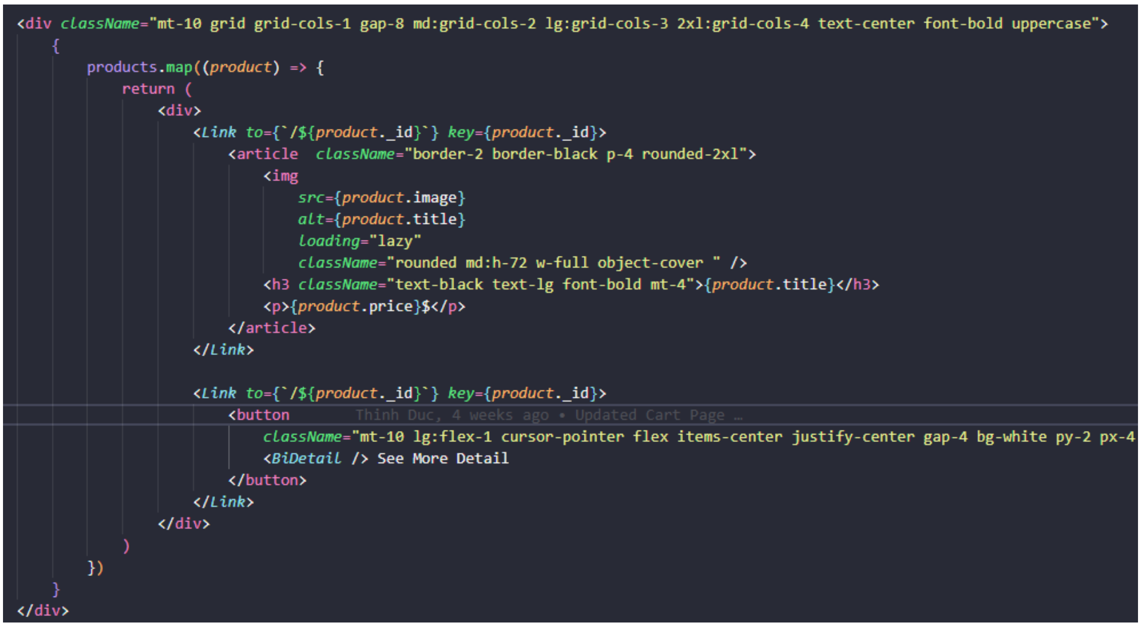
The area of an e-commerce website that is dedicated to displaying products for visitors to peruse is referred to as the "product display section," and it is an unquestionably essential component. On this page, the consumer will be able to see an image of the product and the product's name and price. If a consumer finds a product that interests them, they may access further information about it by clicking on the picture of the button that reads "See More Detail" underneath the product. These items are found in the database and fetch the data in React with Hooks using state and effect hooks. Take a look at figure 25 below.



```
const [products, setProducts] = useState([])
const [text, setText] = useState("")

useEffect(() => {
    const fetchProductData = async () => {
        try {
            const res = await fetch("https://e-commerce-web-app-be.herokuapp.com/products")
            const data = await res.json()
            setProducts(data)
            console.log(data)
        } catch (error) {
            console.error(error)
        }
    }
    fetchProductData()
}, [])
```

FIGURE 25. Fetch Data

After retrieving the data from the database, the map method will be used to display the desired information to the client. The map function in React is used to traverse and show a list of comparable component objects. React does not have a map. Instead, it is the regular JavaScript array function. map() generates a new array by running a specified function on each member of the calling array. Take a look at figure 26 below.

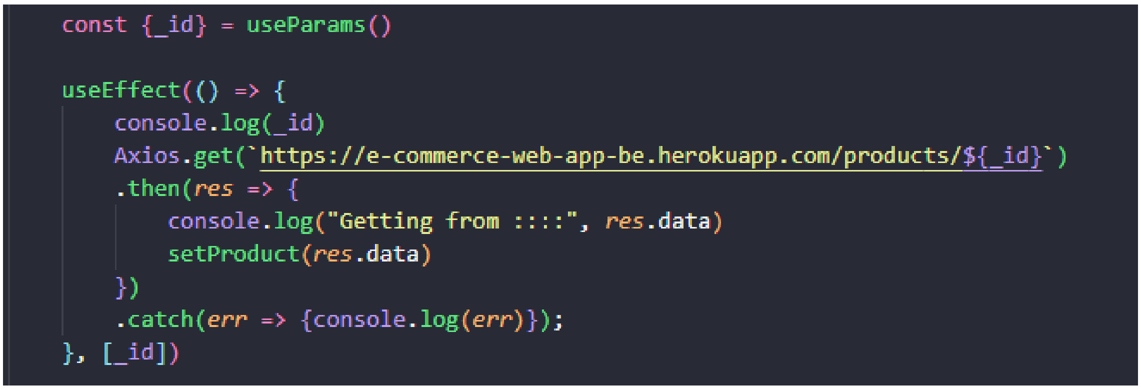


```
<div className="mt-10 grid grid-cols-1 gap-8 md:grid-cols-2 lg:grid-cols-3 2xl:grid-cols-4 text-center font-bold uppercase">
    {
        products.map((product) => {
            return (
                <div>
                    <Link to={`/${product._id}`} key={product._id}>
                        <article  className="border-2 border-black p-4 rounded-2xl">
                            <img
                                src={product.image}
                                alt={product.title}
                                loading="lazy"
                                className="rounded md:h-72 w-full object-cover " />
                            <h3 className="text-black text-lg font-bold mt-4">{product.title}</h3>
                            <p>{product.price}$</p>
                        </article>
                    </Link>

                    <Link to={`/${product._id}`} key={product._id}>
                        <button        Thinh Duc, 4 weeks ago • Updated Cart Page …
                            className="mt-10 lg:flex-1 cursor-pointer flex items-center justify-center gap-4 bg-white py-2 px-4
                            <BiDetail /> See More Detail
                        </button>
                    </Link>
                </div>
            )
        })
    }
</div>
```

FIGURE 26. Call out data

4.4.4 Single Product

It is necessary to make use of the useParams function in order to be able to access a page with product information. Since version 5, the useParams hook has been a standard component of React router to get route parameters from the functional component rendered by the matching route. It is helpful to have this capability. The useParams hook of the React Router is responsible for returning an object. The keys of this object are the parameter names stated in the path string of the Route definition, and the values of this object are the corresponding URL segment extracted from the matching URL. Take a look at figure 27 below.



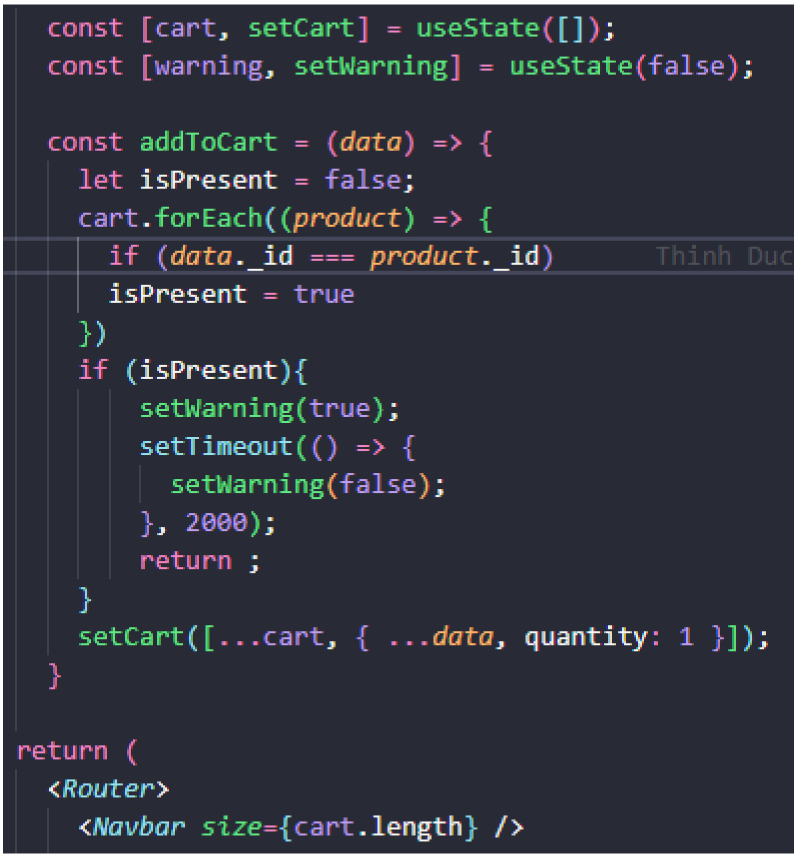
```
const {_id} = useParams()

useEffect(() => {
    console.log(_id)
    Axios.get(`https://e-commerce-web-app-be.herokuapp.com/products/${_id}`)
    .then(res => {
        console.log("Getting from ::::", res.data)
        setProduct(res.data)
    })
    .catch(err => {console.log(err)});
}, [_id])
```

FIGURE 27. Show the product per Id

Customers who have discovered a product that interests them and want to learn more about it will arrive on this page after clicking on the product's name to get further details about it. They will be able to see all the information about that product in this section. Then, if the customer wants to purchase the product, the customer may put it in the cart, and if the product is already in the cart, the client will not be able to add it to the cart again, and the system will show a notice that says the product has been in the shopping cart before. Customers may check the contents of their cart by clicking the cart symbol above the navigation bar, or they can go back to the shop page to continue looking for additional items.

4.4.5 Shopping Cart

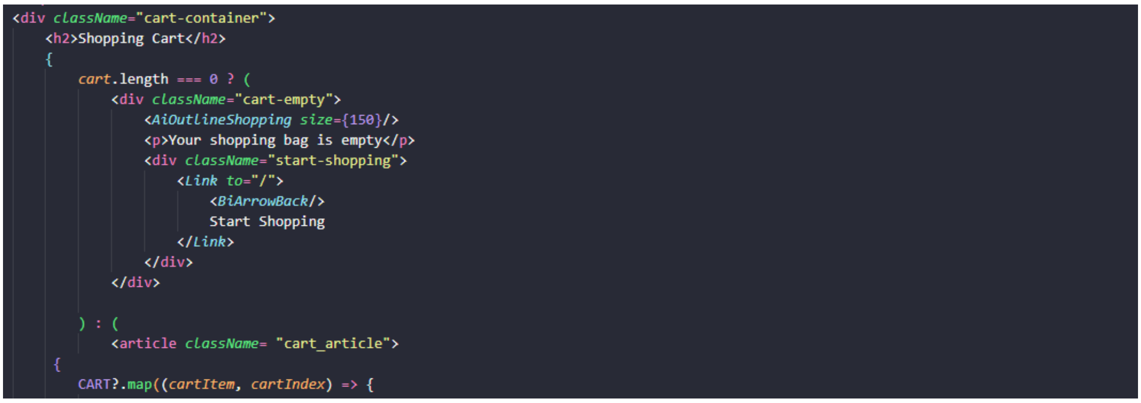
Customers may see what they have in their inventory by clicking the shopping cart symbol above the navigation bar. They can also determine whether or not their cart contains any items by glancing at the little number next to the cart's icon; if there are no items in the cart, the number will display 0. In addition, it will show the number following the total quantity of items placed in the shopping cart. Take a look at figure 28 below.



```
  const [cart, setCart] = useState([]);
  const [warning, setWarning] = useState(false);

  const addToCart = (data) => {
    let isPresent = false;
    cart.forEach((product) => {
      if (data._id === product._id)        Thinh Duc
      isPresent = true
    })
    if (isPresent){
      setWarning(true);
      setTimeout(() => {
        setWarning(false);
      }, 2000);
      return ;
    }
    setCart([...cart, { ...data, quantity: 1 }]);
  }

return (
  <Router>
    <Navbar size={cart.length} />
```

FIGURE 28. Add to cart function

If the consumer does not have any items in their shopping cart, they will get a notification that their cart is empty, and they will be able to return to the page that lists all available goods to continue shopping. On the other hand, the items a client has chosen to purchase are shown in the shopping cart for them to review. Take a look at figure 29 below.



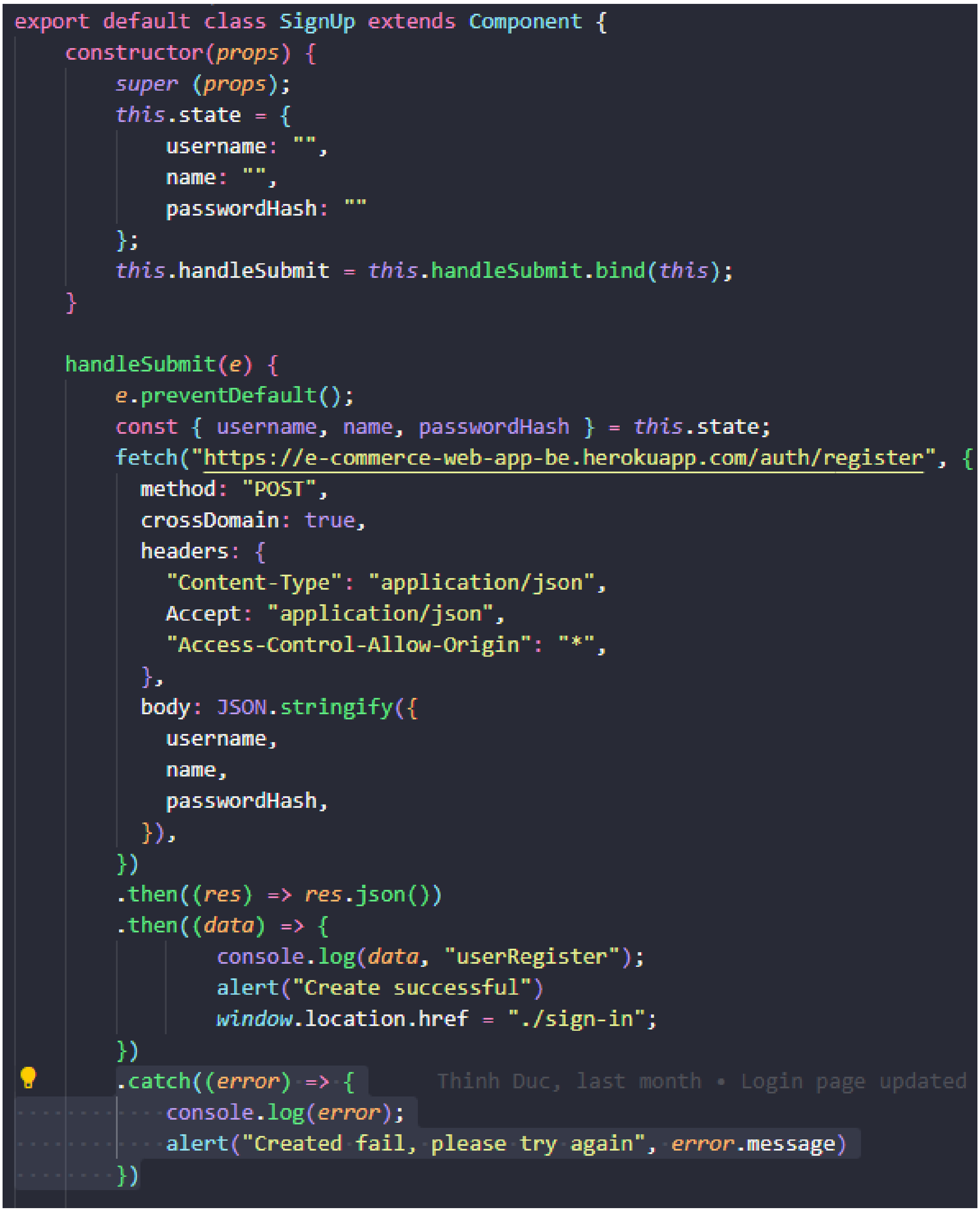
```
<div className="cart-container">
    <h2>Shopping Cart</h2>
    {
        cart.length === 0 ? (
            <div className="cart-empty">
                <AiOutlineShopping size={150}/>
                <p>Your shopping bag is empty</p>
                <div className="start-shopping">
                    <Link to="/">
                        <BiArrowBack/>
                        Start Shopping
                    </Link>
                </div>
            </div>

        ) : (
            <article className= "cart_article">
     {
        CART?.map((cartItem, cartIndex) => {
```

FIGURE 29. Two customer views of the shopping cart

If they choose to modify the amount of an item, they may do so at any time, or customers can delete it entirely if they decide it is no longer appropriate for their needs. Whenever the customer makes modifications, an automated adjustment will be made to the total amount the client is responsible for paying.

4.4.6 Authentication

Visitors need to input three pieces of information to create a new account: their username, name, and password. The user is given unrestricted latitude in the name area to make selections that suit their tastes. If another individual has already chosen the username, the system will alert the user that their registration was unsuccessful. The system will compare the username the client selects with all the other possible usernames stored in the database. For comparison, the fetch method may retrieve existing users from the database. Take a look at figure 30 below.



```
export default class SignUp extends Component {
    constructor(props) {
        super (props);
        this.state = {
            username: "",
            name: "",
            passwordHash: ""
        };
        this.handleSubmit = this.handleSubmit.bind(this);
    }

    handleSubmit(e) {
        e.preventDefault();
        const { username, name, passwordHash } = this.state;
        fetch("https://e-commerce-web-app-be.herokuapp.com/auth/register", {
          method: "POST",
          crossDomain: true,
          headers: {
            "Content-Type": "application/json",
            Accept: "application/json",
            "Access-Control-Allow-Origin": "*",
          },
          body: JSON.stringify({
            username,
            name,
            passwordHash,
          }),
        })
        .then((res) => res.json())
        .then((data) => {
                console.log(data, "userRegister");
                alert("Create successful")
                window.location.href = "./sign-in";
        })
        .catch((error) => {      Thinh Duc, last month • Login page updated
            console.log(error);
            alert("Created fail, please try again", error.message)
        })
```

FIGURE 30. Fetch register

For the registration to be successful, the password section must be at least six characters long. After completing the necessary processes, the client will get confirmation from the system that their registration was successful and be sent immediately to the login page. Customers have to make sure that the username and password they use to log in are correct for them to be able to do so. If any or both of these details are incorrect, the customer will not be able to log in. After a successful login, the client will be sent to their page. The customer's username and the name they registered under on this page will be shown. After they have finished their tasks, they are free to log out of their accounts and close them.

4.5 Deployment

After everything has been double-checked and finished, the website's hosting will be handled by Netlify. It is no longer restricted to being seen on localhost; anybody who accesses the public URL will also see it.

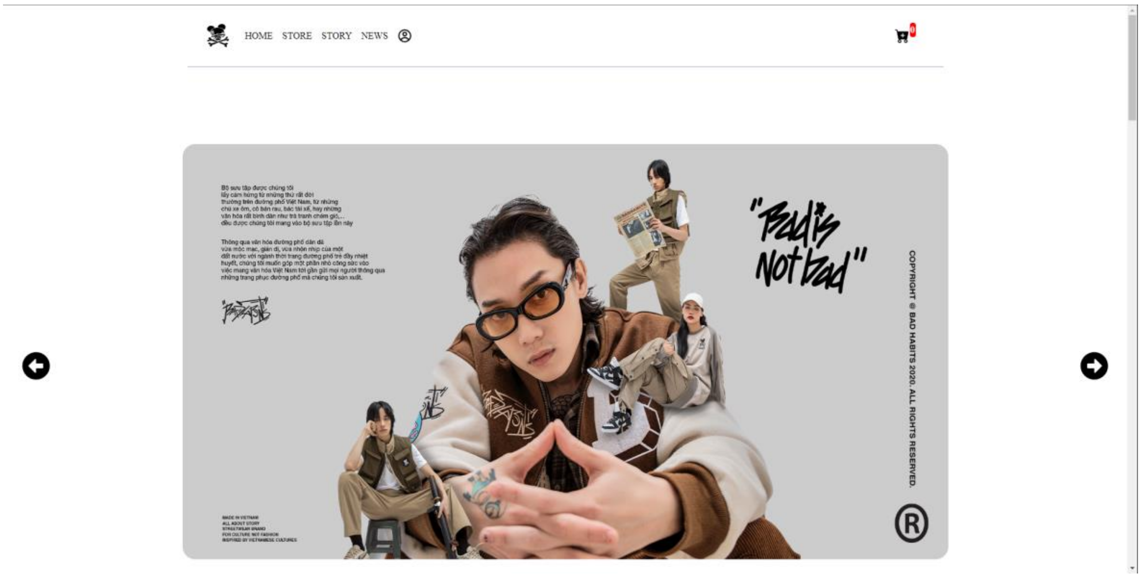
The purpose of the incredible web development platform known as Netlify is to increase user productivity in the most effective manner possible, making it one of the most remarkable tools available today. The platform assists developers in constructing, testing, and deploying websites. As a result of merging the current, decoupled web parts with local development processes and more complex logic, Netlify provides an astoundingly quicker method to guarantee much more efficient, scalable, and secure websites and apps.

The website business is rapidly moving away from monolithic methods **~~and~~** toward decoupled ones, and this shift is constantly taking place. While this is happening, developers are making tremendous strides forward with far more power than ever. On the other hand, Netlify was explicitly designed with this time in mind. It provides unique web automation technologies in addition to web hosting infrastructure. The most astounding part is that Netlify can give solutions to the next generation while being within a highly reasonable price range.

Netlify will adequately connect to the developer's GitHub repository to get the user's website's source code. It will then execute a build process to pre-render the customer website's pages into static HTML. The generated pages will be disseminated and deployed throughout various content delivery networks. On the other hand, if a user requests to visit a user website, it will automatically choose a data centre located as close as possible to serve users better. (22)

5 CUSTOMER VIEW

The application operated at its maximum capacity. The UI version was eventually corrected by correcting the issues detected during testing. This was accomplished by obtaining comments and choosing the proper plan to publish to the application. The evolution of this application's usefulness will constantly depend on user feedback. In conclusion, the development of the application and the identification of improvements are of value to the application. The product images included in the T-shirt e-commerce store were sourced from a local Vietnamese brand called Bad Habit. (23) Take a look at figure 31 below for Navbar



Bộ sưu tập được chúng tôi
lấy cảm hứng từ những thứ rất đời
thường trên đường phố Việt Nam, từ những
chú xe ôm, cô bán rau, bác tài xế, hay những
văn hóa rất bình dân như trà tranh chém gió,...
đều được chúng tôi mang vào bộ sưu tập lần này

Thông qua văn hóa đường phố dân dã
vừa mộc mạc, giản dị, vừa nhộn nhịp của một
đất nước với ngành thời trang đường phố trẻ đầy nhiệt
huyết, chúng tôi muốn góp một phần nhỏ công sức vào
việc mang văn hóa Việt Nam tới gần gũi mọi người thông qua
những trang phục đường phố mà chúng tôi sản xuất.

FIGURE 31. Navbar

Figure 31 plays the website's Navbar with links to other pages such as the product and news pages. Besides, there is also a display of the customer's cart; the quantity in the cart will change based on the number of products added.

Navbar was modified to be viewable on smaller devices. Take a look at figure 32 below.

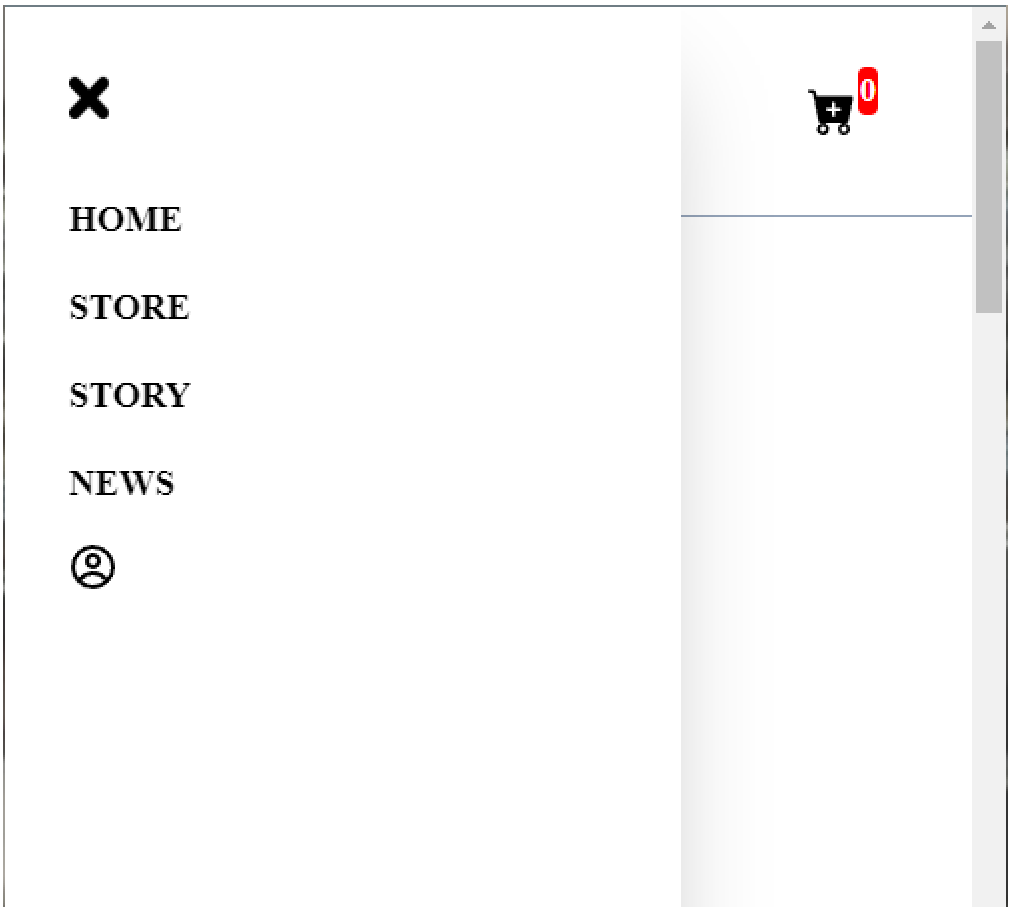


FIGURE 32. *Navbar Responsive*

Figure 32 performs when customers use smaller devices, and the website interface will be adjusted to suit those solutions.

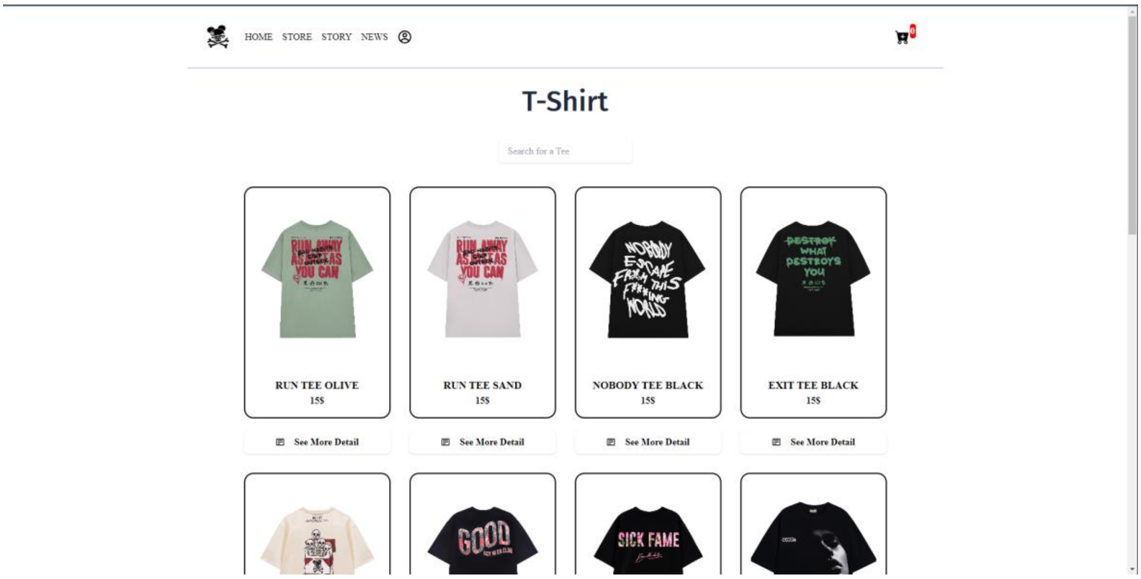
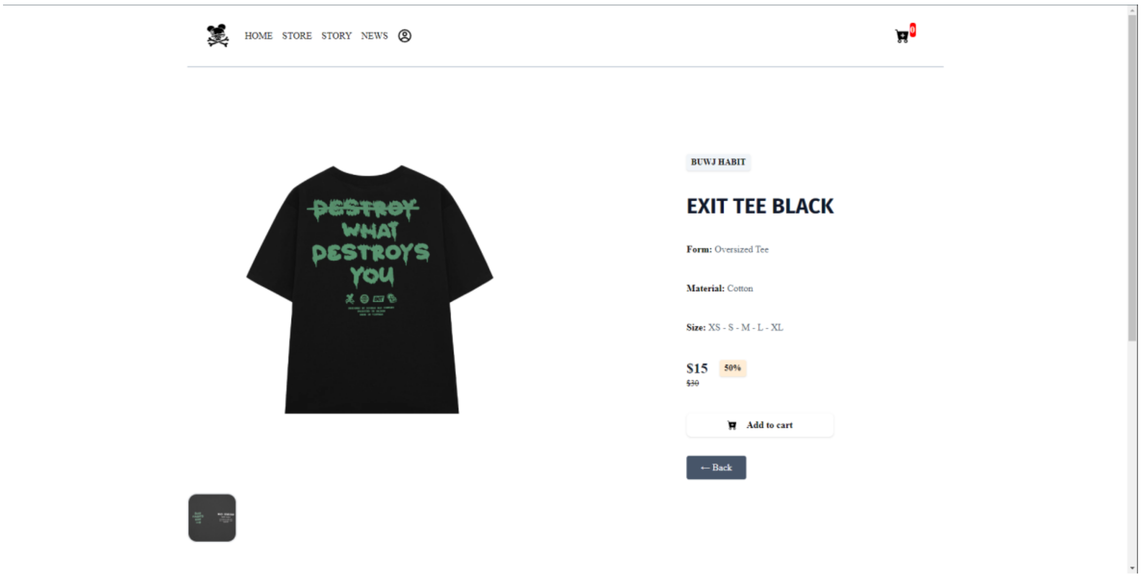


FIGURE 33. Products page

Figure 33 shows the product page, which displays all the T-shirts for which the store has the same name and price. 'See More Detail' button to go to the detailed information page of each product.



EXIT TEE BLACK

Form: Oversized Tee

Material: Cotton

Size: XS - S - M - L - XL

$15 50%
~~$30~~

FIGURE 34. Single Product page

Figure 34 displays detailed product information. In addition, customers can add products to their shopping cart or return to the previous page to continue shopping.

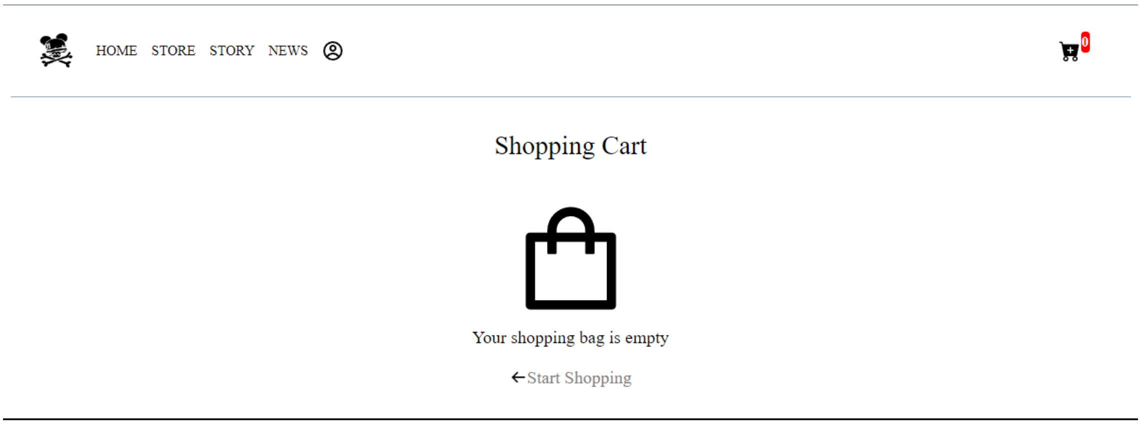
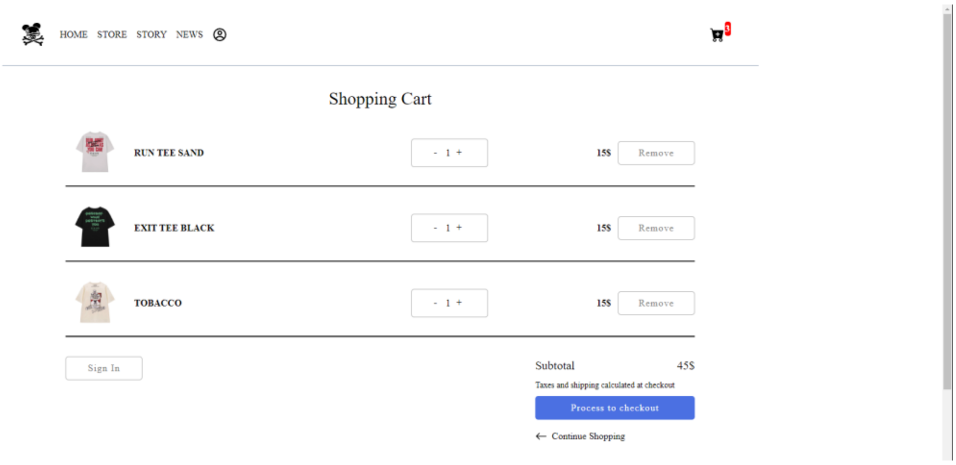


FIGURE 35. Empty Shopping Cart

Figure 35 illustrates the cart page of the customer, ~~but~~ when no products have been loaded. Customers can shop by clicking on the word "Start Shopping."



RUN TEE SAND	- 1 +	15$	Remove
EXIT TEE BLACK	- 1 +	15$	Remove
TOBACCO	- 1 +	15$	Remove

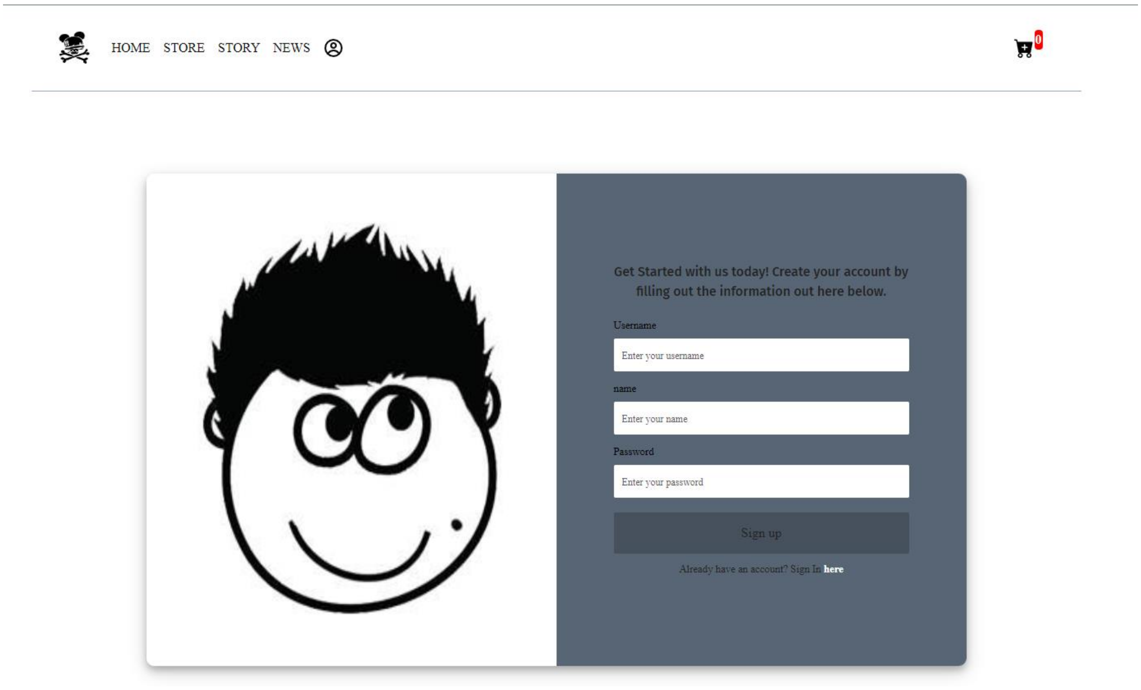
Subtotal 45$

Taxes and shipping calculated at checkout

FIGURE 36. Shopping Cart

Figure 36 demonstrates that the shopping cart when adding products, will display the product's photo, name, price and quantity. The customer can increase the quantity of the T-shirt or remove it from the cart. After that, customers can log in to pay.

Customers need to log in to pay. They can register a new account if they do not have an account. Take a look at figure 37 below.



Get Started with us today! Create your account by filling out the information out here below.

Already have an account? Sign In **here**

FIGURE 37. *Register page*

Figure 37 illustrates that customers need to log in to pay. They can register a new account if they do not have an account.

6 CONCLUSION

In conclusion, the following opinions will be provided: Firstly, an overall evaluation of the performance of the technical languages used to create the T-Shirts web application. In addition, the project results compared to the requirements are set in the original plan. Last but not least, this thesis has given me the abilities I need to advance in the future.

Regarding the performance of the selected technology, concerning the front-end, the speed, performance and usability of ReactJS look pretty excellent. Moreover, it is a widely used and quickly learned language. Besides, the performance of Tailwind CSS has helped the UI grow faster and better than other CSS writing methods. In addition, Express was developed to make creating APIs and web applications simple, and web efficiency is maintained, but a significant amount of coding time is reduced. Another thing is that Express is fast to link with databases like MongoDB, which was chosen for this thesis. With MongoDB's assistance, storing and using data in the database for data processing is more straightforward.

Every requirement was completed at the project's end, and everything was checked extensively to ensure it functions flawlessly. The outcome turned out to be considerably better than anticipated in the end. The application is constantly improving and changing throughout the design and implementation processes. The user may discover things they are interested in buying more quickly and have a better experience because of the interface's simplicity, which is seen as an advantage.

The established project is an example of giving birth to future works that incorporate task assignments in a realistic way to tackle daily difficulties. This example will give rise to future works. Writing my thesis for graduation has been an educational and growth-promoting experience. All of this contributes to improving the structure of the accumulated information, expanding that knowledge, and enhancing the capability for analysis and problem-solving. In the end, the thesis process has established the groundwork for future work and opens the door to the possibility of independent study.

REFERENCES

1. A.N. 29.11.2021. Thương mại điện tử trở thành xu hướng tất yếu. Date of retrieval: 11.10.2022. Available: https://dangcongsan.vn/kinh-te-va-hoi-nhap/thuong-mai-dien-tu-tro-thanh-xu-huong-tat-yeu-598414.html
2. Shopify. Ecommerce. Date of retrieval: 11.10.2022. Available: https://www.shopify.com/encyclopedia/what-is-ecommerce
3. Danielle Zanzalari. 13.9.2022. Advantages of E-Commerce. Date of retrieval: 11.10.2022. Available: https://www.thebalancemoney.com/advantages-of-ecommerce-1141610#toc-advantages-of-e-commerce-for-customers
4. GIANG. 19.06.2018. Web App là gì? Có gì khác với Website .Date of retrieval: 11.02.2022. Available: https://bizflycloud.vn/tin-tuc/web-application-la-gi-co-gi-khac-voi-website-20180619105652369.htm
5. Software Engineer Training. 24.05.2022. Lập trình backend là gì? Tự học lập trình REST API với Golang. Date of retrieval: 11.02.2022. Available: https://viblo.asia/p/lap-trinh-backend-la-gi-tu-hoc-lap-trinh-rest-api-voi-golang-L4x5xAzgKBM
6. Giangpth. 18.05.2018. Web Server là gì? Tìm hiểu cơ chế vận hành của web server. Date of retrieval: 11.02.2022. Available: https://bizflycloud.vn/tin-tuc/tat-tat-kien-thuc-co-ban-ve-web-server-ban-phai-biet-20180515115521302.htm
7. Kyr Doo-Hyun. Hệ quản trị cơ sở dữ liệu là gì? DBMS là gì? Date of retrieval: 11.02.2022. Available: https://quantrimang.com/cong-nghe/dbms-la-gi-172534
8. Audrey. 24.05.2017. TÌM HIỂU VỀ MONGODB. Date of retrieval: 11.02.2022. Available: https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGlY3
9. OpenJS foundation. 23.9.2022. Node.js. Date of retrieval: 6.12.2022. Available: https://en.wikipedia.org/wiki/Node.js

10. Kevin Kononenko. 3.11.2017. Going out to eat and understanding the basics of Express.js. Date of retrieval: 6.12.2022. Available: https://www.freecodecamp.org/news/going-out-to-eat-and-understanding-the-basics-of-express-js-f034a029fb66
11. David Taylor. 12.11.2022. What is MongoDB? Introduction, Architecture, Features & Example. Date of retrieval: 6.12.2022. Available: https://www.guru99.com/what-is-mongodb.html
12. Code institute. What is React.js. Date of retrieval: 6.12.2022. Available: https://codeinstitute.net/global/blog/what-is-react-js/
13. Karina Guerra. 5.1.2021. The Lifecycle of a React Component. Date of retrieval: 6.12.2022. Available: https://medium.com/codex/the-lifecycle-of-a-react-component-8e01332a068d
14. Anna Fitzgerald. 31.05.2022. Tailwind CSS: What It Is, Why Use It & Examples. Date of retrieval: 10.12.2022. Available: https://blog.hubspot.com/website/what-is-tailwind-css
15. INFYNNO SOLUTIONS. Why Tailwind CSS is Trending Now? Date of retrieval: 11.12.2022. Available: https://infynno.com/article/why-tailwind-css-is-trending-now/
16. Sayeda Haifa Perveez. 09.12.2022. What is Git: Features, Command and Workflow in Git. Date of retrieval: 11.12.2022. Available: https://www.simplilearn.com/tutorials/git-tutorial/what-is-git
17. KINSTA. 06.12.2022. What Is GitHub? A Beginner's Introduction to GitHub. Date of retrieval: 11.12.2022. Available:

    https://kinsta.com/knowledgebase/what-is-github/

18. Martin Heller. 08.07.2022. What is Visual Studio Code? Microsoft's extensible code editor. Date of retrieval: 12.12.2022. Available: https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html
19. TopDev. Postman là gì? API Testing với Postman. Date of retrieval: 12.12.2022. Available: https://topdev.vn/blog/postman-la-gi/

20. Jason Jung. 07.05.2021. What is Password Hashing and Salting? Date of retrieval: 19.12.2022. Available: https://www.okta.com/uk/blog/2019/03/what-are-salted-passwords-and-password-hashing/#:~:text=Password%20hashing%20is%20defined%20as,series%20of%20numbers%20and%20letters.
21. umbraco. What is Deployment? Date of retrieval: 26.12.2022. Available: https://umbraco.com/knowledge-base/deployment/
22. back4app. What is Netlify? Date of retrieval: 29.12.2022. Available: https://blog.back4app.com/what-is-netlify/
23. BAD HABITS OFFICIAL ONLINE STORE. 2023. Date of retrieval: 10.02.2023. Available: https://badhabitsstore.vn/