Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Diploma Thesis

Study web application development using MERN Stack

Patel Gaurangkumar

© 2022 ČZU v Praze

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Gaurangkumar Patel

Systems Engineering and Informatics
Informatics

Thesis title

Study web application development using MERN Stack

Objectives of thesis

The thesis will focus on the study and exploration of MERN, a fully JavaScript stack, for developing web applications. It will go through the process of designing the GUI (React.Js), database design in MongoDB. It will implement the backend using Node.js and route the navigation using Express.Js. The main goal is to develop the example application using only & only JavaScript frameworks.

Methodology

To study and explore the MERN stack this thesis will focus on going through the various technologies involved. An e-shop will be created at the end as the result of the application of studying these technologies. After designing a mind map, Use cases, activities and other diagrams, the first step will include design and implementation in the context of MongoDB – a NoSQL document-based database. The next step will be to visualize the managed access to these data documents by using Postman. The backend will use Node.Js for the implementation. Once the data part is implemented, the need for the Front end will be satisfied by developing it using React.js. Express.JS will then be used for routing and creating connections in the application following payment gateway integration.

The proposed extent of the thesis

60-80 pages

Keywords

Mongo.DB, React.Js, Express.JS, Node.js, E-shop, Web development, MERN Stack Study.

Recommended information sources

Vasan Subramanian, Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node

---

Expected date of thesis defence

2022/23 WS – FEM

The Diploma Thesis Supervisor

Ing. Jiří Brožek, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 4. 11. 2022

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 28. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 29. 11. 2022

Declaration

I declare that I have worked on my diploma thesis titled " Study web application development using MERN Stack" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break any copyrights.

In Prague on date of submission 30.11.2022 _____

Acknowledgement

I would like to thank Ing. Ing. Jiří Brožek, Ph.D. for his support and guidance while working on this thesis. Additionally, I would like to thank my family and close friends for their encouragement, support and belief in me.

Study web application development using MERN Stack

Abstract

In contemporary technology, the majority are the use of generation for leading their lives and gratifying their day by day desires. in this generation maximum of us the use of E-trade websites for purchasing for clothes, groceries, and electronics. we've advanced one E-trade net application by way of the use of MERNstack era because it includes MongoDB, specific.JS framework, React.JS library, Node.JS platform.

This utility is fully useful with extraordinary perspectives for consumer and admin and it also has incorporated with fee gateway for checkout. by means of the usage of this internet site we should buy exclusive sorts of t-shirts and we can pick one of a kind varieties of t-shirts primarily based upon customer hobbies. in this assignment, we are able to add unique merchandise and might delete them additionally. we've got developed administrative functions for the website which include create a product, create classes, Admin dashboard, control products, control categories. For clients, they can quick upload their items to the cart. based totally on the objects inside the cart then the bill receives generate and the customer will pay through using stripe.

Keywords:

Mongo.DB, React.Js, Express.JS, Node.js, E-shop, Web development, MERN Stack Study.

Study web application development using MERN Stack

Abstraktní

V současné technologii většina z nich využívá generace k vedení svých životů a uspokojování svých každodenních tužeb. v této generaci maximum z nás využívá webové stránky E-Trade k nákupu oblečení, potravin a elektroniky. jednu aplikaci E-trade net jsme pokročili pomocí éry MERNstack, protože zahrnuje MongoDB, specific.JS framework, knihovnu React.JS, platformu Node.JS.

Tato pomůcka je plně užitečná s mimořádnými perspektivami pro spotřebitele a správce a je také začleněna do brány pro platby za platbu. Pomocí této internetové stránky bychom měli nakupovat exkluzivní druhy triček a můžeme si vybrat jedny z druhů triček především na základě zákaznických zálib. v tomto zadání jsme schopni přidat jedinečné zboží a můžeme je dodatečně odstranit. pro web jsme vyvinuli administrativní funkce, které zahrnují vytvoření produktu, vytvoření tříd, administrační panel, ovládání produktů, ovládání kategorií. Klienti mohou rychle vložit své položky do košíku. zcela na základě předmětů uvnitř košíku pak účet obdrží vygenerovat a zákazník zaplatí pomocí pruhu.

Klíčová slova:

Mongodb, React.Js, Express.JS, Node.js, Eshop, Web development, MERN Stack Study.

Table of Contents

1. Introduction 12
1.1. MERN Stack 12
2. Objective and Methodology 14
2.1. Aims and objectives 14
2.2. Aims and objectives 14
3. Literature review 15
3.1. E-commerce 15
3.1.1. E-commerce definition 15
3.1.2. Types of E-commerce 15
3.1.3. Advantages of E-commerce 17
3.2. Impact of e-Business on Traditional Markets 18
3.3. Success and Failure factors of E-Business 19
3.3.1. Success factors in E-Business 19
3.3.2. Failure factors in E-Business 20
3.4. Single Page website vs Multiple Page website 20
3.5. MERN Stack in Website Development 22
3.6. Principle of Object-Oriented Programming (OOP) 23
3.6.1. Encapsulation 23
3.6.2. Abstraction 23
3.6.3. Inheritance 23
3.6.4. Polymorphism 23
3.7. OOP in JavaScript 24
3.8. MERN Stack 24
3.8.1. JavaScript 25
3.8.2. Node.JS 26
3.8.3. Express.JS 27
3.8.4. MongoDB 28
3.8.5. ReactJS 29
3.9. Benefits of using MERN Stack 30
3.9.1. Open-Source Technology 30
3.9.2. Free Templates 31
3.9.3. Help to Build Fast 31
3.9.4. Easy to Use 31

3.9.5. Full-Stack Development through MERN Technology 31
3.9.6. Community Support 32
3.9.7. Worldwide Exposure 32
3.10. Comparison between MERN stack and Full Stack 32
3.11. Comparison between MERN stack and MEAN Stack 33
4. Methodology 35
4.1. MERN Stack 35
4.2. FRONT END 36
4.2.1. React JS 36
4.2.2. Performance of React.js 36
4.2.3. JSX 37
4.2.4. Virtual DOM 37
4.3. Node.JS 38
4.4. Generation of Database 39
4.4.1. NoSQL Database 39
4.5. Software development life cycle 40
4.6. Software development life cycle stages 41
4.6.1. NoSQL Database 41
4.6.2. Defining Requirement 41
4.6.3. Designing the Software Architecture 41
4.6.4. Building or Developing the Product 42
4.6.5. Testing 42
4.7. Models of SDLC 42
4.7.1. The Waterfall model 42
4.7.1.1. Requirement Analysis 43
4.7.1.2. System Design 43
4.7.1.3. Implementation 43
4.7.1.4. Testing 43
4.7.1.5. Deployment 44
4.7.1.6. Maintenance 44
4.7.2. The Iterative Waterfall model 44
4.7.3. The V model 45
4.7.4. The Spiral model 45
4.8. Waterfall Model Application 46

4.8.1. When to select waterfall model 46
4.9. Reason to select Waterfall model 46
5. Results 48
5.1. Front-end 48
5.1.1. Homepage 48
5.1.2. Sign in and Sign up 49
5.1.3. Cart 51
5.1.4. Dashboard for Admin 52
5.2. Chapter Summary 54
6. Discussion & Analysis 56
6.1. Models 56
6.2. Routers 56
6.3. Controllers 57
6.4. DATABASE 57
6.4.1. . Creation of Databases 58
6.5. Back-end Initial Configuration 58
6.6. Designing a Mongoose Schema for back-end 60
6.6.1. . User Schema 60
6.6.2. Product Schema 62
6.7. API and Routing Documentation 63
6.7.1. Routing and Middleware 63
6.8. API and Routing Documentation 64
6.9. Postman & its usage to request post routes 64
6.10. Front-end development 65
6.10.1. Setup and Routing 65
6.10.2. User Authentication 66
6.10.3. Dashboard Component 69
6.10.4. Cart Component 70
6.11. Payment integration 71
6.11.1. PayPal integration 71
6.11.2. Stripe integration 72
6.12. Discussion 72
7. Conclusion 76
8. Bibliography 77

Table of Figure

Figure 1MERN Architecture 36
Figure 2Waterfall Model 43
Figure 3Home page 49
Figure 4Signin Page 50
Figure 5Signup Page 51
Figure 6Cart Page 52
Figure 7Admin Panel 53
Figure 8Admin Dashboard All products list 54
Figure 9Categories Page 54
Figure 10App.js Middleware 59
Figure 11DB connectivity with mongoose 60
Figure 12Schema example 61
Figure 13Password schema with storage hash 62
Figure 14Product schema 63
Figure 15Routung requests or pages 66
Figure 16Sign up Page 67
Figure 17User Authentication hash 68
Figure 18Admin Area 69

1. Introduction

The website development is considered to be extremely significant as it helps to increase the communication with the visitors in an effective manner. Interaction with the audience is becoming a vital part for the businesses to earn the profit and generate more business. Some of the traditional technologies which were used for the website development includes the JAVA servlets, or PHP. These technologies are quite widespread and provide relatively good features which are used in development. However, these technologies provide some the limitations. The main objective of the study is to explain all the key concepts which are utilized in the MERN stack. It also discusses their benefits and compatibilities which helps to explain the importance of MERN Stack in web development. Web application development today is done with full reliance on these traditional technologies. The major advantages of such an approach are the ease of understanding and extensibility. However, such trade-offs come with some limitations. The key objective of this study is to explain all the key concepts and their benefits which helps to explain the importance of MERN Stack in web development. In web development, the MERN stack was introduced to prove that it is possible to build vastly rich and responsive applications on a single set of tools. It solves the problem of different languages and frameworks; it provides a unified platform with different applications running together seamlessly in one integrated environment.

1.1.MERN Stack

The MERN stack which provides the simplicity, and uniformity. This is also developed to give the better solution in order to solve all the problems occurring in the previous technologies. The MERN stack is basically considered to be the JavaScript based stack which is formulated to provide the procedure of development. It contains four elements which are open-source. Hence the MERN stack is quite significant to use for the website development as it provides various aspects to design the website. The technologies which are used in the MERN stack helps to provide an end-to-end web stack to the developers for the purpose of using in the website development and is also a powerful stack which can help in developing of an e-commerce website (1). In today's technological era, companies need to develop their web application using the most robust technologies. MERN stack is one such technology that

provides many benefits. The main objective of this study was to discuss all the key concepts used in the MERN stack and their importance in web development. It also discusses their benefits, compatibilities, and similarities which helps to explain the importance of MERN Stack in web development. Web Application Development using the MERN stack helped in solving various problems. This study also developed a theme of shifting to a new technology that would help to solve all the problems that arise when using the previous technologies. The four technologies which are utilized in the MERN stack are JavaScript-based. MERN is a powerful stack that can be used for the development of an e-commerce website by providing various components like database management, microservices, etc (2).

2. Objective and Methodology

2.1. Aims and objectives

The main purpose of the current study is to create a web application which will helpful in demonstrating the knowledge about the MERN Stack technology. The website has always been a core identity to any business who has made them available to their users or customers to keep them updated and stay in touch with them. With inspired from this identity era and a background in web application development, this thesis will focus on the study and exploration of MERN a fully JavaScript stack to develop a web application. It will go through the process of designing the GUI (React.Js) to design database in MongoDB. It will implement the backend using Node.js and route the navigation using Express.Js. Main goal is not to develop a web application goal is to develop the application using only & only JavaScript frameworks and why these JavaScript frameworks are batter then the traditional web development libraries and frameworks. For this purpose, the following objectives are structured;

- To explore and develop an understanding of the previous technologies which are used in the website development.
- To design the website using the MERN Stack which helps to gain an insight of the features and the characteristics of this technology.
- To analyse and test the website which is developed from MERN Stack technology and its utilization in the real world.

2.2. Aims and objectives

To study and explore the MERN stack this thesis will focus on going through various technologies involved. An e-shop will be created at the end as the result of application of studying these technologies. After designing a mind map, Use cases, activity and other diagram. the first step will include design and implement in context of MongoDB – a NoSQL document based database. The next step will be to visualize the managed access to this data documents by using Postman. The backend will use Node.Js for the implementation. Once the data part is implemented, the need of Front end will be satisfied by developing it using React.js. Then after use Express.JS for routing and create connections in the application following payment gateway integration with Stripe.

3. Literature review

The e-commerce industry has been growing rapidly in recent years, with new technologies such as cloud computing, software as a service and mobile computing coming into play. Building e-commerce websites have become the most popular type of website for businesses. Therefore, this chapter covers the basics of e-commerce and the numerous e-commerce strategies available. It also aids in the development of the concept of web applications as well as the description of each of the MERN stack technologies. The first portion of this section is about e-commerce, while the second section is about MERN.

3.1. E-commerce

3.1.1. E-commerce definition

Electronic commerce or in short form E-commerce is a term that refers to online operations, purchases, and sales of products and facilities. The term "e-commerce" was coined in the 1960s. Throughout decades of growth, social networking further underlined the strength and explosion of the website as smartphones grew more prevalent. Commercial progress is aided by various launchers.

As previously stated, e-commerce is the activity of acquiring and exchanging physical items and services by an online platform. It entails the exchange of information or money between a number of stakeholders in an attempt to execute a transaction (3). It's part of the wider electronic enterprise or electronically business sector, which includes all of the processes required to run a company online. The terms e-business and e-commerce are sometimes interchangeably used.

E-commerce has enabled organizations (especially organizations with a restricted reach, like small businesses) to get exposure to and create a larger representation in the market by providing better and more effective distribution networks for goods or solutions (4). As a consequence of e-commerce, people's purchasing and use of goods and solutions has improved. Consumers are increasingly placing orders for items that may be delivered swiftly to their homes or offices using their computers or mobile phones. As a result, it has had a negative impact on the retail industry.

3.1.2. Types of E-commerce

3.1.2.1. Business-to-Business (B2B)

The computerized interchange of goods, facilities, or data among companies, instead of among organizations and consumers, is referred to as b2b e-commerce. Internet directory and goods and supplier interchange websites are examples of a facility that lets companies browse for commodities, solutions, and data while also initiating payments via e-procurement portals (5).

3.1.2.2. Business-to-Consumer (B2C)

Business-to-consumer e-commerce is the selling segment of the internet platform (B2C). It arises when businesses give products, services, or data to clients on an individual basis. There are several online companies and marketplaces on the computer currently that offer a variety of retail products (6). A good example of these websites is Amazon, which rules the B2C business.

3.1.2.3. Consumer-to- Consumer (C2C)

Consumer-to-consumer (C2C) e-commerce is a form of web commerce inside which consumers exchange goods, services, and data. These activities are frequently carried out using an online platform provided by a service provider. Online bidding and public advertisements are examples of C2C systems, with eBay and Craigslist being some of the most well-known (7).

3.1.2.4. Consumer-to- Business(C2B)

Consumer-to-business transactions (C2B) When people make their products and services available for businesses to contend with but acquire online, this is referred to as e-commerce. This is the absolute opposite of the traditional B2C business model (8). A C2B portal, such as iStock, is a marketplace that sells premium images, videos, multimedia, and design components.

3.1.2.5. Business-to- Administration (B2A)

The term "business-to-administration" refers to online transactions between corporations and government organizations or departments. Various elements of government rely on e-services or commodities in one way or another, notably when it comes to legislative documents, registrations, social assistance, fiscals, and recruiting (9). Businesses can deliver these services digitally. Previously, investments in e-government capabilities resulted in a large growth in B2A services.

3.1.2.6.Consumer-to- Administration (C2A)

Internet transactions between private consumers and federal agencies are referred to as "consumer-to-administration" (C2A) (9). The government seldom buys products or services from residents, but it does so often in the following states:

In the Education sector, these services include dissemination of data, distant learning/online seminars, and so on. In the Social security sector, it is a type of insurance. Data distribution, transaction processing, and so forth. In Taxes these services include Processing billing, completing tax returns, and so forth. These services include, among other things, scheduling appointments, providing illness records, and payment for medical services in the health sector.

3.1.3. Advantages of E-commerce

There are several benefits to e-commerce, and some of them are listed here. When it comes to accessibility. Apart from planned maintenance or downtime, e-commerce websites are available 24 hours a day, seven days a week, allowing consumers to browse and buy whenever they choose (4). Similarly, When it comes to accessibility. While congestion at a physical store might slow down consumers, e-commerce websites run quickly owing to processing and bandwidth limits on both the computer device and the e-commerce website. Web pages for goods and shopping carts emerge in a couple of seconds. An e-commerce purchase may be completed in less than five minutes with only a few taps (4).

There are a lot of alternatives. E-commerce enables firms to provide a large choice of products, which are then shipped from storage after a transaction is completed. If it's easy to find what they're looking for, visitors are more likely to discover it. In a physical store, shoppers may have difficulties determining where department an item belongs. Customers who visit an e-commerce site can browse item categories pages and use the website search tool to find a product fast.

It appeals to people all around the world. Brick-and-mortar businesses sell to customers who visit retail establishments. Businesses may use e-commerce to provide services to any customer who has access to the internet. E-commerce has the potential to expand a company's customer base (4). It is less costly. Despite the fact that they may incur shipping and storage charges, full-service e-commerce businesses save greatly on physical storefront expenditures such as leasing,

stock, and retail employees. Suggestions for items and customisation On e-commerce websites, users' browsing, searching, and purchasing behavior can be recorded. They may use this information to provide relevant and personalised product recommendations as well as gather crucial market intelligence.

3.2. Impact of e-Business on Traditional Markets

Today, e-Business has become the new tool for all businesses to manage their logistics and communication as well. The main benefit of e-Business is that it saves the company money on the costs of transportation, personnel, and other products. E-Business creates more opportunities for suppliers who can offer different solutions that are tailored to clients' needs by offering them multiple options so that they can choose according to their preferences. As e-Business is more developed, new possibilities for disintermediation emerge. For example, one could argue that if smartphone makers had used the Internet for their product development, they would not have had to go through manufacturers who were much older and probably more expensive. For example, does it make sense to buy a pair of sneakers from Nike? Then it might be better to order them from an internet service provider (ISP), which connects you directly with brands and stores them in your house without any intermediary costs (10).

E-commerce is shifting the role of distributors, retailers, distributors, and manufacturers. These changes are impacting traditional markets in sourcing, distribution, and retailing. The structure of the supply chain has evolved significantly due to the availability of multiple channels (Internet, mobile) used to conduct business in many different industries. In a competitive market, prices are expected to change frequently due to the competitive nature of a particular product. However, in an e-Business context, prices may only change once they have been decided on by the business owner, without any influence from their competitors. In this way, e-Business helps in reducing search costs and price dispersion; meaning that when the business owner looks for resources like labor or machinery at a certain time, this will not be affected by the actual supply and demand situation. However, businesses need to think about how exactly it would affect their cost structure. For example, more people were looking for part-time jobs after work than before because employers no longer offer full-time jobs as easily (11).

3.3. Success and Failure factors of E-Business

Success and Failure Factors of E-Business The world has been moving in the direction of e-business since the middle of the 1990s, but there is still much to be done. E-business is an evolving concept that encompasses all business functions that can be carried out electronically or over the Internet. It incorporates customer or client relationship management (CRM), marketing automation, and sales force automation, and reverses logistics and e-tailers. E-business platforms have created new opportunities and challenges (12). Take advantage of these opportunities to simplify your business and improve sales, reduce costs, and connect with customers in new ways. At the same time, be aware of the risks that accompany these changes in your business model and analyze how such changes could affect profitability. To take full advantage of the E-service, you need to look at your organization from an alternative perspective. The question is how to deal with these changes and at what cost, and at what speed. This is not the time for worries about "disintermediation". It is the time for cooperation, integration, and the consideration of customer loyalty, profitability, and competitive advantage (13).

3.3.1. Success factors in E-Business

The growing interest in e-business from a fresh angle provides an innovative, multidisciplinary perspective on this newly emerging subject. It examines how innovations emerge and why certain business models seem to create value for their users. Also focus on the drivers of value creation which include capabilities, processes, and organizational attributes. By using these three concepts as bases, the authors explore the main challenge that e-business faces today: how to create value through virtual channels without creating significant transaction costs or losses in productivity. The result is a unique book that opens up new horizons for business professionals who want to understand what is going on with their companies in this new era of E-Business. Opportunities, challenges, and critical success factors for E-commerce are presented in three phases (14). During the start-up phase opportunities may include commitment, content, and process improvement. As the business grows and evolves these opportunities become staff, after-sales service, and payment systems. In the growth phase, critical success factors include control, integration, and learning how to manage the competition.

E-business organizations typically face different challenges. They include getting the right information, assessing market needs, and developing the product. These organizations have to identify these factors to ensure they are competitive in their areas of e-commerce activities (13).

3.3.2. Failure factors in E-Business

Failures of e-business are a common occurrence in our world and many other businesses have been affected by such problems. The dot.com bust that took place at the turn of the century provides many examples of unsuccessful implementation of e-Business. The examples include Pathfinder and Argos. Both companies experienced huge losses due to poor Human-Computer Interaction attributes, such as navigation and layout, and an unsustainable revenue model. These things can lead to failed e-Business strategies that do not work out. One example of this is when a company targets a special consumer segment with different selling preference and plan each user to take action by filling out forms with few options (15). Such a business model requires a lot of research on consumer behavior, personal preferences, and demographic characteristics. There are many other examples of failed e-Business strategies in the dot.com bust, Argos, and Pathfinder which you should be aware of before implementing your business plan. To avoid the pitfalls of e-Business, it is important to understand your audience. The main reasons for the failure of most attempts in E-Business are lack of information about your customers and failure to adapt to change. The dot.com bust that took place at the turn of the century provides many examples of unsuccessful implementation of e-Business. If a company cannot successfully implement an e-Business idea and it fails, then that idea will be lost forever. The main reason why e-Businesses fail is that they are not prepared for the obstacles that they face when getting started. Exploring the market and learning about customer needs will help develop ideas that can be implemented successfully in the future (14).

3.4. Single Page website vs Multiple Page website

Single Page Website vs Multiple Page Website Single-page Websites or SPA is the latest way to build web applications. Instead of loading content or changing a page when a user clicks on something, SPAs fetch data on demand and dynamically update the content. MPA refreshes the entire page every time it receives a request from users, causing slow performance and a higher

load on the server (16). A single-page website will load faster, is easier to develop, and is easier to keep up-to-date. Since the content is displayed directly in the browser, you can integrate it with any number of APIs you may require. This can greatly increase functionality and provide an engaging user experience. A single-page application (SPA) is a web application that can be used to create a dynamic web page. The user interacts with the single page and sees changes immediately without refreshing the entire page. Because only a small part of the webpage is updated without reloading the entire webpage, there is a considerable reduction in server load. A SPA or Single-page app uses only a single page, even though it still has multiple HTML files. It loads the full HTML layout once and then updates the content while interacting with the user. A simple analogy would be a book; you read one page, turn a few more pages and then finish reading the whole book (17). Designing a website with multiple pages allows you to get the most out of your exposure by tapping into long-tail keywords. Not only do they allow you to optimize your SEO strategy, but they also increase the likelihood of potential leads staying on your business' site for longer than if it were a single page. This design method also allows you to create a more visually appealing user experience, depending on your target market and written language.

Multiple-page websites have always been more popular than single-page ones because of their ease of use and accessibility. However, with the continuous evolution of the internet and its users, SPA has come to be seen as a better option than MPA. A developer needs to choose an appropriate client-side software that will reduce development costs but still retain ease of use and accessibility. Although multiple-page websites have previously been more popular than single-page sites, this trend is changing. Single-page sites are now becoming more popular because they offer a better user experience and make lighter calls to the server (18). A developer who wants to build a single-page site needs to choose an appropriate client-side software that reduces the cost of development. Vue.js, Angular 4, and React.js are just a few examples of JavaScript client-side software that are popular today. Multi-page design is one of the most popular web design formats out there and it’s not surprising why. By breaking your content down into several pages, you can help search engines understand your website structure and serve better search results for your customers. This design method also allows you to create a more visually appealing user experience, depending on your target market and written language (19).

3.5. MERN Stack in Website Development

MERN is a JavaScript-based technology stack that works on Node.js and MongoDB to build quality web applications in a faster way. It enables developers to build scalable and fast applications that are easy to scale. In the MERN Stack, MongoDB is the database which is responsible for managing data and giving it a structure. Express is the web application framework which allows you to build highly scalable applications as well as websites easily. React.js is an open-source library that can be used to build user interfaces and Node.js is a server-side runtime environment or SRE that runs on Google Chrome's V8 engine and uses JavaScript both frontend & backend (1). The MERN Stack is the perfect solution for creating web applications, such as social networks, real-time messaging services, single-page applications and so forth. This complete, end-to-end stack allows you to build applications that are fast and highly scalable. The database is fully managed and scales automatically, while the development platform provides a rich set of libraries, frameworks, and tools all preconfigured and ready to use (20).

The MERN stack is a full-stack JavaScript solution that combines the MEAN stack (MongoDB, ExpressJS, Angular, Node.js) with React or Vue. Developers can use the MERN stack to work in a full-stack development environment. In other words, they get everything they need to build full-scalable apps, including the latest frontend development tools and technologies, right in the stack. MERN is a JavaScript-based development stack for modern web applications. It consists of four distinct, but complementary technologies: MongoDB (database), Express.js (server), React.js (frontend), and Node.js (backend) (21). MERN allows the programmer to work in the same place with all of these technologies in the most convenient way possible and save time on installing various dependencies separately. MERN is a JavaScript-based framework used to describe a set of technologies commonly used in the development process. It helps engineers achieve optimized results and more efficient functionality, ensuring that any newly developed application will be functioning at an acceptable level. The process of web application development can be broken down into many different steps. This is where MERN comes in, MERN provides developers with the resources needed to complete their job as efficiently and effectively as possible. From building their site

to deploying it, this framework is a culmination of almost everything an engineer would need to complete their tasks in one place (22).

3.6. Principle of Object-Oriented Programming (OOP)

The four essential concepts of OOP are abstraction, encapsulation, inheritance, and polymorphism, as well as classes and objects. The underlying premise of any object-oriented programming language is a mix of these two ideas.

3.6.1. Encapsulation

It's a data execution concealing approach that prevents public or open functions from being accessed. In order to do this, instance attributes are marked as protected, while accessor functions are marked as public (23).

3.6.2. Abstraction

A notion or concept that is not related to a specific event is referred to be "abstract." We use an abstract class/Interface to express the class's concept rather than the implementation stage. In certain aspects, one class does not need to understand the inner workings of another in order to benefit from it; merely recognizing the relationships should suffice.

3.6.3. Inheritance

To indicate the relationship between two elements, the phrases "is-a" and/or "has-a" are employed. Because of the inheritance feature, we may reuse the code of existent superclasses in derived classes. In Java, "is-a" refers to either class inheritance (through extensions) or the implementation of interfaces (using implements).

3.6.4. Polymorphism

"Single identity, numerous forms" is how it's translated. It comes in two varieties: sturdy and flexible. Static polymorphism is achieved by function overloading, whereas dynamic

polymorphism is achieved through function overriding (23). It has a great deal to do with heirlooms. User e can provide a feature that works on both the super-class and any subclasses.

3.7. OOP in JavaScript

JavaScript is a computer language that is object-oriented and cross-platform. Objects in the hosting system can be linked to JavaScript to create new methods to interact with them. JavaScript can be used to create both standalone files and applications, as well as embedded in HTML code (24). JavaScript has standardized libraries for objects, including Array, Math, and Date, as well as the core components of computer languages, such as administrators, controlling frameworks, and commands.

JavaScript is an event-driven language that is object-oriented and dynamically typed, meaning that the programmer does not need to specify details about the data used in a program but rather lets the computer decide how best to interpret it. JavaScript might be extended for several concepts by introducing objects, such as:

Client-side JavaScript: By creating objects Javascript is created that manages the browsers and the Document Object Model (DOM). Client-side modifications, for example, allow a program to impact elements on an HTML website and respond to consumer activity such as cursor glides, form input, and page switching (25).

Server-side JavaScript: The supplemental objects necessary to execute JavaScript on the server is implemented in JavaScript. This server-side extension, for example, allows a program to link to a server, move data regularly from one query to the other, and execute a program with some other functional file on the servers (26). NodeJS is an open-source platform that supports these features by simplifying its task management system, allowing the user to develop fantastic web applications.

3.8. MERN Stack

A web application is comprised of multiple technologies that need to be connected together, and this is known as “stack”. MERN is one of the latest and trendy full-stack JavaScript frameworks. MERN is a full-stack JavaScript framework based on React, Express, Node.js, and Mongo DB.

It combines the benefits of four top technologies today: React is a JavaScript library used in web and mobile applications, Express is a lightweight web application framework for Node.js that helps programmers create Restful APIs, Node.js is an event-driven I/O environment that can be utilized to design various scalable software products, and Mongo DB is a database. MERN does not require developers to master different technologies for building an application but mainly uses JavaScript (27).

The MERN stack supports a vast number of open source packages and dedicated communities from all over the world, allowing developers to increase their product's scalability and usability by reducing maintenance costs. With the MEAN stack, an entire web application development environment in a single package can be obtained. The Node.js runtime environment runs on each web server and provides access to useful modules such as sockets, file system operations, and database connectivity. Express takes this module set and extends it with easy-to-use middleware components designed to handle common tasks like routing HTTP requests and responses, making data available via JSON or XML, handling cookies and sessions, rendering HTML views in templates, and sending email via SMTP (1).

MERN is like MEAN in that it involves Mongo DB as its data set, yet substitutes React for Angular JS. Subsequently, it utilizes no front-end steering usefulness and consequently doesn't have the full capacity of a MVC system. There's something else to making a total web application besides picking an innovation stack. You likewise need to pick corresponding devices and libraries that make the cycle simpler, particularly when you get into enormous scope creation or working with genuine informational collections underway.

3.8.1. JavaScript

Navigating in the web has become a part of our daily lives. JavaScript is the technology that makes this possible, by allowing for dynamic, interactive websites. The history of JavaScript started in 1995 as a language created by Netscape employee, Brendan Eich. It was first named Mocha and was released under its current name in 1996 when it became an official ECMAScript standard. Since then, JavaScript has been steadily evolving. Today, 95% of websites use JavaScript as of March 2021 (21).

JavaScript (JS) has been around for the past 20 years. It's the language of the web and one of the most popular programming languages today. Despite being a young language,

JavaScript has evolved into a very mature programming language and can be used for much more than front-end web development. It is also commonly used to build server-side software using Node.js and can be run on both Windows and Linux operating systems. If you're looking for an easy way to make web pages interactive, JavaScript is the answer. Tools like JQuery UI, Bootstrap and WordPress can also be used to create websites that look good and work well across browsers and devices (smartphones and tablets).

3.8.2. Node.JS

Node.js is a fast and occasion-based JS runtime climate. Node.js utilizes occasion driven as the central idea for its current circumstance, which provides us with the different number of APIs that are occasion based and non-concurrent in nature which has helps in building the site utilizing node.js for back-end advancement. As we utilized Node.js, it utilized the relating callback work as indicated by our web application's business rationale. These callback capacities are executed separately, they don't rely upon the code written in which they show up, yet rather hang tight for the execution of the relating event (28).

Node.js is an event-driven platform that uses JavaScript for both the server and the client. At the point when Node.js processes an Input / Output activity, rather than hindering the string and squandering CPU cycles pausing, it will continue the tasks when their reaction returns. This permits Node.js to deal with great many simultaneous associations with a solitary server without presenting the weight of overseeing string simultaneously, which could produce a large number of bugs (29).

Node.js is a framework for quickly creating fast, scalable and flexible applications based on Chrome's JavaScript engine. Node.js is lightweight and efficient because its event-driven, non-blocking input output approach, which is ideal for computational real-time applications that operate across multiple devices. Node has grown in popularity due to its ease of use in developing high-performance, real-time online applications. Node allows you to utilize JavaScript from start to finish, on both the server and the client. Originally, JavaScript could only be used in a web browser, but due to high demand, it has been moved to the server. JavaScript has come a long way, and it now dominates server-side programming. Because Node uses JavaScript, which has security flaws, we need to investigate the security vulnerabilities in

Node apps. Rather than using threads, Node uses events. Node scales to millions of concurrent connections by using an event loop within a single thread rather than numerous threads. A single thread in Node may achieve high concurrency. Every input output action in Node is asynchronous, which means the server may continue executing incoming requests while the I/O operation is running. Because Node is likewise concurrent, AJAX may be incorrectly confused with Node, despite the fact that the two are vastly different (30).

The use of non-blocking, event-driven requests that execute in a single thread are the major distinguishing elements of the Node architecture. Concurrency is handled by traditional web servers by creating new threads for each new request, which can rapidly deplete available memory. Node is small, fast, and unique. Because of its unique features, it can support tens of thousands of simultaneous connections. Node may reach high concurrency rates even with minimal memory and a single thread by avoiding context changes across threads.

3.8.3. Express.JS

ExpressJS is a lightweight and minimalistic framework for Node.js that provides a set of features, various libraries, and different helpful tools to develop web applications. It can also be used in the browser. Express is highly flexible and powerful, but is not a full stack framework like Ember or Meteor. NodeJS is a wonderful JavaScript runtime environment that allows us to build server-side applications. However, by itself, NodeJS doesn't know how to handle HTTP requests, serve files, or respond with different HTTP methods. This is where Express JS comes in. Express JS extends the capabilities of NodeJS by isolating path segments and making routing easier. It also provides an elegant API for interacting with the request object (req) and response object (res), which are two fundamental components involved in building an API.

Utilizing ExpressJS permits you to utilize JavaScript on both ends of web development. It permits you to involve JavaScript in every stage of the development which results in saving a lot of time during improvement. The language is more productive since both frontend/backend engineers share a typical language, making correspondence simpler and makes your group more coordinated. As JavaScript continually advances, any new updates with JS will likewise be reflected in Express, permitting admittance to new improvements and updates as they show up (31).

3.8.4. MongoDB

MongoDB is an open-source document-oriented NoSQL database. It is developed by MongoDB Inc. To help developers design and build applications with high scalability and fast performance, MongoDB has set up a marketing program that provides free services such as free software, community service, training, and professional support around the world in different languages. This product includes basic functionality that can be used for commercial purposes at no cost.

MongoDB is an open-source, high-performance, schema-less NoSQL database. It is easy to set up and provides more agility than traditional databases. As data is written as JSON objects, it is simple to store and retrieve virtually any kind of data in a document. In MongoDB you can use dynamic queries (queries that are not determined until run time) and map Reduce (a powerful parallel processing tool capable of executing large tasks in seconds) for greater flexibility and control over the data stored in your database. Built with dynamic schemas and native JSON support, it stores data as documents that are sorted and indexed in a ternary tree structure, providing fast access to data that does not fit well into relational tables. A wide variety of applications use MongoDB for mobile app development (32).

Mongo DB functions utilizing records, which are documents that comprises of data structure made up of attribute and value pairs. Mongo DB's basic information unit is just the document. The documents resemble JavaScript Object Notation, although they employ a binary JSON version. The advantage of utilizing BSON is that it can handle a wider range of data formats. These documents have fields that are equivalent to cells in a relational database. According to the Mongo DB user handbook, the values included can be a number of data formats, including other documents and arrays. A main key could be used as a unique identifier in documents.

Collections are groups of documents that work in the same way as relational database tables do. Collections can hold any form of data; however, the data in a collection cannot be dispersed over many databases. Mongo DB's open source releases provide the mongo shell as a basic feature. Allows users to connect the mongo shell to their operating Mongo DB instances once Mongo DB is enabled. The mongo shell is a JavaScript connector to Mongo DB that helps individuals to search and update data, as well as perform admin tasks.

Mongo DB, like other NoSQL databases, does not need preset models. It can store any sort of information. This allows users to generate any amount of features in a document,

allowing Mongo DB systems evolve relatively easily than relational databases. The fact that documents translate to native data types in a variety of computer languages is one of the benefits of utilizing them. Embedded documents also decrease the requirement for database connections, which can save money. Mongo DB's horizontal scalability is a key feature that makes it a good choice for enterprises with large data applications. Furthermore, sharding enables the database to share data across a group of servers. Newer Mongo DB editions additionally permit the formation of data zones based on something like a shard id (33).

3.8.5. ReactJS

React is a JavaScript library that was developed by Facebook Inc. and is maintained by numerous free designers. It offers different augmentations for whole application building support. This library additionally makes it simpler to assemble huge applications with information that changes over the long run. You could have proactively used it without knowing it, Facebook App gets inherent part with React JS. It is used for building user interfaces. It is widely used within the web development community as a front-end tool that helps create interactive and fast web applications. It works efficiently and seamlessly with other open source libraries to create large projects easily.

The chief reason to choose React boils down to one simple advantage: its speed. When you build applications, the speed of your content delivery matters a lot. If it's fast, people will take less time to comprehend and interact with your app. As such, it will grab their attention for longer periods of time which can translate into more engagement and greater loyalty towards your brand based on return users. React is popular in part because it uses a virtual DOM (Document Object Model) to render changes quickly before they make their way to the real DOM. The result is faster load times and improved performance overall (34).

React.js provides solely HTML views, unlike other JavaScript libraries such as jQuery, which creates an MVC (Model, View, and Controller) design. Model controls the application's data, View presents the model for the client, and Controller updates the model when anything changes in MVC design. Views in a traditional MVC architecture listen to the models and update themselves as the model changes. When using the MVC approach because every time the view is modified, the model is updated, and that update may trigger another modification in

the view. For this sole purpose, React.js was created in the first place, to make dealing with data easier.

Declarative views are used in React.js. Because the developer just needs to create the component for each view, and React.js quickly updates only the part that has changed, clear views make the code more predictable and easier to maintain. This is accomplished through the use of Document Object Model, which is an in-memory data model developed to represent the data when the component is specified. React.js compares the virtual and actual Document Object Models and updates the actual DOM accordingly if there is a discrepancy. In comparison to jQuery's alternative method of updating the DOM, the computation of the difference is more expensive, but React.js provides an efficient method. The evaluation of the change is more expensive in React.js than in jQuery's alternative method of modifying the Document Object Model (35).

3.9. Benefits of using MERN Stack

MERN stack is the new technology in the web development sector which provides aid to the web developer in order to develop an application or a software very quickly with precision and maximum performance. MERN stack is an open-source technology of stack development which also provides end-to-end framework to the developers. In the present day, MERN stack technology is the most beneficial, efficient, and popular technology in order to develop software and web applications (36). The major benefits of using MERN stack are as follows.

3.9.1. Open-Source Technology

MERN stack is the open-source technology which means that every individual has the access of this technology and all of them can use this technology in the web development. Due to being an open-source technology, mostly freelancers use this technology to develop the application and efficient website for their online customers. Individuals who want to start their own business as a start-up, they also prefer MERN stack technology to get a kick start in their business by using most efficient technology. Moreover, open-source technology also means that there is no vendor lock in the technology, any individual can approach any section and change it as per the

requirements. Due to being an open-source technology, MERN stack is constantly improving by the experts from all around the globe.

3.9.2. Free Templates

According to the study by (37), since MERN stack is a new technology, there are many templates that are available on the internet which can be used in the development of any software and web. These free templates can save a lot of time for the individuals in the creation of new software. Developing new software using new template takes three times longer than the free template which can be saved through free templates of MERN stack. These free templates also belong to the experts of MERN stack that is why individuals who are using specific free templates can ask for help from the expert and creator of the template.

3.9.3. Help to Build Fast

MERN stack is an open-source which means so many ideas about the project are already available free on the internet which would help individuals to build new software and web faster than usual. This also means that developers do not have to make everything from the scratch, they can just use these free online sources and start working on them. Developers only have to pay detailed attention to whatever is required and find the suitable template for the project and complete the work accordingly (1).

3.9.4. Easy to Use

MERN stack technology was developed by expert who kept this technology free for everyone on the internet so every developer can easy approach this technology. According to the study by Nguyen (38), this technology is well documented in the internet from where any developer can easily get in and use it as per requirements. This technology is very easy to use and very simple to understand which is making it ideal for the new developer and encourage them to adopt this technology in their projects.

3.9.5. Full-Stack Development through MERN Technology

Full-stack development in the MERN stack technology means that the developer has to build all the components of the software both frontend and backend components. This also means that

the developer has to use the principles of UI design and software engineering as well. This full-stack nature of MERN stack technology makes it very popular among the entrepreneurs while saving their money to do frontend and backend programing separately.

3.9.6. Community Support

As per Tran (22), MERN stack technology has widely supported from the internet community due to its versatile nature in the development sector of web and software. This support makes it very easier to find the technical queries of the new developer on online sources. Whenever developers have the queries about any aspect of the technology, they go inside the community and ask questions and the community answer them perfectly which speeds up the development process and allowing them to finish their product faster and cheaper as compare to other stack technologies.

3.9.7. Worldwide Exposure

According to the study MERN stack technology was already adopted by hundreds of thousands of developers and experts in every part of the world which increased its worldwide exposure all around the globe. This technology has spread so widely that every developer can find the native experience while working on the projects. It can also provide a compelling experience for the developers who are using it in their products. This worldwide exposure also makes it easier to monetize the features of the real-life while developing and also allows developers to create real values rather than fixating of the things.

3.10. Comparison between MERN stack and Full Stack

According to the research conducted by full stack technology is a programming language and a set of tools which is used by the full stack developers. Full stack developer works on the frontend and backend as well to develop an application or a software single handed. Full stack developers work on the complete requirements and aspects of the application or software such as backend technology, frontend development, servers, database, API, and version controlling. The full stack developers must have ability to efficiently design the UI/UX and also many backend aspect of the application. Full stack developers has to start work from the scratch and keep

working on the product till it gets delivered while working on the entire function of the application. Whereas, MERN stack technology is the open-source technology which can easily be excessed by all the developers. MERN stack technology has free templates available online which can be used effectively by the developers which also allows developers to work on those free templates by making their work easy. Due to these templates, MERN stack developers do not have to start their work from scratch which save them a lot of time to work on the product. MERN stack developers also do not have to work on the entire product, these templates will help them to start work from the specific point.

According to the study, full stack technology has four step of working, frontend, backend, testing, and mobile app. Whereas, MERN stack technology has the working sequence such as mango DB, express JS, react JS, and note JS. MERN stack technology also covers the JavaScript as well. Developers who use the full stack technology has the benefits of having expertise in the multiple technologies and aspects of the web development. Full stack web development has to develop the project completely which requires a lot of time. On the other hand, MERN stack web developers can use open-source free templates that are available on internet which would take less time than full stack technology (32).

3.11. Comparison between MERN stack and MEAN Stack

MEAN stack technology is based on the JavaScript assembly language which is used to develop complicated web applications and websites for both responsive and progressive. MEAN stack technology is widely used in the industries where development are in demand. MEAN stack technology provides many benefits as the optimization tools for the administration system. MEAN stack technology is used to develop web apps, websites, and APIs to solve complex problems of the development and complete complex challenges easily. On the other hand, MERN stack technology is used by the beginners in the field of web development and it is also widely used by the entrepreneur because of its web development techniques which is easier than MEAN stack.

According to Aggarwal and (29), MEAN stack technology and MERN stack technology are very close related to each other in the context of the development. Whereas, MEAN stack is

the highly demanded in the industries while MERN stack is used in the creation of real-life software and web developments. Big industries like Forbes, Tumblr, YouTube, PayPal, PayTm use MEAN stack in their mobile applications and project development. On the other hand, MERN stack technology can be used in any development software and web that can be developed for any usage purpose. Components of the MEAN stack technology are mango DB, express JS, Angular, and node JS which provides the best framework for the industries.

However, MEAN stack has isomorphic coding structure which means that one coding database can be used for all kind of operating systems like iOS, android and Linux and the same code can be executed in all types of platforms without altering any code syntax. Whereas, in MERN stack technology has the end-to-end development method which helps the developers to use development tools of the technology and execute the application easily. MERN stack developers provides a lot of flexibility to the developers while MEAN stack is a bit complex technology which uses the cloud-based solution in order to development and testing purpose of the application. The major difference lies in their architecture where MEAN stack depends on the regular document object module (DOM) while MERN stack depends on the complete virtual DOM which makes it easier for the developers to access while using MERN stack technology.

4. Methodology

This section is dedicated to the project's methodology. There is discussion in this section about the Mern Stack, Front-end, and its subtopics such as React Js, its performance, Virtual DOM, and others. In the same section, Node.js, databases, and the software development life cycle are discussed in depth, as well as how these components work together to form a web application.

4.1. MERN Stack

It has been studied that the MERN stack refers to the dictionary which is used to explain the particular set of technologies which are JavaScript-based that are then utilized in the process of web application development. It is designed with an aim of creating the development process as smooth as possible. It has been studied that all these factors helps to play a significant role in the web application development procedure. This has provided the final framework for the engineers to do the work. The developers are continuously working towards this application to enhance the user interface of an application and to enhance the process of development for the purpose of building application to implement different projects within a given deadline. MERN is mainly based on the JavaScript, so it gives the developers an easy to develop projects as it only requires to know one coding language (1).

It includes the set of four pieces which includes the Mongo DB and React to build the applications. It enables the developers to build a 3-tier architecture which includes the frontend, backend and database completely using the JavaScript and the JSON. MERN consists of four open-source elements which consists of MongoDB as the database, express as the server framework, React.js used as the client library and the environment which is used to run the JavaScript is the Node.js on the server. These technologies help to provide an end-to-end web stack for the people to use in the development of application (29). The figure given below helps to represent the architecture of the MERN stack. First of all the Express is used with the Node.js to create an API which is then utilized for creating the logic and to associate with the MongoDB. When the client from React sends the request of HTTP to the back-end server, the server examines the request, take the data from the database and give the response back with the data. Then that react client gets updated with the data or the information which is retrieved.

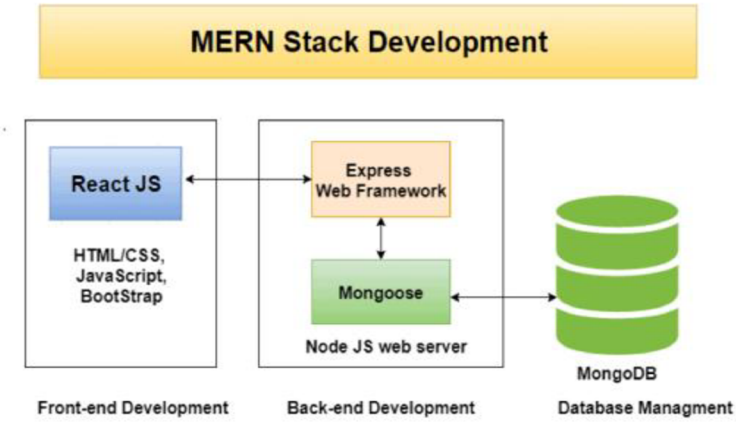


Figure 1MERN Architecture

4.2. FRONT END

4.2.1. React JS

React.js refers to the library of JavaScript which is designed to create the user web resources. React Virtual DOM refers to the completely memory and is also a web browser. React is usually build around the objects not to the templates like various other frames. React.js refers to the top tier of the MERN stack. It is considered to be the declarative framework which is used for building the client-side applications in a dynamic manner using the HTML. It helps to create the various interfaces with the help of different components. It then connects to the data which is present on the backend server and render them in the form of HTML (32). The strength of the React.js is the management of the stateful or the data driven front-ends which includes the minimal code. It includes all the features which is usually expected from the modern web framework such as the error management, management of events, list and many more.

4.2.2. Performance of React.js

It has been studied that the high performance of the react and various other frameworks is due to the scripts of JavaScript. The sole reason behind this is that these frameworks helps to give a better framework for developing the user interface rather than accomplishing a high level efficiency in any type of environment. Further the location of JavaScript uses the same base of the API of JavaScript. The difference between the performance lies on the usage of any of the

additional code of the framework. The main reason behind its usage in building the application is its straightforwardness because it does not have a functionality which is quite different from the single framework in order to run. React.js provides large amount of benefits as compared to various other frameworks (39).

Some of the common benefits includes that it provides the faster learning curve, support different programs from the company like Facebook and has a comparatively stronger documentation which is quite easy to read and provides useful services. User interfaces refers to the collection of many on-screen menus, search bars and things which helps to interact with the website. Before the development of react.js, developers mainly create the user interfaces using the Vanilla javascript. It is created by hand which ultimately results in more errors and bugs. Moreover it took a relatively long time to develop. Therefore in the year 2011, Facebook Engineer Jordan Walke built the React JS particularly to enhance the development of user interface (40). In addition to the providing reusable react library code, it basically comes with two main features which usually appeal the Javascript developers. This includes the JSX and Virtual DOM. For the purpose of getting a better understanding of react JS, its features are explained in detail in the below mentioned sections.

4.2.3. JSX

It has been studied that the HTML is considered to be the heart of the websites. The web browsers read the different documents and then display those documents on the systems, or the phone in the form of web pages. During all this procedure, browsers formulate something which is called as a DOM (document object model). This is considered to be the representational tree and shows how the specific web pages should be arranged and represented. The developers can add huge amount of dynamic content on the projects through the modification of the DOM with different languages such as JavaScript. JSX which is short form of JavaScript extension is referred to as the React extension which makes it easy for the number of developers. This helps to alter their respective DOM with the help of simple language of HTML (41). Further the browser of React JS is supported to all of the modern web browsers therefore JSX is largely compatible with any of the modern platform in which the developer has to work.

4.2.4. Virtual DOM

If the react JS or the JSX is not chosen or used by the developer, then the website will utilize the HTML for the purpose of updating their respective DOM. This refers to the procedure of making things change on the screen without the interference of any user. This would work absolutely fine for the static websites, however it would not work for the dynamic websites which includes the heavy interaction. If the developer utilizes the JSX for the purpose of manipulating the DOM, then the React JS would build something which is called as the Virtual DOM. As the name implies, Virtual DOM refers to the copy of the site of the DOM and the react JS utilizes this copy of the virtual DOM to watch which parts of the actual DOM requires to be altered when any of the event occurs. This particular type of addition or alteration usually takes less computational power and thus less loading time (42). The react helps to create the strong website which is considered to be a data-driven interfaces with the less involvement of code.

4.3. Node.JS

Node.js is considered to be the provider of JavaScript and is also the core of the MERN stack. It has been studied that it is widely utilized as a free open source web server environment. It enables to execute the JavaScript code on the server. It is capable to run on various platforms which includes the windows, Linux and Mac OS. It is greatly dependent on the Google V8 engine which is considered to be the main element of the Chrome browser. Moreover the languages such as C and C++ can be used in Node and V8. It gives the better performance in different aspects such as the speed and the consumption of memory. An application of Node runs in the individual procedure with the help of event looping. It does not need any kind of new thread which is required for each and every operation. This operates in an opposite manner to the conventional servers which uses the limited number of threads for the purpose of handling the requests (43). Further the Node is referred to as the non-blocking, and an event-driven engine which aims for a relatively scalable development of an application. Most commonly the node libraries are built utilizing the non-blocking pattern. A request is called by an application and then to slowly drifts towards the next task. It does not stalls while waiting for the response. On the completion of the request, a function of callback is used to inform the application about the outcome or the result. It then allows for the various associations or the requests which is send

to the server to be properly executed in a simultaneous manner. This is considered to be extremely important to scale the applications. MongoDB is invented and is utilized asynchronously (44). Due to this reason, the MongoDB is greatly compatible with the applications of Node.js.

4.4. Generation of Database

The database generation is the integral step of the generation process of the system through which a set of data can easily be generated. The database generation provides aid in order to generate the data which can be analysed to formulate the results. The database generation is also acquired in order to formulate much research which requires a specific amount of data. Generation of the database consists of many kinds of data, it also provides an ease to the developers in order to perform the many tasks such as web development, applications, and software (32).

4.4.1. NoSQL Database

NoSQL database is a type of database that can be generated in order to perform many tasks of the database. There are two types of NoSQL databases which have a big difference in their nature where the mechanism NoSQL database stores data and the relational database perform tasks. On the basis of the data model, the NoSQL database can be differentiated into four types, key-volume database, graph database, document database, and wide-column database. All these kinds of data are different from each other but all of them shares the dynamic schema which is one common thing in all database. These databases can easily be scaled with the large loads of data and high volumes of the users. NoSQL can also easily handle the relationship data very well. So many developers believed that this model of databases is easier for the NoSQL database due to its nested data structure (45).

According to the study by Gauthier (46), during the late 2000s, it was discovered that the creation of the complex database is not needed anymore because this complicated data model creates duplicate data which can be avoided through this new database. NoSQL databases developed the database system significantly and provided sufficient options for the developers and many large volumes of data applications. The number of applications that are needed to

store the database is increasing significantly day by day and the requirement for the database applications is increasing. For the schema, it is almost impossible to absorb all the databases at once which is causing a hindrance in the storage of the database. Therefore, NoSQL databases are the innovation in the database storage technology which provided the flexibility to the developers in order to store a very large amount of data at once and structure the data itself through its process. This process helps the developers to sort the data in the structured form.

4.5. Software development life cycle

In current society, managing development of projects related to softwares is an obvious need. A approach can be considered as a layout. Software engineering is a discipline concerned with the development of strong software for computer networks. This field mainly focuses on significance of standard, procedures, techniques, and tools used in the creation of a software. The development process establishes a system for various responsibilities and processes. During software development, the process specifies which actions must be fulfilled (or method of developing the software). Software project development management explains how to execute software techniques such as interaction, requirement specification, architecture design modelling, software construction, validation, and maintenance in a technical manner. SDLC can be defined as the method for accomplishing software projects on schedule and with top standard. SDLC, outlines the actions that must be completed during the creation of a system. Software development is made up of a series of processes that enable any software development firm to readily manage the end product. To finish the software development process, the SDLC models take a step-by-step technique. If the method is solid, the end product will be reliable, and the project will succeed (47).

The designers who seem to be explicitly or implicitly involved in the development of a quality software product should keep the following considerations in mind while doing so.

- A quality concentration
- Procedures
- Methodologies

4.6. Software development life cycle stages

The SDLC stages include:

4.6.1. NoSQL Database

Plan along with requirements gathering are the most major aspects of every life cycle process. After a session with the client, product is developed by prominent members. The necessity for client satisfaction, threat assessment, and a development plan are all important aspects of this phase (36).

The stage is important because it defines what a product must do and how it must work. This stage enables programmer to avoid designing something that will fail when used in the real world, improves the ability to communicate what client want from users, and ensures that both client and vendor are on the same page. In this phase, planning along with requirements gathering are the most main aspects of every life cycle process.

4.6.2. Defining Requirement

Following the completion of the strategy and need assessments, the next step is to describe the objectives in depth, document them, and obtain client consent. This stage's outcome is the SRS document containing all of the specifications of the project to be produced and constructed (36).

The goal of this stage is to develop a thorough understanding of all aspects of application of the strategy, including assessing key needs, detailing the solution and its essential characteristics, identifying risks and other project constraints, and developing an effective plan for achieving the objectives.

4.6.3. Designing the Software Architecture

Designs a product is an iterative process that starts by defining the requirements for a new piece of software. This information is then used to determine the right software structure, which consists of multiple layers of functionality and security checks. Once everything has been built and tested, it is time to deploy this system in production. Therefore, the software's structure is

constructed utilising the software requirement definition as a starting point. Multiple designs are constructed at first, and the best one is picked once it has been vetted by key stakeholders based on threat assessment, dependability, structural versatility, schedule, and budget (36).

4.6.4. Building or Developing the Product

The real designing of the project starts in this phase in conformity with the presented framework. If the previous step went smoothly, this stage should not be too difficult. Compilers, interpreters, and debuggers are among the equipment used by developers to create code. (36). Compilers transform a human-readable source code into machine understandable assembly language which is later processed by the CPU. Interpreters translate source codes into high-level programming languages. Debugging tools aid program developers in finding inaccurate results or inefficient algorithms so that bugs can be fixed as early as possible.

4.6.5. Testing

Throughout this stage, the final version of project is assessed for determining if it meets the requirements of the clients as outlined in the software design brief. To achieve a good product, software faults are found, tracked, rectified, and reevaluated. After it has been extensively tested, the service is then deployed within the true context and later maintained. The process of a customer proving his wants is known as acceptance testing. Based on client feedback, more enhancements are implemented. (36).

4.7. Models of SDLC

4.7.1. The Waterfall model

The waterfall model can be considered as a sequent operating paradigm that does not overlap. It means that the next phase is unable to start until the preceding one is completed.. It is straightforward and simple to comprehend. Below is a diagrammatic representation of the water fall model.

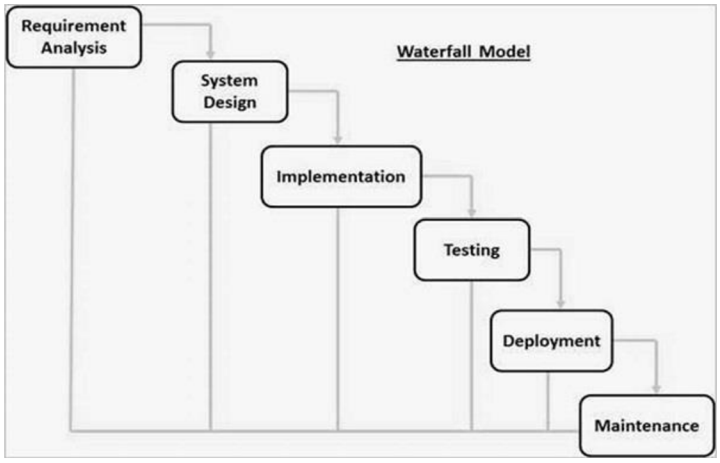


Figure 2Waterfall Model

The waterfall model consists of the following stages.

4.7.1.1. Requirement Analysis

This stage includes gatheing of software project's needs and documents them in a software specification document.

4.7.1.2. System Design

In requirements analysis stage, the core architecture of the software product is defined. The architecture is a plan for structuring a project's solution that reflects how it will be built and linked together. In this phase, the architecture designer lays down an idea of what a possible product would look like and the purpose behind each part.

4.7.1.3. Implementation

In this phase, software development begins. It develops in units, which are little chunks of code. These pieces are then evaluated for functionality before being merged in the next stage.

4.7.1.4. Testing

In this stage, integrate all the units generated during the implementation stage. After integrating the units, the entire product is validated to see if it satisfies its objectives. Defects and problems in software are reported, and if they are found, they are rectified and reevaluated.

4.7.1.5. Deployment

After the software has been validated and certified for both functional and non-functional requirements, it is released in the user's system.

4.7.1.6. Maintenance

If there are any issues in the user's system, these should be resolved during the maintenance stage. If the user is not completely pleased, certain enhancements can be made during this phase.

Advantages:

- Simple to comprehend
- Determine first, then build
- Utilized for product development having clear schedules.

Disadvantages:

- Idolized
- Issues are really not discovered unless they're tested.
- There is a lack of clarity in the objectives.
- Delay in delivering.
- Managing risk is difficult to implement.
- Making additional adjustments is challenging.

4.7.2. The Iterative Waterfall model

The shortcomings of the traditional waterfall paradigm prompted the development of another better approach. The iterative waterfall waterfall model was created to address the shortcomings of the traditional one. It is a more advanced version of the traditional model that can produce faster results with less effort and greater adaptability. The iterative methodology divides the

project into manageable segments, allowing the development team to focus on their main objective more easily and rapidly while also receiving vital feedback from customers. Each aspect of the project, which is broken down into smaller chunks, is a micro waterfall process.

Advantages:

- A far improved systems development paradigm.
- The customer may receive feedback.
- When the objectives aren't obvious, this method is utilized.
- Procedure that is guided by documentation.
- When used on a team that is inferior, it is very effective.

Disadvantages:

- Managing is difficult.
- Milestones are not properly defined.

4.7.3. The V model

The V Model is an improved waterfall model wherein validation features are implemented at every phase of product creation rather than at the end, resulting in improved product creation. In this paradigm, one cannot go onto next phase until one has completed the previous one. One does not deviate from the product aim under this framework since each phase is tested. The V model grants a higher level of product delivery, as well as better stability and quality.

4.7.4. The Spiral model

Spiral model can be considered as a blend of systemized processes that lead the benefits of iterative development and combines them with the convenience of the waterfall approach, as well as extra risk assessment components. The Spiral model's operation is splitted into four stages (identifying, designing, building, evaluating, and risk analyzing), which are repeated until the project is completed. This strategy allows for gradual updates to software application releases. The spiral approach is most adapted for completely individual software

applications since user involvement and assessment begin early in the design process. However, one run the risk of building a never-ending spiral for a project that never ends.

- The approach improves level of risk assessment.
- Quick software development in the entire lifecycle.
- Appropriate for huge projects
- There is a slim risk of failure
- After some spiral, progress can be stopped and a workable system will be accessible.

4.8. Waterfall Model Application

One SDLC model cannot possibly accommodate all sorts of software packages. There are many software development process models to choose from when building an SDLC. The model used is actually a tradeoff between the various factors that affect the overall system performance. An optimal software process model will provide optimum performance and help reduce costs throughout the project lifecycle. As a result, selecting the optimal software process model for a certain product is critical.

4.8.1. When to select waterfall model

- When software components are quite well understood, defined, clarified, and corrected..
- Stability of software design.
- Tech is well-understood and should not be spontaneous..
- Software requirements are clear and unambiguous.
- Availaility of asked resources.
- When product is not large.

4.9. Reason to select Waterfall model

The application being created using mern stack can be developed using one of the software development life cycle approaches i.e. Waterfall model. The reason is that the application is small, all the resource required are available, the software needs are clear and unambigous and

all the functional and non-functional requirements are completely understood and documented. Another reason is that the software developers can focus on implementing feature that are required and work on those that are not needed.

Since Waterfall model is the software life cycle that starts with requirements gathering and ends at development. It is a sequential approach where each step has only one objective and ensures that requirements are met before proceeding with the next step of the process. Waterfall model helps to define business goals, prioritize problems and explicitly document them as a set of functional requirements, technical requirements and acceptance criteria. Morever, all the details provided by the client are easy to understand and can be implemented without any difficulty. If there was any risk parameter in the project development then spiral model is used.

5. Results

This chapter covers topics including front-end and back-end applications. It also covers the controllers, which are responsible for accepting user requests and generating the response on the server. Database management topics such as databases, storage engines, and modules are also introduced in this chapter. Further more the code or the built version of the E-shop is uploaded on the GIT: https://github.com/gaurang5899/Mern

5.1. Front-end

5.1.1. Homepage

An app's home page is its primary web page. The term may also refer to the homepage shown in a browser when the application first opens. Figure (1) represents the homepage of the e-commerce-based web application. The home page of this web application contains a list of t-shirts, which are saved in the database; they can be viewed by clicking on any of those buttons labelled with a bullet.

Visitors would be able to view all varieties of T-shirts on the main page. For instance, this page offers a variety of T-shirts, including one T-shirt, two T-shirts, three T-shirts, and so many more. If the customer wishes to add a new style of T-shirt, this could be easily entered into the database. Similarly, a variety of T-shirt content can be incorporated in a similar manner. Numerous rates can be readily adjusted for various t-shirts depending upon their quality. Consumers can add products they want to their shopping carts. If the consumer has any problems with the items, rates, or anything else, he will have an alternative way to approach the administrator. All of these elements may be seen in the figure below.

Figure 3Home page

5.1.2. Sign in and Sign up

When a user who wants to create a new user account or to sign in goes to the page to create an account or sign up, the user will be able to choose the "Sign in" or "Sign up" option. After choosing, if the person isn't logged in at that time, then the "Sign in" menu item will be visible on top of other menu items.

If the person is logged in, then the "Sign up" menu item will also be visible on top of others. Afterwards, if the user changes the mind after choosing one option and wants to go back either by selecting the old one(e.g., "Sign in") or selecting another one(e.g., "Sign up"), then those options can come back again by just clicking them both. The above two options link to a webpage where the user can fill out a form to register an id or sign in to an existing account. These options can be observed in Figures (b) and (c).

A wise and organised approach to database management is always to register users in the system. A website must have an easy-to-use registration process in order for visitors to find it to be user-friendly. The signup form for this project is pretty straightforward and asks for the user's name, email address and password. Users must enter their username, email address, and matching credentials. One email address cannot be used for two accounts. The error

message "Sorry, email is already taken, please try again with another email" will appear on screen if the email has already been registered and is stored in the system database. If all needed forms are completed, registration will be successful and the record will be kept in the database until the user specifically requests that their user account be edited or deleted. The registration form page is attactched below.

Once a person has successfully registered their information on the website, they may log in using the right username and password they chose while registering. The error message such as login credentials are wrong, please try again will appear on screen when you input an erroneous email address or password. If a person has trouble remembering their password, there is a forgot password option accessible. The user receives an email with a link that allows them to reset their password and establish a new one for future use.

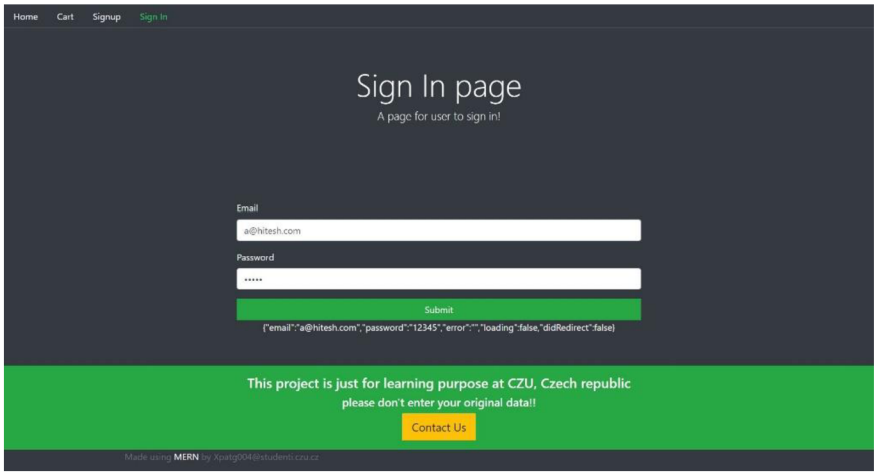


Figure 4Signin Page

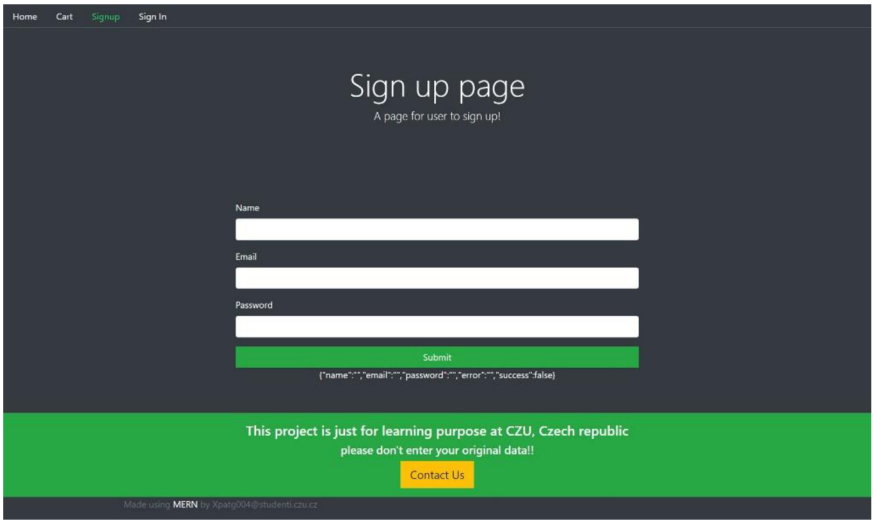


Figure 5Signup Page

5.1.3. Cart

User interface can simply be put as a connection between human and a computer on a webpage, applications or in any devices. The user interface should be efficient, simple, easy and user friendly in order to be successful. After a user is logged into a system, it is very essential to give them their freedom of changing their own details provided to website. Since the goal of this project was to create a very userfriendly website for e-commerce store, we have provided users that opportunity to tweak and edit their account details later on even after registering an account. Users may change their name, passcode, email id, profile picture, and contact details through this edit account interface.

This part is perhaps the most essential aspect of the web - based application. All of the company's merchandise are aggregated here. Customers can browse their merchandise on this webpage upon picking it, and the transaction will be processed here. This cart has many payment methods that allow customers to make payments using debit cards, credit cards, and UPIs. Figure shows the cart page of the e-commerce-based online application (d).

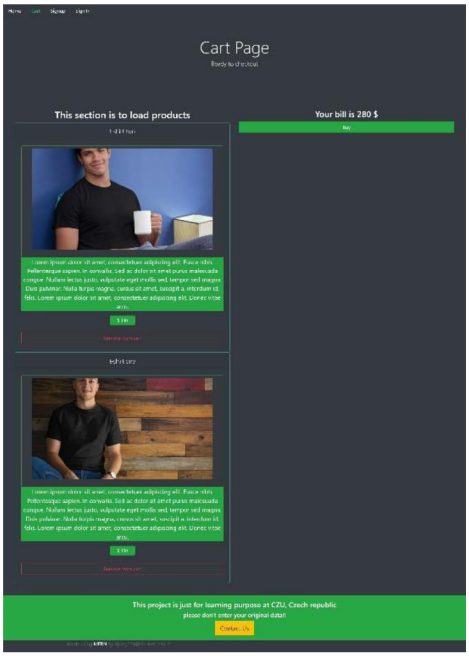


Figure 6Cart Page

5.1.4. Dashboard for Admin

This webpage cannot be utilized by the consumer since it is only accessible to administrators; thus, it is recognized as the Admin Dashboard. Admin Platform have the ability to generate several categories and add merchandise to all those sections, as well as to deactivate or disable products and upgrade their rates; these procedures can be seen in figure (e), which reflects the admin panel page.

With the help of the Admin navigation section, the admin can use navigate multiple options like managing the orders, managing the products, and even can manage the categorization of the t-shirts. Similarly, the admin can also create a space for a new product or categorizes it.

Administrators and staff members of the shop mostly utilize the interface to manage the system. However, only admins have full authority to carry out CRUD operations. Employees at the shop have access to enter stock, see customer orders, and monitor sales and expenses. The system may be used by the staff to generate either monthly or yearly reports. Personnel will be

alerted via a system's notification feature when the inventory meets the benchmark set by the administrators. Additionally, a message will show up as customers place new orders.

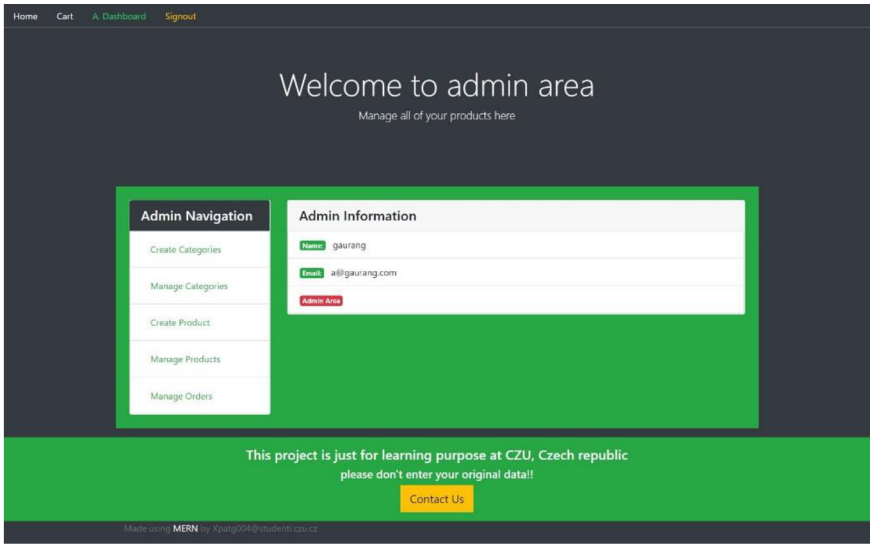


Figure 7Admin Panel

Similarly, figure (7) represents the page from where the admin can perform multiple operations like product upgradation or deleting the product. The admin can perform the same operations as previously mentioned for this page. To perform the upgrading procedure, the administrator has to click on the "Update" button and provide all the relevant data. Similarly, to perform the deletion procedure, the administrator has to click on the "delete" button and the related product or category is deleted or disabled easily by employing such steps. In a similar manner, figure (8) represents the admin dashboard for the categories based on the season.

Figure 8Admin Dashboard All products list

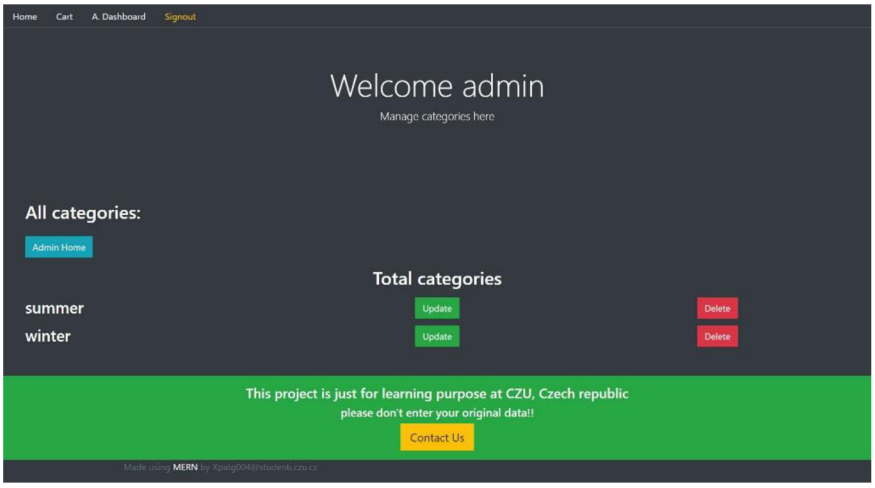


Figure 9Categories Page

5.2. Chapter Summary

This chapter discusses various components related to the e-commerce-based web application. All of these components are necessary because if one is missing, the operation cannot be

performed. However, detailed information about these components is provided in previous chapters, and back-end operations are discussed in depth along with the results in the following section.

6. Discussion & Analysis

In this section, there is a discussion and analysis of the implementation process from the front-end to the back-end of the e-commerce-based web application. Based on the limitations of this dissertation, it is impossible to discuss every phase of the project in depth. However, some of the fundamental and important components required to develop the MERN-based application are discussed. In a similar manner, the fundamentals of some third-party library resources or modules are also described.

6.1. Models

Models define the data structure as it must be in the central database. By utilizing several configurations that aid in the storage of information collected in a central database, such as Mongoose, one of the most well-known library resources in NodeJS, and Mapping Schema, an essential part of Mongoose. Mongoose creates conceptual frameworks that can acknowledge the identities and different kinds of data. To characterize interactions and characteristics, the library employs an intuitive framework and an abstract object syntactic model.

Therefore, Models are the fundamental concept for all information stored in MongoDB. A model is a description of how an object (collection) of documents is persisted, i.e., when a programmer inserts a document using the insert() method into an index collection, the operation will be as if they created it from scratch. Note that each model gets its own schema and can have any number of collections. For this project, different models have been created and their respective schemas too and then utilize as a table to store the data. Various Mongoose features will be utilized to validate e.g find a specific model with a certain parameter value.

6.2. Routers

This is the backbone of the entire website. Every page, every slider, and every website element can be created in this directory. In short, this is where all of the page routing work was completed. ExpressJS is a fast, flexible, and minimal framework for building web applications. Anyone can also use it to make RESTful APIs. It's widely used in the Node ecosystem for developing single-page apps, web applications, and APIs with Rails-like syntax. This directory contains CURD operations and routing-related code.

The configuration file contains all of the general information, including the website URL, domain name and the platform that you want to build the app. Everything in this directory is set up to be used as an ExpressJS app. This directory contains all of the code that allowed the developers to manage users, create levels, and progress through an effective response to their clicks. As a result, this directory serves as the website's foundation. Every page and every slider and website element can be created in this directory. In short, most of the page routing was completed here.

6.3. Controllers

The proposed method and tool support the definition of a network environment. The interpretations of the operations proclaimed in the route will be deposited in controllers, as will the middleware codes. The function interpretations of the functions asserted in the routers will be completed during the controller phase. The controller phase begins with the assembly of the Routing Information Base (RIB) and the functions that complete a router. Predefined Routing Information Bases (RIBs) provide a summary of all possible routes between any two routing tables, along with information on the connectivity among various components in the IP world. RIAs can be used to define a network topology, verify it, or construct a new one. During this phase, logic to compute a linear network from routes and then select the best path between two endpoints can be built. During the controller phase, logic to compute a linear network from routes and then select the best path between two endpoints can be built. The controller phase usually takes half the time that was spent in the design phase. The interpretation of the operations will be completed during this phase, and along with it, a summary of the functionality of each function will be compiled. This can then be sent to other teams to find out how they want their pieces to interact with one another. Various middleware are also defined here.

6.4. DATABASE

A database is a collection of information. Typically, to manage data access and updates, all database systems use some kind of management module which can create, update, and delete the data stored in the database. Nowadays, there are numerous database management systems in use. MongoDB serves as the database for this project. By using the Mongoose library,

MongoDB can be connected. This library contains numerous methods for creating schemas and saving data to databases.

The routing design phase ends with the completion of a network definition and the creation of middleware. Middleware is a set of software components that can be installed on computers over a network (such as TCP/IP). It provides services, called "functionality", to enable communications between networks and systems, such as a router or switch. The control theory is concerned with making sure that each piece of middleware performs its task in an orderly manner without causing any problems for other pieces.

6.4.1. . Creation of Databases

A node package called Mongoose is installed to connect the MongoDB database to the Express.js app. Mongoose is an object data modelling tool that helps MongoDB create schemas. Models are defined based on the schema interface. The customer can describe the fields reserved in each file, as well as their verification techniques and initial values, using the schema (Raju, 2021). The schema is essentially a framework for building the DBMS.

Robo 3T is a multi-platform pictorial consumer interaction technique for organising MongoDB workflows that is compact, accessible, and based on the casing. Robo 3T allows users to generate databases, and collections, add users and documents, run one-time queries with auto-completion, and visualise outcomes via a graphical interface.

6.5. Back-end Initial Configuration

The initial step is to create an Express app by generating an app.js script, that serves as the program's entry point and stores several of the required middleware and node packs. The application variable defines the Express instance.

Figure 10App.js Middleware

The following are the functions of the middlewares depicted in the above figure: Morgan records the details of the request. bodyParser retrieves the entire body portion of user requests and stores it in the req.body object. In addition, Middlewares perform two functions in Express, they are request and response. Request middlewares receive input from the user and get modified before sending it to the API, this doesn't fulfil their responsibility completely because response middlewares perform more than just modification of an API request. The CookieParser dismantles the cookie header and, under the cookies property, stores all cookies in the object of request. Dotenv allows programmers to utilise environment variables by acquiring the.env file, and Mongoose's objective has already been addressed previously.

The next step is to interact with the DBMs. As summarized in the previous sections, MongoDB Atlas is utilized because of the numerous benefits it offers to the operation. Atlas is a cloud-based database management service, so no deployment is permitted. By visiting the MongoDB Atlas authorized website and following the instructions, a cluster is generated with the programmer's individual preference of cloud vendor and territory, accompanied by an interface string stored as the MONGO URI attribute in the environmental script. Mongoose then uses it to communicate with the database. Once the interaction is established, the command prompt displays "DB Connected." Or else, a statement indicating an access error is published, as illustrated in Figure (9).

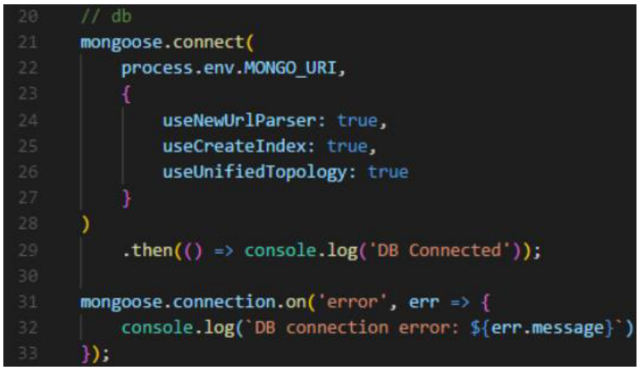


Figure 11DB connectivity with mongoose

The strategy for handling the back-end package and its various sections is the most vital aspect of any application. Therefore, the back-end package is divided into three sections, which are models, routes, and controllers. Whereas the models constitute the whole of the mongoose schemas that are required by the app, routes characterize the whole of the routes of the API, and controllers encompass the logic that must be executed upon every arriving request that matches with the associated route. This helps in designing a robust and scalable application architecture.

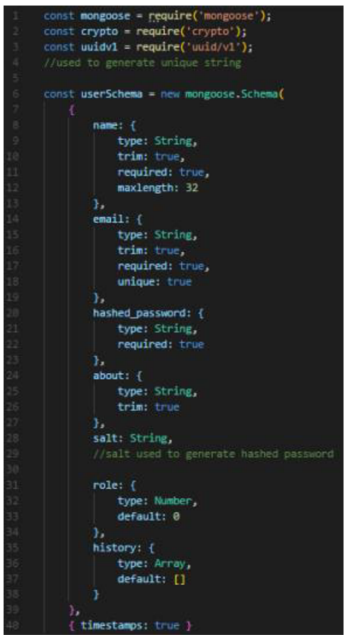
6.6. Designing a Mongoose Schema for back-end

The Mongoose schema is a standardized schema project. To begin the creation process, the programmer must first generate some schemas with Mongoose's Schema. This is equivalent to creating a model in the traditional relational database world. Each mongoose schema requires a user, product, category, and order schema (where product is the document of the product schema and category is the collection of products). This application implementation contains four schemas: product, category, order, and user. Therefore, each mongoose schema denotes a manuscript architecture. Even though a user template is required to manage verification and permission processes, a user schema is created initially.

6.6.1. . User Schema

A user schema, as shown in figure (10), has three essential attributes: user name, email ID, and password. The role property is fixed to 0 by definition, indicating that it is a standard customer. The user will become an administrator by configuring the character value to another (replace

the value 0 to 1 for administrator). To be incorporated into the DBMS, the user model just recognizes the characteristics described in the framework. Unlike history, which demonstrates an array of items purchased and made by the consumer, timestamps demonstrate the period of formation and adjustment of a user entity.



```
const mongoose = require('mongoose');
const crypto = require('crypto');
const uuidv1 = require('uuid/v1');
//used to generate unique string

const userSchema = new mongoose.Schema(
    {
        name: {
            type: String,
            trim: true,
            required: true,
            maxlength: 32
        },
        email: {
            type: String,
            trim: true,
            required: true,
            unique: true
        },
        hashed_password: {
            type: String,
            required: true
        },
        about: {
            type: String,
            trim: true
        },
        salt: String,
        //salt used to generate hashed password

        role: {
            type: Number,
            default: 0
        },
        history: {
            type: Array,
            default: []
        }
    },
    { timestamps: true }
```

Figure 12Schema example

For safety purposes, only an encoded passcode is stored in the DBMS; thus, the programmer utilized Crypto, which is a Node plugin, to parse the passcodes while combining them with a platform termed Mongoose Virtual. A Mongoose Virtual is an unstored characteristic in MongoDB. Mongoose Virtual, which also includes setter and getter techniques, is an excellent choice for processing characteristics.

```
// virtual field
userSchema
    .virtual('password')
    //password from client side
    .set(function (password) {
        this.salt = uuidv1();
        this.hashed_password = this.encryptPassword(password);
    });

userSchema.methods = {
    authenticate: function (plainText) {
        return this.encryptPassword(plainText) === this.hashed_password;
    },

    encryptPassword: function (password) {
        if (!password) return '';
        try {
            return crypto
                .createHmac('sha1', this.salt)
                .update(password)
                .digest('hex');
        } catch (err) {
            return '';
        }
    }
};

module.exports = mongoose.model('User', userSchema);
```

Figure 13Password schema with storage hash

The figure above illustrates how a 'hashed password' is generated using Mongoose Virtual. Initially, Virtual generates a password property for the object of the user. Whenever a new user is created, the significance of the req.body's 'password' property is established to the virtual property and forwarded as an assertion to the setter feature. The passcode is then encoded and assigned to the 'hashed password' characteristic. In addition, a function is delegated to the userSchema technique object in order to generate an authorization technique that compares the encoded source passcodes with the hashed one in order to validate the consumer.

6.6.2. Product Schema

Figure (12) depicts the architecture of a product schema. It has plenty of characteristics that every digital commerce product requires, such as a title, details, cost, volume, type, and so on. The selling field is initially established at 0 and is adjusted after every consumer acquisition. The category field in the user schema earlier in this section has an aspect of object ID, whereas the "ref" option defines what model to relate to during the population process. This shows the association between the product model and the category model, and by utilizing the populate technique, category statistics are modified with a mongoose document harvested from the DBMs rather than its authentic _id.

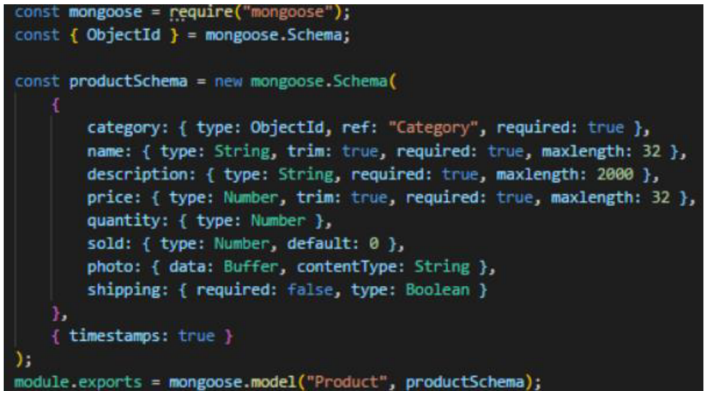


Figure 14Product schema

Even as user and product schemas appear to be complicated, the category schema is quite simple and concise, with hardly any name and timestamp fields.

6.7. API and Routing Documentation

Distribution and sharing of data among multiple systems has become an important component in software creation. In the scenario where a consumer searches for documents in a particular genre, an API is invoked and the application is forwarded to the server system. The server analyses the application, takes the essential course of action, and ultimately returns to the consumer a list of documents as a reaction. That's how the REST API actually works. An API (Application Programming Interface) is a collection of regulations that enable multiple apps or programs to communicate with one another.

REST regulates the appearance of the API. It is an abbreviation for "Representational State Transfer," which is a conceptual description of the most widely used Internet communication technology. REST is a protocol that allows multiple computer frameworks to interact over HTTP in the same way that web servers do. It is a collection of principles that development teams must pursue when creating an API.

6.7.1. Routing and Middleware

Routing governs how endpoints (URIs) in an application communicate with customer demand. The fundamental syntax is comprised of an object as an example of Express and the appropriate HTTP request technique, including app.get() and app.post() to handle GET and POST requests.

In the request-response phase of an application, middleware functions can encounter the "request object", the "response object", and the "next" function. Middle-ware implementation returns a result, which can be both the ultimate outcome or an argument to the next middle-ware till the stage is completed (1).

The Express router's 'next' function begins a further middleware immediately after the existing middleware finishes. If a middle-ware does not complete the req-res process, the next () function must be called to pass control to the next one. Alternatively, the request will be withdrawn. An express app is essentially a collection of middleware requests or calls. The call requests are determined by the order in which they are declared.

6.8. API and Routing Documentation

APIs interact using a system of regulations that characterise how computer systems, implementations, or devices can interact with one another. The majority of web APIs exist between the application and the database server. The customers make an API call instructing the implementation to go and do a particular task, and the application then uses an API to instruct the network connection to do anything. The API acts as a bridge between the implementation and the web host, and the API call represents the request. And, whenever a person accesses technology for communication with other software or online web servers, APIs are used to receive the necessary data. Finally, the API behaves as a go-between for any two machines that need to communicate with each other for a specific task.

6.9. Postman & its usage to request post routes

"Postman" is a customer resource used for API integration testing. Testing entails gathering APIs and determining whether they perform as expected for capabilities, serviceability, effectiveness, and stability, as well as returning the appropriate approach. API testing determines whether the outcome is well-structured and beneficial to some other application.

All of the application's APIs were tested in Postman by submitting a request to the server-side and receiving the reactions. It operates on the web server and ensures that every API end-point functions properly. Postman offers a set of API calls, which must be followed in order

to evaluate application APIs. The informational body, status code, and headers comprise an API reaction (21).

A requirement like addition, removal, or update can submit variables, access control details, or any necessary information. When a request is made, Postman displays the response from the API server in a manner that enables the initial reaction to be analysed, visualised, and remediated if required. The user can save all of the responses to a requirement and redeploy them as needed.

The Postman supports a variety of request methods. The four most common request methodologies in the apps are POST Request: This type of request is used to create or update data.PUT Request: For data updating, GET requests are used to retrieve and fetch data, while DELETE requests are used to erase data .

6.10. Front-end development

Front-end development is the process of making websites use the MERN stack. The MERN stack is a collection of tools designed to handle development and maintenance for JavaScript, CSS, and template files. The MERN stack refers to having them all in one place so that you can do your front-end development with the tools you know best rather than jumping through hoops of knowing how to do CSS selectors with Javascript.

6.10.1. Setup and Routing

Unlike back-end, front-end development is performed with only React.js. It is not impossible to set up a React application from scratch, but this process includes many intimidating and time-consuming steps to set up Babel to convert JSX to compatible JavaScript that old browser can understand and configuring Webpack to bundle all the modules. React has been a popular framework for web development ever since its launch and no wonder it is incredibly powerful. React can render any kind of user interface you want. It acts like a vehicle for data, which can be any object (e.g., a text or an image).

Next we create a npm package for each component that we want to add to our app, inside the root directory of our project named components/[component name]/[component name].js. The Routes.js file is used to create routes and routes usually starts with the URL prefix, for

example, if we want to access /foo route then we need to first go through the main.js file where the module called react-router-dom will be included and then specify routes. This is how all the routes of an application will look like.

```
const Routes = () => {
    return (
        <BrowserRouter>
            <Switch>
                <Route path="/" exact component={Home} />
                <Route path="/shop" exact component={Shop} />
                <Route path="/signin" exact component={Signin} />
                <Route path="/signup" exact component={Signup} />
                <Route path="/product/:productId" exact component={Product} />
                <Route path="/cart" exact component={Cart} />
                <AdminRoute path="/admin/dashboard" exact component={AdminDashboard} />
                <PrivateRoute path="/user/dashboard" exact component={Dashboard} />
                <PrivateRoute path="/profile/:userId" exact component={Profile} />
            </Switch>
        </BrowserRouter>
    );
```

Figure 15Routung requests or pages

The front-end development team builds out each component and routes necessary to do so. As seen in figure 8, there are two routes that lead to the Profile component. The route '/shop' leads any type of user, but only authenticated users can see the Dashboard or Profile component. React Router provides three main components: Switch, Route, and Navigator. The Switch component is used to navigate between different routes. Route paths match the beginning of the URL, not the whole URL. Therefore a <Route path="/"> will match every URL. That is why <Route> should be located last in the <Switch>, otherwise <Route exact path="/"> can be used to match the whole URL. The Route component is used to match path parameters against a given URL and route it to the correct component behind it. It also provides wonderful support for URL history and loading hit testing along with supporting modern browsers like React Native and other native platforms as well. Finally, the Navigator component allows you to load content before your application has even started!

6.10.2. User Authentication

- Sign up Component

The Sign-up Component allows new users to sign up and log in to their account. This can be done through a specific form or through a dedicated link. If you do not require the user to sign up, there is a simple <form> element that can be used instead of this component.

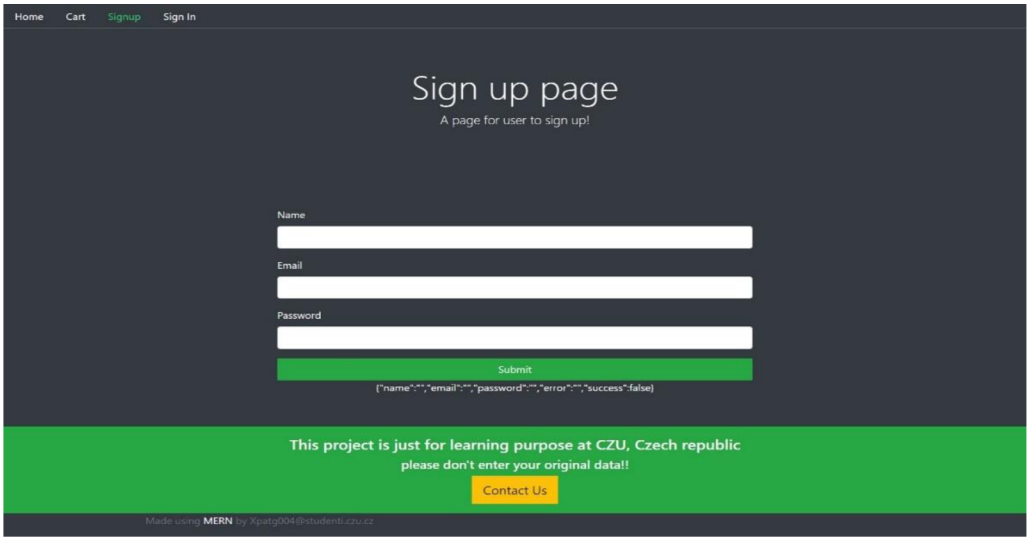


Figure 16Sign up Page

In figure 9 Sign up component is used to create a login/registration form. It renders a simple sign-up form with inputs for name, email and password. Users can select an option from the dropdown list and click submit button to proceed with registration. Create a sign-up component which has multiple inputs for name, email and password with a submit button. When you receive valid user data from the client, create a new instance of the user model based on req. body and save it into the database. Then return a JSON file containing user data without hashed password back to the client. Finally, show success alert up and redirect the user to the sign-in page.

- Sign in Component

  This is the sign in component. The same features as the sign up except the data flow when the user press submit button. After the server handled the POST request from the client, a token is signed and sent back to client along with information about how this user can be authenticated by using JWT (a new token type in Bootstrap 4).

```
export const authenticate = (data, next) => {
    //check if we have window obj
    if (typeof window !== undefined) {
        localStorage.setItem('jwt', JSON.stringify(data))
        next();
    }
};

export const isAuthenticated = () => {
    if (typeof window == undefined) {
        return false;
    }
    if (localStorage.getItem('jwt')) {
        return JSON.parse(localStorage.getItem('jwt'));
    } else {
        return false;
    }
};
```

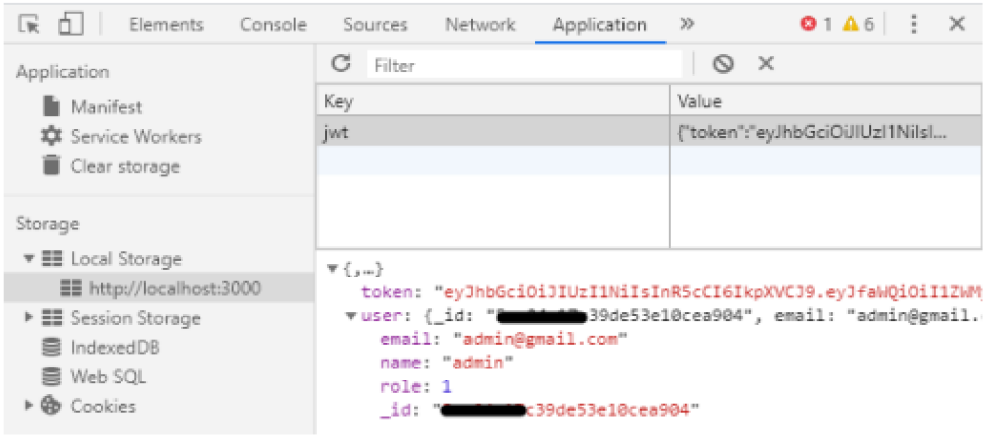


Figure 17User Authentication hash

Added by clicking Sign up Component, users can login and sign-up for your app. User data such as email address and password will be saved in local storage. When the user is signed in, information about his/her role can be retrieved from local storage using JSON.parse(). Once the user is signed up, he/she will automatically be signed in to all features of the dashboard. The login component is just like the logout component, except it sends back token and information about how this user can be authenticated by using JWT (a new token type in Bootstrap 4). Added by clicking Sign up Component, users can log in and sign-up for your app. User data such as email address and password will be saved in local storage. When the user is signed in, information about his/her role can be retrieved from local storage using JSON.parse()

6.10.3. Dashboard Component

- **Admin Dashboard.**

  Admin Dashboard provides a clean and organized way to get the job done. It has a powerful admin area with easy to use functionality around managing all of the product. Everything from managing your products, to setting up asset builds, to configuring single sign-on can be done right at your fingertips. It's also got an easy-to-use config flow so you can configure your system fast and get back to business. This component creates the management system for admin, as seen in the image below. It has two section components, both of which use the card feature of Bootstrap. Aside from specific admin operations like adding classes and goods, viewing, and controlling orders, all admin information is displayed on the left side of the screen. The admin will be directed to the appropriate component where he or she may establish a category or a product by clicking on the tasks link. The admin may also see and fulfil multiple orders, changing their status from pending to dispatched or delivered, for example. Additionally, the Manage categories component gives administrator the ability to edit or maybe even remove goods from the system.

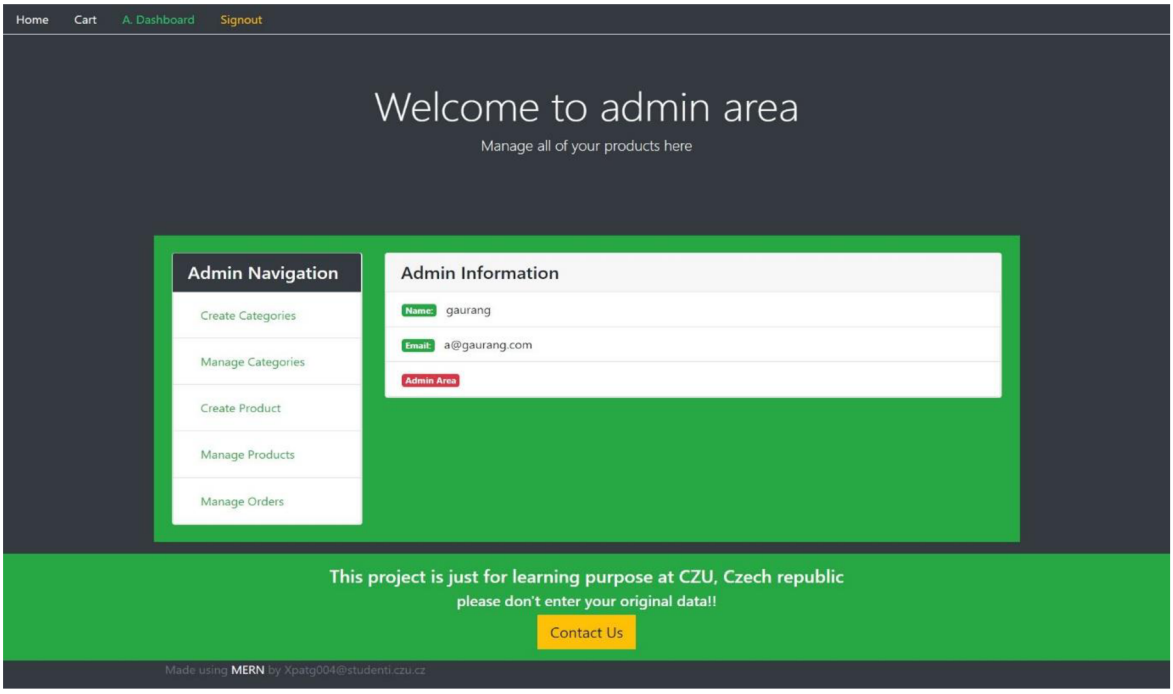
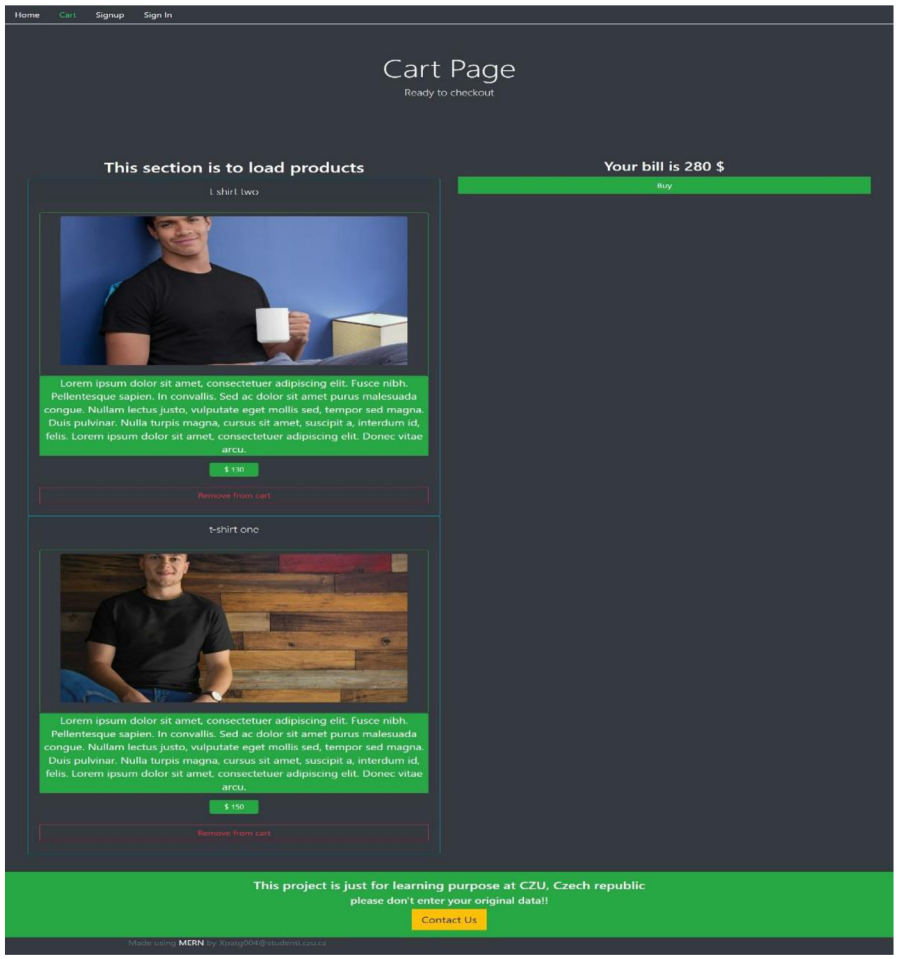


Figure 18Admin Area

Figure 12 is presented in the admin dashboard, which consists of 2 section elements - the left one displays administration actions like creating categories and products, viewing and managing orders. The right one contains additional information about the admin user like the custom fields that were assigned to each product or category. The Admin component has also a handy link that allows the admin to view orders and position them in various statuses - such as ordered, processing and shipped. ManageProduct component enables the admin user to modify products or even delete them from the database.This component comes with a functional pop-up window similar to that in Admin Panel.

6.10.4. Cart Component



The final page consumers must visit before making a purchase is the order page. A card component, which displays the item users wish to purchase, as well as a checkout element, which displays the total price, the delivery address, and a form to complete with payment details, are also included, in addition to certain common shared components. The Card component in the Cart differs somewhat from the Household or Store component's counterpart. It features a delete button for removing the item from the basket and an input area for users to change the final product quantity. The illustration makes it easy to see the cart component.

A product item will be added to the local storage when a customer taps the Add to cart button on a particular product. After that, the customer is sent to this Checkout page. To prevent the data from being lost every time the user resets the website, cart data is stored temporarily. Until it is deleted or the transaction is completed successfully, cart data is stored in storage device.

6.11. Payment integration

The payment gateway is the most important feature of any e-commerce framework. Braintree is among the most effective transaction handling options in the industry. If the business only accepts debit cards, it may lose a lot of clients because several people these days prefer to use PayPal. As a result, this web application incorporates both the bank card and PayPal systems.

Braintree is used by several of the world's most well-known companies, including Google, Github, and Dropbox. Braintree must be implanted separately in the node API (Sharma, 2021). The token can then be urged in the backend and delivered to the front-end. Confidentially, the.env database is created for the use of configuration files.

Once the identifier (token) has been submitted to the front-end Consumers can sign in to one‘s PayPal profile and charge for previous orders. Customers simply complete their purchase and return to the Home page after reimbursing for their order via PayPal or credit card.

6.11.1. PayPal integration

PayPal is perhaps the most widely used payment gateway for programmes that demand a transfer of funds. Payment controls are provided as a convenient and simple remedy for processing money in the PayPal transaction criterion.

A REST API app has indeed been generated in the PayPal Apps Control Panel after this webpage successfully linked its PayPal Business Account to the Braintree Dashboard. This app enables Braintree to communicate with the PayPal API; in the absence of this, the web application cannot operate PayPal transfers through the Braintree profile (38). It is strongly advised not to remove the REST API app at any moment.

6.11.2. Stripe integration

Stripe is a corporate fund transfer service provider and credit card processing platform. Whenever a consumer purchases a commodity available on the internet, the funds must be transferred to the vendor. Stripe facilitates the safe and efficient processing of funds via credit card or bank transfer to the company account.

After successfully connecting this website's Business Account with Stripe to the Braintree Dashboard, a REST API app was formed in the Stripe Applications Controlling Unit (48). This app enables Braintree to communicate directly with the Stripe API; if this is not present, the web service cannot proceed with the Stripe money transfers through the Braintree profile. It is strongly advised not to disable the REST API app anytime soon.

6.12. Discussion

The examination of the essential core components that make up the majority of the application, including MongoDB, Express.js, Node.js, and React.js—also referred to as the MERN stack—was the first step in developing the web service. The thesis investigated these 4 fundamental technologies, and even some of its inner mechanisms and operations, in addition to addressing the issues stated earlier by producing a full web service. A study of the technology, information on why it was selected, and a description of how it was employed in the development of the service were all included in this research that examined a core technology. During examination, it was discovered that each component of the MERN stack was suitable for the service's needs on an individual basis. Mern utlizies the REST API and The REST API makes it possible to integrate applications together like lego blocks. REST APIs are premised on Hypertext transfer protocol and mimic internet communication methods, rendering them ideal for usages in the MERN stack.

While contrasted to the MEAN Stack's Angular. js, the MERN Stack's React. js is superior for User experience stack extrapolation. Its accumulation of interactive user configurations easily accessible inside the library allows it to implement coding quickly. As the current MERN stack is created using three technologies: Node.js, Express Framework and MongoDB. These technologies were chosen as they provide a level of consistency throughout the MERN stack. Node.js and Express are JavaScript based on the Model View Controller (MVC) pattern and not only help us create applications but also allow us to rapidly prototype ideas quickly without having to develop them from scratch. Finally, MongoDB is a document database that can store millions of rows and millions of documents, which makes it an ideal database for building real time solutions that require strong data integrity.

In addition, the ability to utilize JavaScript as the programming language across all service levels, which substantially facilitated the development process, makes the MERN stack extremely consistent. Several customized scripts were used to transform the data from text documents or HTML node graphs into Json format until it was ultimately stored in the database. The user app and the API paths that it uses are served by a Node.js web server that also supports the Express.js technology. The React.js framework was used to develop the client application, which has distinct views for the various sorts of information involved as well as a search function for locating passages where specific phrases are referenced. Bootstrap and standard Cascading Style Sheet were used to style the application's visual appearance.

There are various technologies to select from while designing the application. However, we have opted to work with MERN stack. We utilized MERN as the tools to improve this as a web application. A few benefits of MERN are excellent efficiency, flexibility, integrated security, and accessibility. Although it is challenging to understand and put into practice, it generates less bugs due to code reuse and ensures development stability with higher security than applications that are manually programmed. To produce scalable and flexible applications, it is one of the most widely used industry standards for web development tools. The frontend and the backend are its two parts. Users may see products that are offered in-store, make purchases, and maintain their profile information from the front end. Frontend development is the part of a web site that allows users to view and access business objects, products and services. This includes the database, user interface, and associated dynamic pages. On the other

hand, backend development relates to databases and specific services that are required in order to use or create an application or website.

The system's backend framework maintains track of both orders placed and purchases that have been made. Additionally, the system maintains track of expenditures and purchases made for various items. It may determine the overall quantity of purchases and sales and also the most and least popular things. Thus, this will lessen the effort of the suppliers by enabling them to obtain an instant figure when doing a sales estimation of their company. A sales estimation solution is a must for any business involved in online sales. Providing a detailed statistic of total order quantity and the rate of consumption, this app helps clients to give larger discounts and offers specific promotions to increase conversion rates.

Customers have access to the interface. There are two types of user's guest and registered users, they have the option to view the products that have been put online from a system, but guests cannot place orders. One must sign up in a system if they want to place an order. On the other hand, registered users can place orders whenever they choose, and their products will be dispatched in just a few working days. Additionally, authorized users will have access to their accounts, credit records, and order-related notifications. With such a service accessible, clients do not have to wait in lines or deal with issues like product shortages after placing an order. In short, All users can access the interface through which they can access product information and view all items that have been put online. However, Registered Users are allowed to place orders as well, but guests cannot and one must register with a system if they want to place an order.

Since the suggested system does not include a payment gateway, payment can be made in cash when the requested products are received. According to a report, most internet shoppers are impulsive and often decide whether or not to stay on a website within the first few seconds. "Website design is similar to a store's decor. The client is likely to move on to another website if the store has a terrible appearance or is similar to hundreds of other stores.

The site is comprised of three main nodes: People, Products, and Browsing History. Each node contains numerous links to subnodes, and each subnode contains its own links to other user's inventory that can be viewed by those users whose inventory is located on the same subnode. As a result, we have made every effort in the project's design to make it as simple for users to navigate, retrieve data from, and offer the essential feedback. The user of this project is given access to an online shopping store through an e-commerce website.

The project website has designed to provide an interactive front end that enables the user to navigate through a series of screens and sub-pages. These pages are used to showcase products, provide more detail about those products, and allow customers to add items for purchase. The pages are also used for offering various forms of messaging such as confirmations and order status messages. Users can enhance their shopping experience by accessing a secure online account for placing orders and adding notes about individual items purchased.

7. Conclusion

MERN stack is the best place for web applications with a large number of pre-built connections. It's ideal for developing web applications with high interaction rates and long lifetimes, and in cases, wherein many services must be integrated into a single application. Although other stacks are also good to have, MERN is suitable for large web applications where it may feel the same way with other stacks but will find that it is much easier to do so with MERN. In React, the method of systematically producing HTML (actually DOM objects) uses JavaScript. The MERN stack is an open-source, scalable, and highly interconnected technology that is being used to improve the structure of modern e-commerce applications.

The thesis performed an in-depth analysis of the fundamental principles underpinning each technology within the MERN stack and then implemented them as part of a fully functioning application for an online shop. The thesis was able to demonstrate the various features of the MERN stack, including server-side rendering, and testing. In addition, it also demonstrated the ability to construct complicated full-stack applications such as a shopping cart application. However, it did not perform as well regarding responsiveness design for various devices and third-party functionality features.

8. Bibliography

1. **Mehra, M., Kumar, M. Maurya, A. and Sharma.** *MERN Stack eb development. annals of Romanian society for cell biology.* 2021.

2. **Appelt, Š.** *Website Development as a Credence Good: A Field Experiment.* 2020.

3. **HUSSIAN.** *A Case Study: Introduction to E-Commerce.* 2020.

4. *E-Commerce: advantages and limitations. International Journal of Academic Research in Accounting Finance and Management Sciences,.* **Taher, G.** 2021, pp. 153-165.

5. **Heinemann, G.** Grundlagen, Geschäftsmodelle und Best Practices im Business-to-Business Online-Handel, Wiesbaden, Springer Gabler. 2020, Vol. B2B eCommerce Vol. 31.

6. *Innovative solutions to increase last-mile delivery efficiency in B2C e-commerce: a literature review. International Journal of Physical Distribution & Logistics Management.* **Mangiaracina, R., et al.** 2019.

7. *Trust in C2C electronic commerce: Ten years later. Journal of Computer Information Systems,.* **Leonard, L.N and Jones, K.** 2021, pp. 240-246.

8. *International Conference on E-Commerce and Internet Technology (ECIT).* **Wu, Q, Ma, J and Wu, Z.** 2020. pp. 206-221.

9. *THE NATURE AND THE SCOPE OF THE MODERN E-COMMERCE. International Independent Scientific Journal,.* **Anev, G.** 2021, pp. 13-16.

10. *Protection Of Private Data Consumers P2P Lending As Part Of E-Commerce Business In Indonesia.* **Winarso, T, Disemadi, H.S and Prananingtyas, P.** 2020, pp. 206-221.

11. *Impacts of e-commerce on planning and designing commercial activities centers: A developed approach. Ain Shams Engineering Journal,.* **Mohamad, A.H., Hassan, G.F and Abd Elrahman, A.S.** 2022, Vol. P.101634.

12. *Implementation Of E-Business Information System In Indonesia: Prospects And Challenges. International Journal of Cyber and IT Service Management.* **Setyowati, W., Widayanti, R and Supriyanti, D.** 2010, pp. 180-188.

13. *E-business strategy in developing countries: A framework and checklist for the small business sector. Sustainability.* **Wynn, M and Olayinka, O.** 2021, Vol. p. 7356.

14. *Overview of the e-Business strategy framework. In Strategies for e-Business.* **Jelassi, T and Martínez-López, F.J.** 2020, pp. 35-48.

15. *E-Business Strategy for Logistics Companies: Achieving Success through Information Systems Planning. Logistics,.* **Kamariotou, M, Kitsios, F and Madas, M.** 2021, p. 73.

16. **Brickley, D., Burgess, M and Noy, N.** Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In The World Wide Web Conference. 2019, pp. pp. 1365-1375.

17. *SHOPIFY ONLINE E-COMMERCE WEBSITE.* **Narwaria, M., Tiwari, A and Singh, A.**

18. *Analisa Perbandingan Performa Website Multi-Page Application dan Single-Page Application pada E-Commerce Mentari (Doctoral dissertation, Universitas Muhammadiyah Malang).* **Prakoso, Y.R.N.** 2021.

19. *Multi-Website Single-Repository Architecture for E-Journal Web Platform. In 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology.* **Aleksandrov, M and Petrova, M.** 2021. pp. 38-42.

20. *Pro MERN Stack. Apress.* **Subramanian, V.,.** 2017.

21. *Improving web development process of MERN stack.* **Nguyen, B.** 2021.

22. *Developing a social platform based on MERN stack.* **Tran, H.** 2021.

23. *Encapsulation and Polymorphism. In Learn Java with Math.* **Dai, R.** Berkeley CA : s.n., 2019, pp. 181-183.

24. *Postponing the Concept of Class When Introducing OOP.* **Passerini, N and Lombardi, C.** 2020. pp. 152-158.

25. *SmilesDrawer: parsing and drawing SMILES-encoded molecular structures using client-side JavaScript.* **Probst, D and Reymond, J.L.** 2018. pp. 1-7.

26. *A server-side javascript security architecture for secure integration of third-party libraries. Security and Communication Networks,.* **van Ginkel, N, et al.** 2019.

27. **(n.d.)., IJRASET.** https://www.ijraset.com/research-paper/mern-full-stack-development. *www.ijraset.com.* [Online] 2020.

28. **Male wade, S. (n.d.).** https://www.ijert.org/research/performance-optimization-using-mern-stack-on-web-application-IJERTV10IS060239.pdf. *https://www.ijert.org/.* [Online] 8 2019.

29. *Comparative analysis of MEAN stack and MERN stack. International Journal of Recent Research Aspects,.* **Aggarwal, S and Verma, J.** 2018, pp. pp 127-32.

30. **Chhetri, N.** A comparative analysis of node. js (server-side javascript). 2016.

31. **Kapoor, A.** https://enlear.academy/benefits-of-using-mern-stack-7e0c732b5214. *https://enlear.academy.* [Online] 2021.

32. **Porter, P., Yang, S. and Xi, X.** *The Design and Implementation of a RESTful IoT Service Using the MERN Stack.* 2019. pp. 140-145.

33. **SearchDataManagement.** https://www.techtarget.com/searchdatamanagement/definition/MongoDB. *https://www.techtarget.com.* [Online]

34. *Real-Time BPMN Website Menggunakan Teknologi MERN Stack.* **Handoyo, R., Santoso, L.W. and Setiawan, A.** 2019, pp. 75-80.

35. **Johnson, S. (n.d.).** https://www.diva-portal.org/smash/get/diva2:1416891/FULLTEXT01.pdf. *https://www.diva-portal.org/.* [Online]

36. *MERN Stack Web Application. Annals of the Romanian Society for Cell Biology,.* **Raju, S., Soundararajan, S and Loganathan, V.** June 10, 2017, A Study of Software Development Life Cycle Process Models.

37. **Le, P.** *Development of an eCommerce website for Ngoc's MaxiNutri Company.* 2022.

38. **Nguyen, C.C.** *A social platform using modern web stack (MERN).* 2021.

39. *Food Ordering website "Cooked with care" developed using MERN stack.* **Chauhan, A.S, et al.** 2022, pp. pp. 1690-1695.

40. *College Query Management System by using MERN Stack.* **Kalwani, B., Sharma, A and Gupta, S.L.** 2020, pp. pp.17-24.

41. *The development of an e-commerce web application using MERN stack.* **Tran, T.T.H.** 2022.

42. *College Website Using MERN Stack.* **Patil, R., et al.** 2022.

43. *A Sense of Time for {JavaScript} and Node. js:{First-Class} Timeouts as a Cure for Event Handler Poisoning.* **Davis, J.C, Williamson, E.R and Lee, D.** 2018, pp. pp. 343-359.

44. *Mininode: Reducing the attack surface of node. js applications.* **Koishybayev, I and Kapravelos, A.** 2020, pp. pp. 121-134.

45. *Performance comparison for data retrieval from nosql and sql databases: a case study for covid-19 genome sequence dataset.* **Chakraborty, S, Paul, S and Hasan, K.A.** 2021.

46. **Gauthier, J., et al.** *A brief history of bioinformatics. Briefings in bioinformatics.* 2019.

47. **Bhatnagar, Vivek.** *A comprative study of sdlc model. IJAIEM.* 2015. pp. 23-29.

48. **Sharma, A.K.** *BIG BUY (E-COMMERCE) by using MERN.* 2022.

49. **A, Chen.** *Design Applications in Website Development.* 2020.

50. **S.N.S, Hassan, et al.** *Managing Website Development for Darul Hana Mosque in Sarawak by implementing ADDIE Approach.* s.l. : UMRAN International Journal of Islamic & Civilizational Studies., 2021. pp. 23-37.

51. **Wangmo, J.,, et al.** *Feasibility Study of E-Commerce Website Development for the Cooperative Store at College of Science and Technology.* 2018. pp. 1-6.

52. **Wangmo, J., et al.** *Report on the Feasibility Study of E-Commerce Website Development for the Cooperative Store at College of Science and Technology.* 2018. pp. 1-6.