

# Thrift Shop Using MERN Stack

## Project Report



Submitted by:

Yashvardhan  
Bachelor's of  
Technology  
Computer Science  
and Engineering  
181500833

Taran Singh  
Bachelor's of  
Technology  
Computer Science  
and Engineering  
181500755

# Contents

<i>List of Abbreviations</i>	<i>1</i>
<i>1 Introduction</i>	<i>2</i>
<i>2 E-commerce</i>	<i>3-5</i>
<i>2.1 Definition</i>	
<i>2.2 Types</i>	
<i>2.3 Advantages</i>	
<i>2.4 Challenges</i>	
<i>3 MERN Stack</i>	<i>6-13</i>
<i>3.1 JavaScript</i>	
<i>3.2 NodeJS</i>	
<i>3.3 Express.js</i>	
<i>3.4 MongoDB</i>	
<i>3.5 ReactJS</i>	
<i>3.5.1 Virtual-DOM</i>	
<i>3.5.2 Component</i>	
<i>3.5.3 Props and State</i>	
<i>3.5.4 Pros and Cons of ReactJS</i>	
<i>3.6 MERN Stack in Website Development</i>	
<i>3.6.1 Concept of Stack technology</i>	
<i>3.6.2 Concept of MERN Stack</i>	
<i>3.6.3 Highlights in MERN Stack</i>	
<i>4 Thrift Shop E-commerce Web Application</i>	<i>14-19</i>
<i>4.1 Home Page</i>	
<i>4.2 Login System</i>	
<i>4.2.1 Sign Up</i>	
<i>4.2.2 Sign In</i>	
<i>4.3 User Dashboard</i>	
<i>4.4 Products Details Page</i>	
<i>4.5 Cart</i>	
<i>4.5.1 Selecting Payment Option</i>	
<i>4.5.2 Payment Gateway</i>	
<i>4.5.3 Order Progress</i>	
<i>5 Responsiveness of Web Application</i>	<i>20</i>
<i>6 Summary</i>	<i>21</i>
<i>References</i>	<i>22-23</i>

## **List of Abbreviations**

- DOM Document Object Model
- W3C World Wide Web Consortium.
- ISO International Organization for Standardization
- NPM Node Package Manager
- API Application Program Interface
- HTML Hypertext Markup Language
- HTTP Hypertext Transfer Protocol
- URL Uniform Resource Locator
- JSON JavaScript Object Notation
- JS JavaScript
- JSX JavaScript XML
- XML Extensive Markup Language

# 1 Introduction

It is true that technology has become an essential tool for online marketing nowadays. However, there are numerous small shops and grocery stores with mostly offline business model in Vietnam recently. With this commerce model, it will bring a lot of bad experiences for both buyers and sellers.

For instance, the seller has the product want to offer but the buyer may not know it, or the buyer may urgently need to purchase something, but the store is out of stock. Moreover, online shopping helps customers to choose a wide range of products, prices and they can compare them to each other easily.

Encountering the inadequacies and the weaknesses of the offline business model, making a website application for searching and buying things for each shop is very necessary right now. Recently, there have been many e-commerce sites exported such as Amazon, e-bay or the stores that can sell products via social media channels like Facebook.

However, customers still find it difficult to choose the products they want because of the large variety of products on these sites and not focus on specific things. Moreover, the sellers have to spend a high amount of money on marketing or paying for fees. From there disadvantages, implement an online e-commerce web application for small grocery stores helps retailers can manage products on their own systems and not depend on the 3rd party website. For the customers, they can quickly search the products if it is available and come to store to pick it up and they can contact directly to the shop owner to learn more about the products that they are looking for.

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

## 2 E-commerce

### 2.1 Definition

E-comm, EC for short (E-commerce) is a concept referring to transactions, purchase and sale of goods and services by the internet.

E-commerce was first known in the 1960s. After years of development, as mobile devices became popular, social media increasingly affirmed the power and the boom of the web page. Launchers promote the rapid development of commerce (E-commerce).[1]

### 2.2 Types

Currently, there are many forms of e-commerce, including the following basic forms:

**B2B (Business to Business):** is a trade between companies, businesses and organizations. About 80% of e-commerce today falls into this category.

**B2C (Business to Consumer):** is an Internet-based business to directly exchange the goods and services it creates or distributes to consumers.

**C2B (Consumer to Business):** is a consumer who sells their products or services to a business or organization.

**C2C (Consumer to Consumer):** is when a consumer sells his goods or services to another consumer.

There are also G2C, G2B, etc., but used less often than these four basic forms.

## 2.3 Advantages

- **Global market:** Clearly, when you open a physical store, you will only be able to deliver your goods and services in a small geographic area. E-commerce will help you solve that problem. E-commerce helps you reach the market quickly, expanding the market to the maximum level compared to direct sales, so that products and services are easily introduced, purchased and sold through retailers. and online market.
- **Always open:** In e-commerce, running an online business is much easier, it's always open 24h / 7/365. For businesses, it's a great opportunity to increase sales opportunities all the time.
- **Budget savings:** Compared with traditional forms of commercial business, all costs when e-commerce business are reduced: the cost of renting booths, salespeople and management is much more economical. . Naturally, when sellers save operating costs, they can offer more incentives and better discounts for their customers. At this time, the customer is the next beneficiary. Mutual benefit, isn't it great?
- **Inventory management:** By using electronic tools to speed up the ordering, delivery, and payment processes, e-commerce businesses can save billions of operating costs and reduce amount of inventory.
- **Most accurate customer marketing:** With access to customer data and the opportunity to track customers' buying habits, e-commerce businesses can quickly identify and market products and services. service. Service most suitable for consumers.
- **Work anywhere, buy anywhere:** Running an e-commerce business allows you to not need to sit in the office, and buying does not force you to go to the supermarket. Everything the seller and the buyer needs is an internet-connected device and that's all.[2]

## 2.4 Challenges

**Internet access required:** When participating in the EC, to be able to buy and sell, you need a device connected to the internet. Currently, most people have internet access but, in many areas, it is still very limited.

**Not enough to trust:** Products and services that cannot be seen, touched, held or felt directly, are not allowed to try as a prudent buyer. Doubt in both buyers and sellers leads to many incomplete transactions, especially when they have dealt with untrusted partners before.

**Limited payment methods:** Currently, the most popular payment method in Vietnam when buying goods online is to receive and pay. Payment gateway in Vietnam is growing quite strong, but not reliable enough for users to use as the main payment method. Therefore, it also contributes to teething.

In addition, e-commerce business also faces many other challenges: technical, competitors, payment, etc.

## 3 MERN Stack

### 3.1 JavaScript

JavaScript is a scripting, object-oriented, cross-platform programming language. Objects of host environment can be connected to JavaScript and arrange ways to operate them.

Standard libraries for objects are contained by JavaScript, for such as Array, Date, Math, and the essence component of programming languages for instance managers, control framework and statements.

By adding objects, JavaScript could be protracted for many principles, such as:

- **Client-side JavaScript:** JavaScript is developed by implementing objects for controlling the browser and DOM. For instance, an application is granted by client-side extensions to influence components on an HTML page and answer to user behavior like mouse hovers, form input and page changeover.
- **Server-side JavaScript:** JavaScript is developed by implementing the supplementary objects required to run JavaScript on the server. For instance, an application is granted by this server-side extension to connect to a database, transfer data frequently from one request to other section of the application or execute application with another function file on the server.

In 1996, JavaScript was officially named ECMAScript. ECMAScript 2 was released in 1998 and ECMAScript 3 was released in 1999. It is continuously evolving into today's JavaScript, now works on all browsers and devices from mobile to desktop. Open standard language can be used by association to establish their own JavaScript applications. The ECMAScript Standard is one of the parts of the ECMA-262 specification.

ISO has approved the ECMA-262 standard at ISO-16262. The ECMAScript standard does not include descriptions for the DOM, it is standardized by the W3C.

The DOM specifies how your scripts display HTML objects. To get a advance anticipate of the distinctive innovations used when programming with JavaScript, check out the JavaScript technology analysis article. [3]



## 3.2 NodeJS

Node.js is an open source, a system application and furthermore is an environment for servers. Node.js is an independent development platform built on Chrome's JavaScript Run time that we can build network applications quickly and easily. Google V8 JavaScript engine is used by Node.js to execute code. Moreover, a huge proportion of essential modules are written in JavaScript Node.js accommodate a built-in library which allows applications to serve as a Web- server left out demanding software like Apache HTTP Server, Nginx or IIS.

An event-driven, non-blocking I / O mechanisms (Input / Output) are implemented by Node.js. It optimizes application throughout and is extremely high extensible. Node.js use asynchronous in it functions. Therefore, Node.js processes and executes all tasks in the background (background processing).

Products that have a lot of traffic are applying Node.js. Nonetheless, Node.js handle the application that need to spread expeditiously, develop innovation, or build Startup projects as rapidly as possible.[4]

### **Applications using NodeJS:**

- Web Socket server
- Notification system
- Applications that need to upload files on the client.
- Other real-time data applications.

### **NodeJS Pros:**

- Node.js is the exclusive application that with only a single thread, it can obtain and handle numerous connections. Building new threads for each query is not needed, therefore the structure expends the least amount of RAM and run rapidly. Secondly, Node.js produces the most of server property without generate latency with the JavaScript's non-blocking I/O.
- **JSON APIs.** JSON Web services can take advantages of that because of the event-driven, non-blocking I/O structures and JavaScript-enabled model.
- **Single page application.** NodeJS is very suitable with an application on a single page. Node.js has the capability to handle different requests concurrent and quick return. Node JS should be used in an application that does not have to reload the page, including users who makes a vast number of requests and need a quick procedure to show professionalism.

- **Shelling tools Unix.** Node.js usually uses Unix to work. They can handle multiple processes and return them for best performance. Programmers often use Node.js to build real Web applications like chat, feeds, etc.
- **Streaming Data.** Typical websites send HTTP requests and also receive responses. Node.js can handle many questions and feedback, so they are suitable if the developer wants to create an application on the page. In addition, Node.js also builds proxies to stream the data, this is to ensure maximum operation for other data streams.
- **Real-time Web Application.** Node.js is sufficient to develop real-time innovations like chat apps, social networking services like Facebook, Twitter because of the opening of mobile application.

#### **NodeJS Cons:**

- Resource-intensive applications. Node.js is written in C ++ & JavaScript, so when programmers need to handle applications that use a lot of file conversion, video encoding, decoding, etc., they should not be used Node.js. Programmers need to use it more carefully in this case.
- The final purpose of NodeJS is like other programming languages such as Ruby, PHP, .NET, Python, that is developing web application. Therefore, do not expect NodeJS to outperform other language for now. But with NodeJS the application can be developed successfully as expected. [5]

#### **NodeJS should not be used when:**

- Build resource-intensive applications: Do not use Node.js when creating a video converter application. Node.js often comes down to bottlenecks when working with large files.
- An all-CRUD-only application: Node.js is not faster than PHP when doing heavy I/O tasks. In addition, with the long-term stability of other webserver scripts, its CRUD tasks have been optimized. Node.js will come up with odd APIs and never be used.
- Stability in the application: Within 11 years of development (2009-2020), the current version of Node.js is already v14.2.0. Every API can be changed in a way that is not backwards compatible.
- Lack of knowledge about Node.js: Node.js is extremely dangerous in this case, you will fall into a world full of difficulties. With most non-blocking/async APIs, not understanding the problem will cause an error that you do not even know

where it came from. Moreover, when the Node.js community is not strong enough, and there will be less support from the community.

#### **NodeJS should be used when:**

- **Building RESTful API (JSON).** You can use Node.js in building RESTful API (JSON). They handle JSON very easily, even more than JavaScript. API servers when using Node.js usually do not have to perform heavy processing, but the number of concurrent requests is high.
- **Applications that demand alternative connection protocols,** not just http. With TCP protocol backing, any custom protocol can be built easily.
- **Real-time applications.**
- **Stateful websites.** Every request on the invariable procedure is handled by Node.js, therefore building caching is simpler: store it to a comprehensive variable then all requests can approach the cache. The status of one client can be saved and shared with other clients and do not have to go through external memory. [6]

### **3.3 Express.js**

Express.js is a framework built on top of Node.js. It provides powerful features for web or mobile development. Express.js supports HTTP and middle-ware methods, making the API extremely powerful and easy to use.

Express implements extra features to developer which help them get a better programming environment, not scaling down the speed of NodeJS.

Importantly, the well-known frameworks of NodeJS apply Express.js as a substance function, for instance: Sails.js, MEAN.[7]

### **3.4 MongoDB**

MongoDB is an open source database; it is also the leading NoSQL (\*) database currently used by millions of people. It is written in one of the most popular programming languages today. In addition, MongoDB is cross-platform data that operates on the concepts of Collections and Documents, providing high performance with high availability and ease of expansion.[8]

(\*) NoSQL is a source database format that does not use Transact-SQL to access information, this database was developed on JavaScript Framework on JSON data type. With its introduction, it has overcome the disadvantages of RDBMS relational data model to improve operating speed, functionality, model scalability, cache ... Furthermore, MongoDB is a cross-platform database, performing on Collection and

Document approach, it produces sharp production, huge availability, and effortless scalability.

### Commonly used terms in MongoDB:

- **\_id:** Almost every document required this field. The `_id` field illustrates a exceptional value in the MongoDB document. The `_id` field can also be interpreted as the primary key in the document. If you add a new document, MongoDB will automatically generate a `_id` representing that document and be unique in the MongoDB database.
- **Collection:** A group of many documents in MongoDB. Collection can be interpreted as a corresponding table in the RDBMS (Relational Database Management System) database. Collection resides in a single database. Collections do not have to define columns, rows or data types first.
- **Cursor:** This is a pointer to the outcome set of a query. The client can emphasize over a cursor to get the result.
- **Database:** The location of the collections, similar to the RDMS database that contains the tables. Each Database has a separate file stored on physical memory. Some MongoDB owners may contain various databases.
- **Document:** A transcript belonging to a Collection. Documents, in turn, include name and value fields.
- **Field:** A name-value pair in a document. A document may not need all the fields. The fields are like columns in a relational database.
- **JSON:** Short for JavaScript Object Notation. Human readability is in the plain text format representing structured data. JSON currently supports a lot of program- ming languages.
- **Index:** Exclusive data structures used to save a small allocation of data sets for simple scanning. The index puts the value of a individual field or sets of fields, sorted by the value of these fields. Index effectively supports the analysis of queries. Without an index, MongoDB will have to scan all the documents of the set to choose the documents that pair the query. This scan is ineffective and requires MongoDB to progress a vast amount of data.

MongoDB Atlas is MongoDB's cloud database launched in 2016 on AWS, Microsoft Azure and Google Cloud Platform.

The data in each Cluster in the Atlas is stored by Replication mechanism, with 3 nodes: 1 master (primary) and 2 slaves (secondary).

## 3.5 ReactJS

### 3.5.1 Virtual-DOM

Virtual-DOM is a JavaScript object, each object contains all the information needed to create a DOM, when the data changes it will calculate the change between the object and the real tree, which will help optimize re-render DOM tree. It can be assumed that is a virtual model can handle client data.[9]

### 3.5.2 Component

React is built around components, not templates like other frameworks. A component can be created by the create Class function of the React object, the starting point when accessing this library.

ReactJS creates HTML tags unlike we normally write but uses Component to wrap HTML tags into stratified objects to render.

Among React Components, render function is the most important. It is a function that handles the generation of HTML tags as well as a demonstration of the ability to process via Virtual-DOM. Any changes of data at any time will be processed and updated immediately by Virtual-DOM.[10]

### 3.5.3 Props and State

**Props:** are not controlled by Component, actually stands for Properties.

The title = "Title" line creates a name attribute with the value "Title". It looks like a function call. It is true that props are passed to the component in the same way that an argument is passed to a function.

A component may also have a default props, so if the component does not pass any props, it will still be set.

**State:** private data is controlled by Component.

Like props, state also holds information about the component. However, the type of information and how to handle it varies. State works differently than Props. The state is a component of the component, while props are passed in from the outside into the component. It should be noted that we should not update the state directly using this.state but always use setState to update. Update the state of the objects. Use setState to re-renders one component and all its children. This is great, because you don't have to worry about writing event handlers like any other language.[11]

### **3.5.4 Pros and Cons of ReactJS**

#### **Pros of ReactJS:**

- Update data changes quickly.
- React is not a framework so it offloads the constraints of libraries together.
- Easy access to who understands JS.

#### **Cons of ReactJS:**

- ReactJS only serves the View tier, but the library size is on par with Angular while Angular is a complete framework.
- Incorporating ReactJS within common MVC frameworks demands reconfiguration.
- Hard to reach for beginners on website developing.[12]

## **3.6 MERN Stack in Website Development**

### **3.6.1 Concept of Stack technology**

The technical stack is a combination of technologies / frameworks / programming languages, etc. to create a complete software.

With current software, there are usually two parts: client side and server side, also known as frontend and backend. Therefore, people also split the backend stack, the frontend stack as well. We often use the first letter to name the technical stack: LAMP (Linux, Apache, MySQL, PHP), MEAN (MongoDB, Express, Angular, NodeJS).[13]

### **3.6.2 Concept of MERN Stack**

The MERN stack is a complete open source combination, all JavaScript-related technologies are also the hottest today: MongoDB, Express.js, React / React Native,

NodeJS. People use the MERN stack to build Universal React App.[14]

### **3.6.3 Highlights in MERN Stack**

- Hot Reloading for React Components.
- The code snippets are divided by React Router.
- Multi-language support.
- Generate code support.
- Modular structure.
- Docker support.
- Can customize the MERN version for individual users. [8]

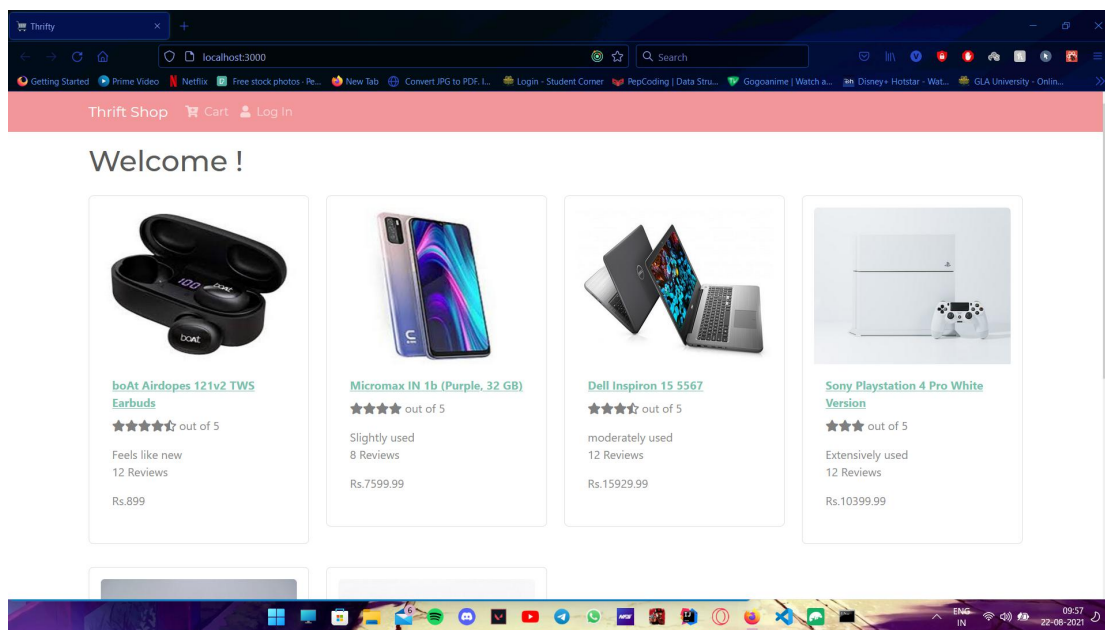
## 4 Thrift Shop E-commerce Web Application

Thrift Store is an E-commerce Web Application using MERN stack that can help a toys retailer bring their products to the customers.

### Main function:

- **Sign up and log in:** Requires Users to register using their phone number or email
- **Shopping cart:** this feature helps users buy and check goods directly on the application
- **Search:** Users can search directly by typing in the search box for the product they want to see.
- **Buy and pay:** Customers who buy through the app can pay through many different payment gateways.

### 4.1 Home Page



*Figure 1. Main Home Page*

On the above figure we can see the home page of the app. It includes the indispensable components of an online store like the function bar and a list of products. Customers who visit the store can search for the products they want to buy, they can view ratings and other details of the products.

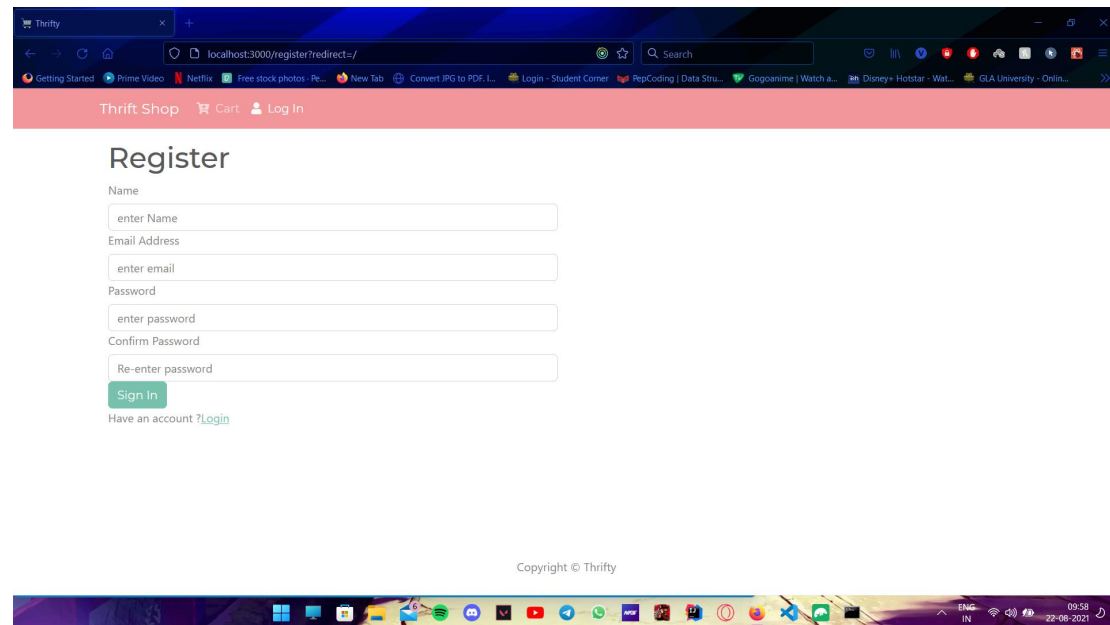
The Home page section lists all the products that are recently added to the database in MongoDB. As it can be seen at figure 1, there is a Cart section in the function bar.



## 4.2 Login System

The login system of the app will ask the user to provide the Name, email and password for registration. We can use these details for login function in future.

### 4.2.1 Sign Up

A screenshot of a web browser displaying the 'Register' page of a website called 'Thrifty'. The browser's address bar shows 'localhost:3000/register?redirect=/' and the page has a pink header with 'Thrifty Shop', a shopping cart icon, and a 'Log In' link. The main content area is titled 'Register' and contains five input fields: 'Name', 'Email Address', 'Password', 'Confirm Password', and 'Re-enter password'. Each field has a placeholder text like 'enter Name'. Below the fields is a green 'Sign In' button and a link that says 'Have an account ?Login'. At the bottom of the page, there is a copyright notice 'Copyright © Thrifty'.

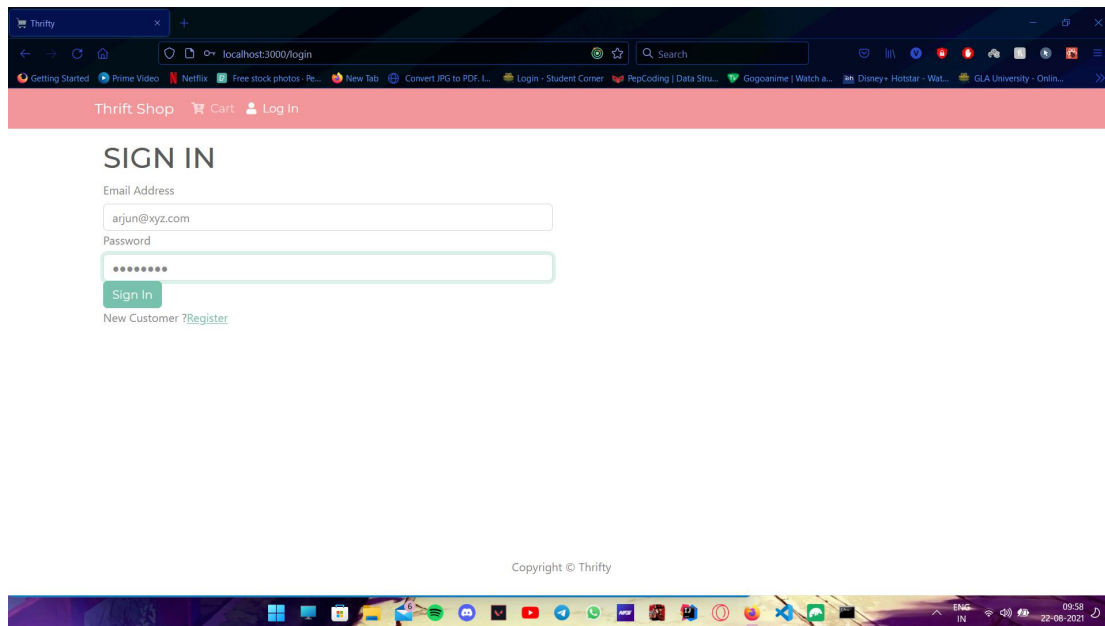
*Figure 2. Sign -Up screen for the users.*

Users login into the website need to sign in otherwise they have to register a new account. In figure 2, the sign-up screen only has 4 fields that require users to fill in is Name, Email, Password and Confirm Password.

All the details is going to be stored in database and can be used for future confirmation.

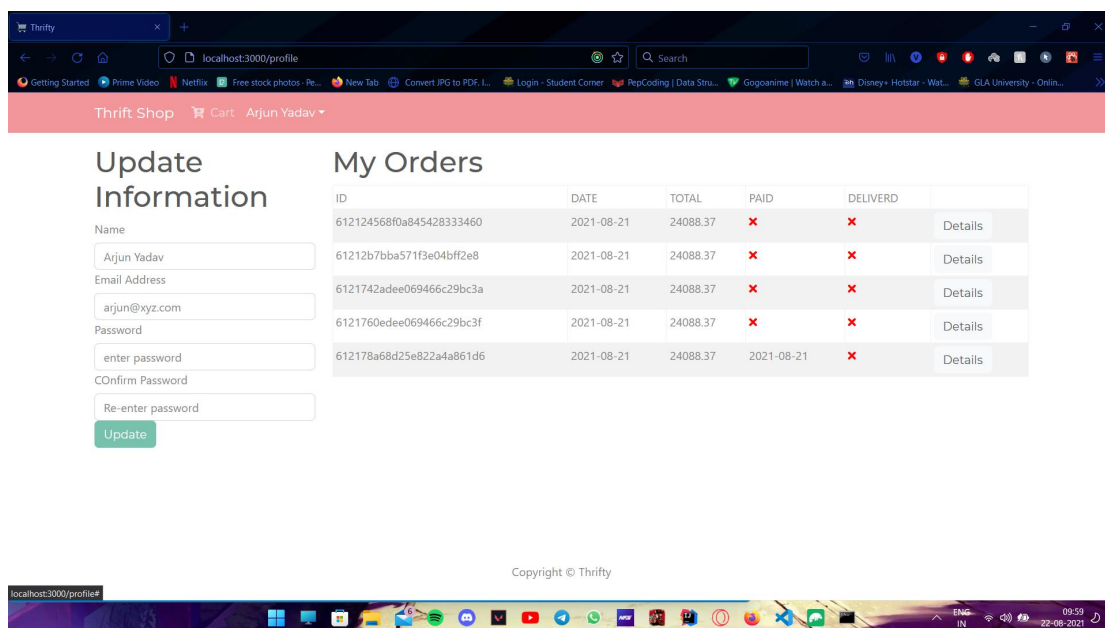
### 4.2.2 Sign in

The Sign In is similar to the Sign-Up screen, only the email and password field are display. After the sign in is successful, the user will be redirected to the Home page.



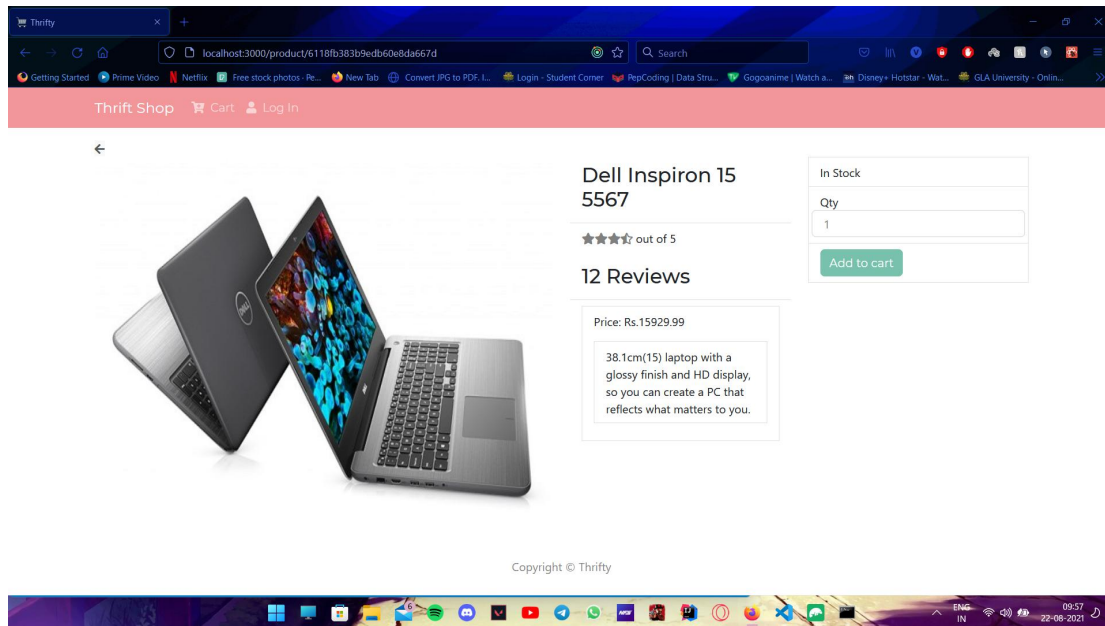
After typing email and password, these strings need to be read and authenticated, if there is no error then the login session is successful.

## 4.3 User Dashboard



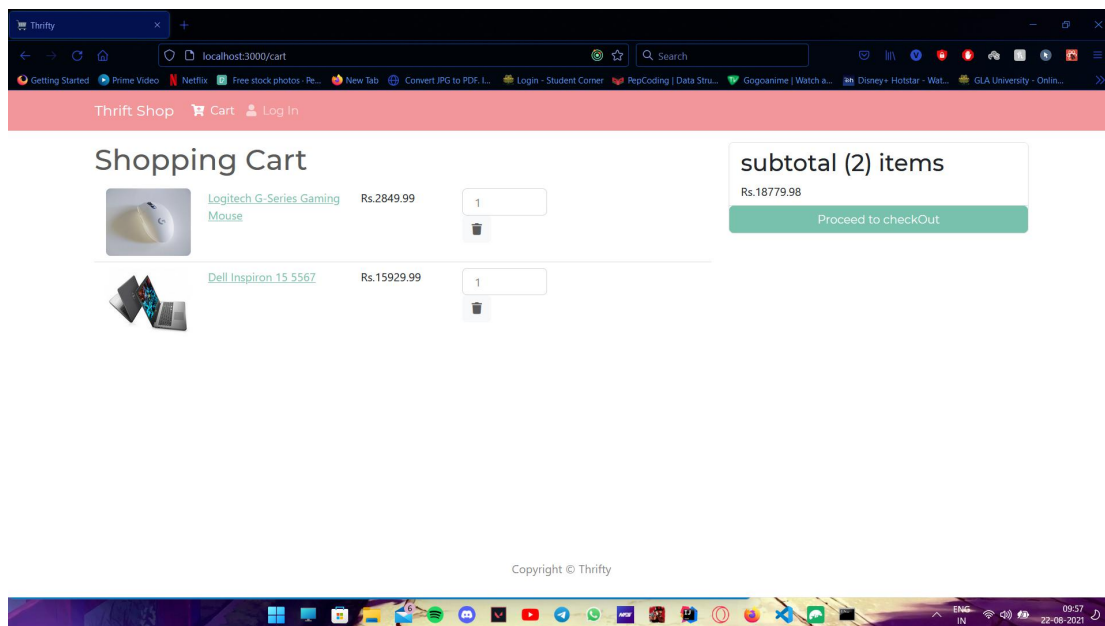
On this page user can see the complete order details and a brief summary like unique order id, date, total price incl. Of taxes, payment and a details button which give full description.

## 4.4 Product Details Page



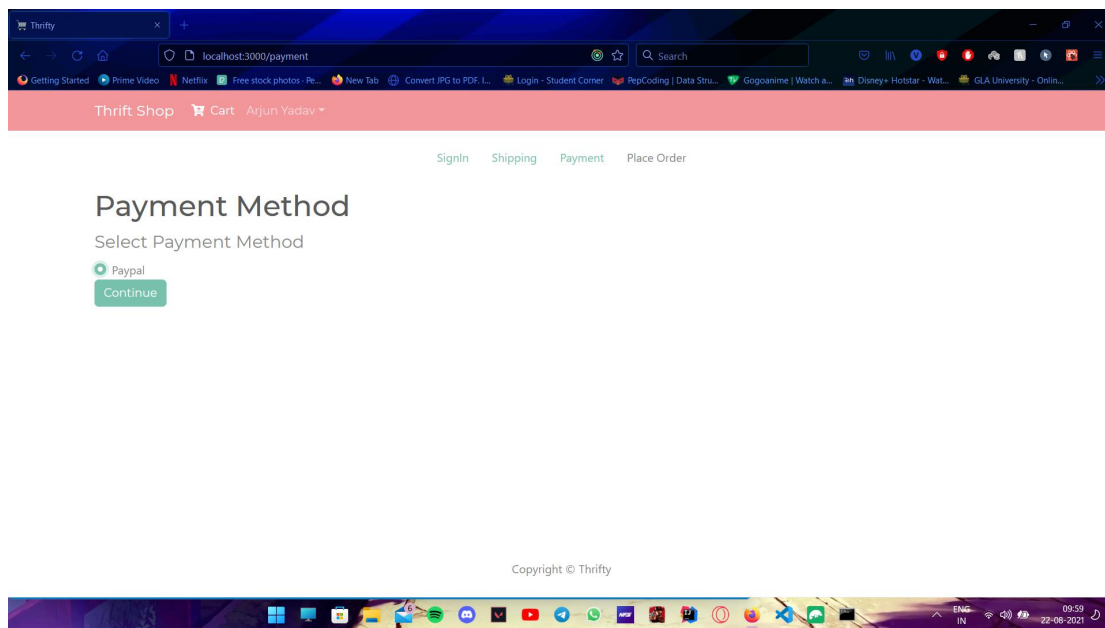
On this page there are different details related to product like rating, total number of reviews, shorthand description of product. We can select the quantity of product and add it to the cart.

## 4.5 Cart

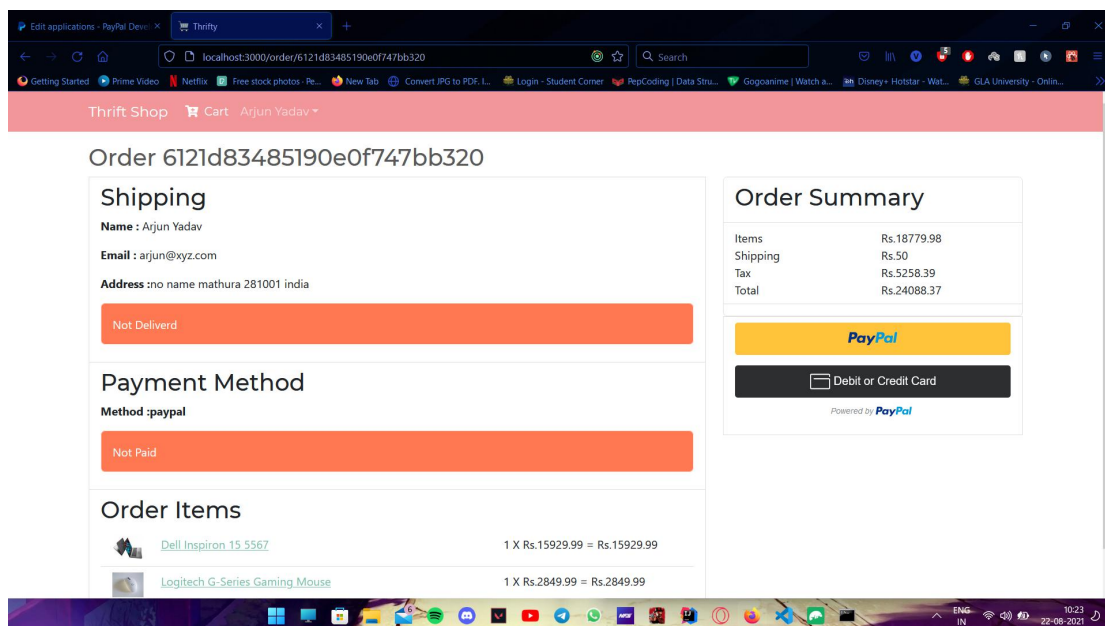


This is a checkout page for a final confirmation of your order and after that you can proceed to checkout.

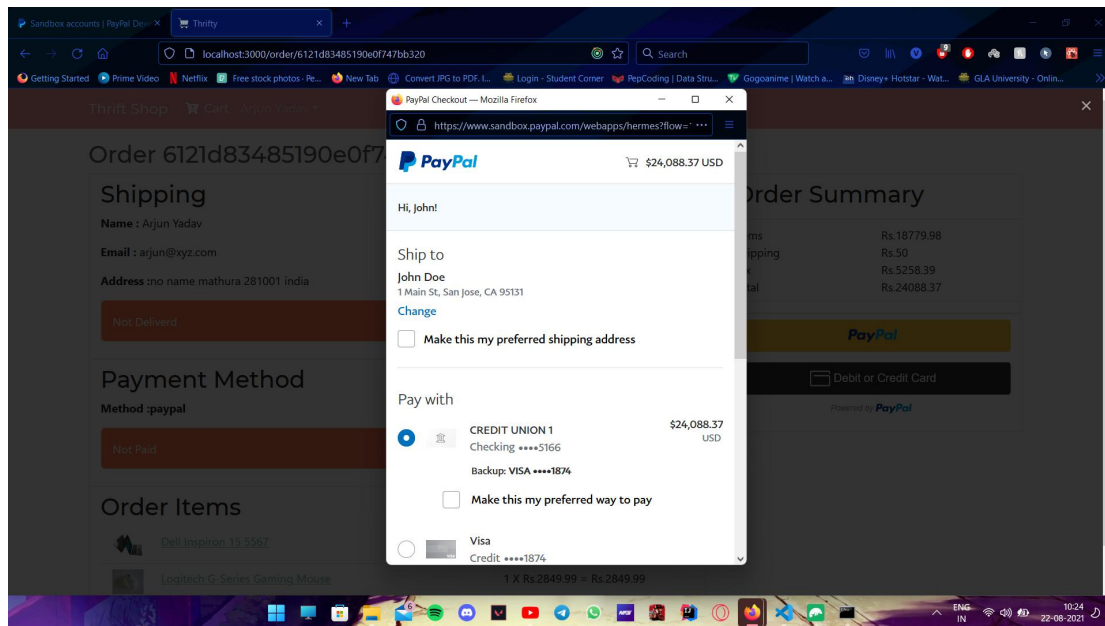
## 4.5.1 Selecting Payment option



This is a page we've to choose payment Gateway, we can also include more payment methods as per the requirement.



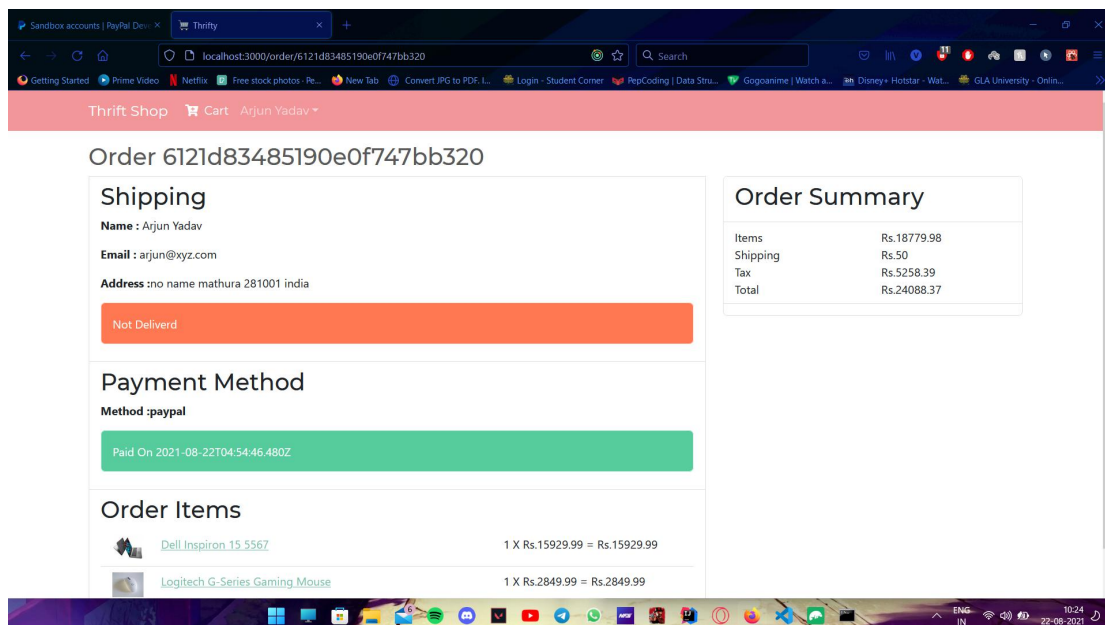
## 4.5.2 Payment Gateway



The most important feature of any e-commerce application is payment gateway.

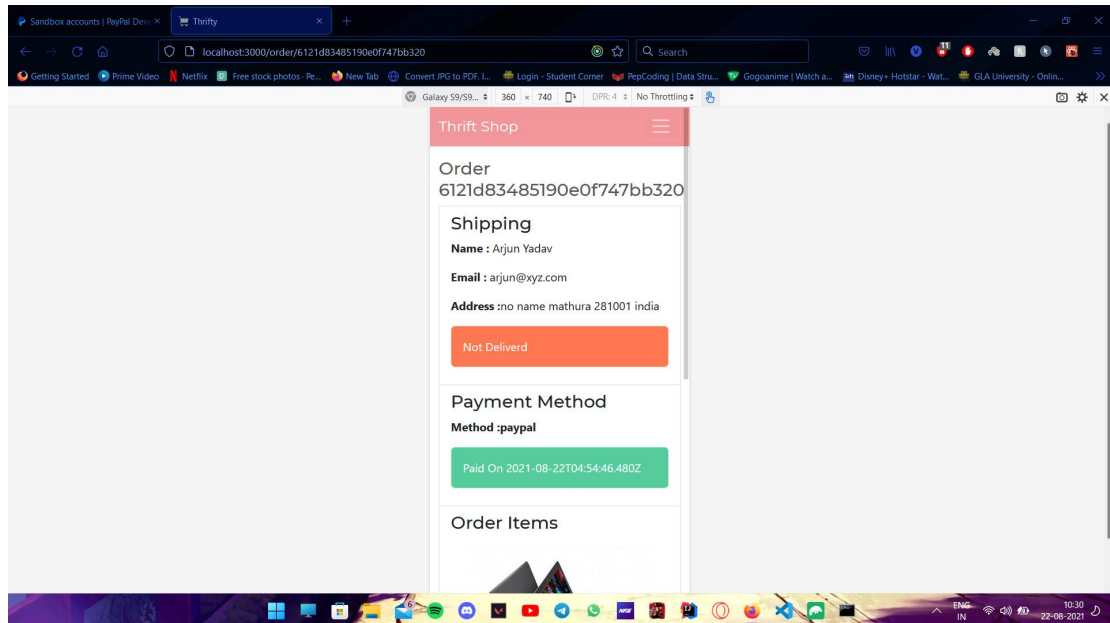
If the store only wants to use the credit card system, then it could be losing a lot of customers because many people nowadays want to use PayPal. Therefore, the application integrates both credit card and PayPal system.

## 4.5.3 Order Progress



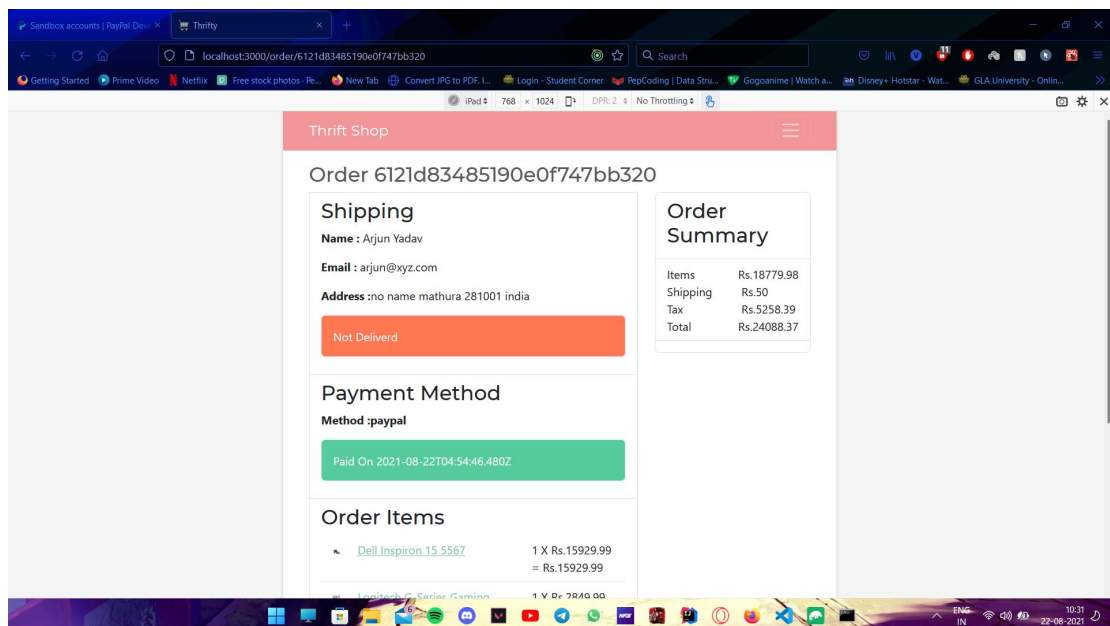
## 5 Responsiveness of web application

### Mobile View:



The complete web application is responsive to all the platforms like mobile and tablets which increases it utility to a great extend.

### Tablet View:



## 6 Summary

The achievement of the thesis is researching the basic components of MERN stack technology: MongoDB, ExpressJS framework, ReactJS library and NodeJS platform. Using MERN stack technology in conjunction with Braintree to build an e-commerce web application with payment gateway.

The advantages are performing the basic functions of a product search website for customers, making it easy for customers to find categories that have the products they are looking for. Gives small stores a platform to store information and promote their products. Password data of accounts when logging in to the system is stored in a secure database. The management interface, statistics of the user and admin are easy to use for everyone.

The disadvantages are online chat functions between shop owners and customers are not yet supported as well as between shop owners and administrators. The current product search algorithm locates by the user location that is not really optimal, needs to be improved to speed up the search even more.

Since the purpose of the thesis is the e-commerce application, the understanding about MERN technologies and applying it to this app is the most important. Overcome current shortcomings, listen to customers' comments and making improvements, helping users have a great experience in the future.

## References

- E-commerce Definition – What is E-commerce? [Internet]. Shopify.com. Available from: <https://www.shopify.com/encyclopedia/what-is-ecommerce>
- Advantages of E-commerce [Internet]. Thebalancesmb.com 2019 [cited 20 November 2019]. Available from: <https://www.thebalancesmb.com/ecommerce-pros-and-cons-1141609>
- JavaScript [Internet]. Mozilla.org. Available from: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- NodeJS Introduction [Internet]. Tutorialspoint.com. Available from: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)
- NodeJS Pros and Cons [Internet]. Mindinventory.com. Available from: <https://www.mindinventory.com/blog/pros-and-cons-of-node-js-web-app-development/>
- NodeJS use cases [Internet]. Credencys.com. Available from: <https://www.credencys.com/blog/node-js-development-use-cases/>
- Express.js Introduction [Internet]. Mozilla.org. Available from: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction)
- MongoDB [Internet]. MongoDB.com. Available from: <https://docs.mongodb.com/manual/introduction/>



- Virtual-DOM [Internet]. Reactjs.org. Available from:  
<https://reactjs.org/docs/faqinternals.html>
- Component [Internet]. Reactjs.org. Available from:  
<https://reactjs.org/docs/components-and-props.html>
- Props and State [Internet]. Flaviocopes.com. Available from:  
<https://reactjs.org/docs/components-and-props.html>
- Pros and Cons of ReactJS [Internet]. Javatpoint.com. Available from:  
<https://www.javatpoint.com/pros-and-cons-of-react>
- Stack technology [Internet]. Stackshare.io. Available from:  
<https://stackshare.io/stacks>
- MERN stack concept [Internet]. Mongodb.com. Available from:  
<https://www.mongodb.com/mern-stack>