Bachelor's thesis

Information and Communications Technology

2020

Thao Huynh

THE DESIGN AND DEVELOPMENT OF AN E-COMMERCE WEB APPLICATION

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information and Communications Technology

2020 | 35 pages

Supervisor: Tapani Ojanperä

Author: Thao Huynh

THE DESIGN AND DEVELOPMENT OF AN E-COMMERCE WEB APPLICATION

E-commerce has recently become a popular way of shopping for customers because of its convenience. Since the number of smartphone users worldwide has risen significantly, a responsive commercial website which is able to adapt the screen of mobile devices plays an important role in business these days.

The aim of this thesis was to develop a responsive, user friendly, scalable e-commerce website with a modern, high performance JavaScript framework called React.

React, HTML5, CSS3, Bootstrap 4, Node.js, and JSON are technologies used to built this project. After the developing process, the project was deployed on the Internet. Netlify was chosen as the hosting platform.

This website was built to enable users to complete several actions such as seeing the list of products with images, sorting the list in an ascending or descending order, filtering the products by size, adding preferred items to cart or removing them.

The responsive test, functionality test, and browser compatibility were performed on different platforms. The result was good as the website met all the requirements.

The application is simple, elegant, and functional with important features for an online store. The website works well as requested, which makes this project successful.

KEYWORDS:

E-commerce website, responsive web application, React, shopping cart, online store, JSON, Netlify.

CONTENTS

FIGURES

TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CRUD	Create, Read, Update, Destroy
DOM	Document Object Model
ES7	ECMAScript 7
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
I/O	Input/Output
JSON	JavaScript Object Notation
RWD	Responsive Web Design
REST	Representational State Transfer
UI	User Interface

1 INTRODUCTION

E-commerce is known as buying and selling processes or other types of commercial transaction through the Internet. In 2016, e-retail was worth about 1.86 trillion U.S. dollars, and it is expected to grow at a more rapid pace (Vakulenko et al. 2019, 461).

E-commerce has recently become a popular way of shopping for customers because of its convenience. It also enables small and medium companies as well as large corporations to boost up sales and maximize profit, which is hard to do with traditional offline retail.

Since the number of smartphone users worldwide is rising significantly, a responsive commercial website which is able to adapt to the screen of mobile devices plays an important role in business these days.

This thesis aims to design and develop a responsive e-commerce website for small shops so that they can run their own e-business with almost no set up cost .

JavaScript is now one of the most powerful programming languages for web development in the world. JavaScript enables developers to code highly responsive websites that provide dynamic functionality to react the user's requests, thereby improving the user experience. (Johannes et al. 2019, 1271)

React creates a data structure cache stored in memory. Every time the code is modified, React recognises the changes made and then updates them to the browser. This special feature will enhance the performance of the webpage as the React library only renders components that actually change instead of loading the whole page like other traditional ways. (Javeed 2019)

That is why React, an open-source JavaScript library, is used in this thesis to build a user friendly, scalable webpage with high performance.

The goal of the thesis is to develop a functional responsive e-commerce web application using React for an online fashion store.

This thesis consists of eight chapters. The first one contains the background of the project, thesis goal and structure. The second chapter describes the technology used to develop the website. The general web functionality for different users, database, front-

end and back-end side of the project is explained in chapter three. Chapter four introduces the development process and the result obtained. The fifth chapter defines the domain name and hosting platform of the website. Testing process is explained in chapter six while chapter seven is about the discussion with suggested further improvement. Chapter eight presents the conclusion of the whole project.

2 METHODS

There are several tools and environments needed to build this web application for both front-end and back-end side. This chapter explains the technology such as HTML5, CSS3, Bootstrap 4, JavaScript, React, Node.js, JSON, Netlify, Visual Studio Code, ES7 snippets used to develop the project.

2.1 HTML5

Hypertext Markup Language (HTML) is the standard markup language. It is often used to display content of a webpage in a web browser. It is also compatible with other technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents that define the structure and textual content of the page from a server or a local storage and then render them into actual web pages with multimedia.

HTML consists of a set of elements which are represented by tags, written using angle brackets (Figure 1).

```
<!DOCTYPE html>
<html>

<head>
    <title>Title of the page</title>
</head>

<body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
    ......
</body>

</html>
```

Figure 1. The basic structure of HTML page.

This thesis project uses HTML5 which is the fifth version of HTML. HTML5 allows JavaScript to run in the web browser while HTML does not. Moreover, HTML5 supports audio and video elements which are unavailable with HTML (Tumuluri et al. 2019, 373).

2.2 CSS3

Cascading Style Sheets (CSS) is a style sheet language used for adding style to web documents.

CSS allows users to separate the content and its style, including fonts, colors, borders and padding. The style declarations are often saved in external .css files. By making changes to the CSS style sheet, developers can automatically apply it to all pages of a website. The more complex the website is, the more time CSS saves for developers. Another benefit is that web pages always have consistent styling. (Ndia et al. 2019)

This thesis project uses CSS3 which is the latest version of the Cascading Style Sheets language (Figure 2). It aims to provide users with floating effects which place buttons on the right side of the navigation bar, as well as new flexible box or grid layouts. (Wang and Si 2019, 360)

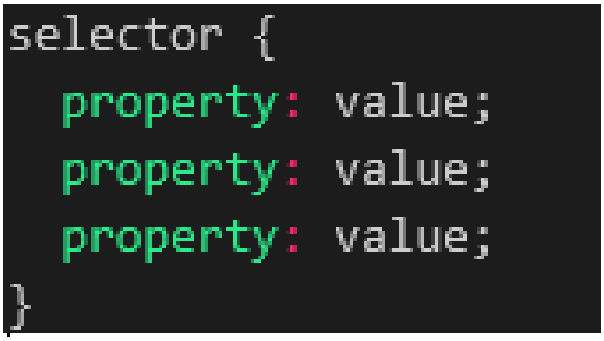

Figure 2. The basic syntax of CSS.

2.3 Bootstrap 4

Bootstrap is a front-end framework that offers reusable code written in HTML, CSS, and JavaScript. It provides users with different styles of color, size, font and layout (Subramanian 2019, 315). The result is to add more eye-catching design elements to a project, allowing interactivity such as transitions, animation, and modal popups (Krause 2016a, 173).

Bootstrap offers several JavaScript components with jQuery plugins, allowing developers to deal with dialog boxes and tooltips (Krause 2016a, 189). Each Bootstrap theme contains an HTML file, CSS definitions, and JavaScript code.

The most important features of Bootstrap are its layout components. The basic layout component is "Container" which contains other elements in the page. Developers can make a choice of a fixed-width option and a fluid-width option. The former uses predefined widths, based on the size of the screen of devices whereas the latter automatically span the entire width of the viewport, creating a full width container. (Krause 2016b, 34)

Bootstrap 4, the latest version of Bootstrap was chosen for this thesis because it has several major improvements as follows:

- The CSS source file changes from LESS to SAAS is one of the main difference between the two versions of Bootstrap.
- Bootstrap 4 uses rem or em as the unit system instead of pixels (px).
- Bootstrap 4 offers 5 grid tiers to improve the grid system while Bootstrap 3 only allows 4 tiers.
- Table inverse is available with Bootstrap 4 whereas Bootstrap 3 does not support it. (Krause 2016c, 23)

2.4 JavaScript

JavaScript, known as JS, is a programming language for implementing complex features on web pages. JavaScript offers a variety of beautiful interfaces for creating well-designed websites, improving the interactivity on the page. (DiPierro 2018, 9)

JavaScript has a full integration with HTML and CSS. JavaScript also enables developers to create servers, mobile applications. That is why JavaScript has recently become an essential part for creating browser interfaces.

JavaScript is supported by all major web browsers and used in server-side deployments, usually via Node.js. (DiPierro 2018, 10)

2.5 React

React is a JavaScript library for building user interface component for the development of web or mobile applications. It is an open-source framework responsible for the application's view layer.

It is easier to create dynamic web elements with React compared to JavaScript because React allows its component to be reused, which offers less coding and development time.

There are 2 key features of React as follows:

2.5.1 JSX

JSX is a syntax extension to JavaScript which allows HTML quoting to define what the user interface should be displayed. By using JSX, developers can write HTML tag syntax in the combination of JavaScript. Developers can choose to write in pure old JavaScript but using React will be much easier to modify the webpage due to avoiding complex JavaScript DOM structures. (Japikse et al. 2017, 353)

2.5.2 Virtual DOM

When a web page is rendered, the browser creates a Document Object Model (DOM) of the page that defines HTML elements as a tree of objects (Figure 3).

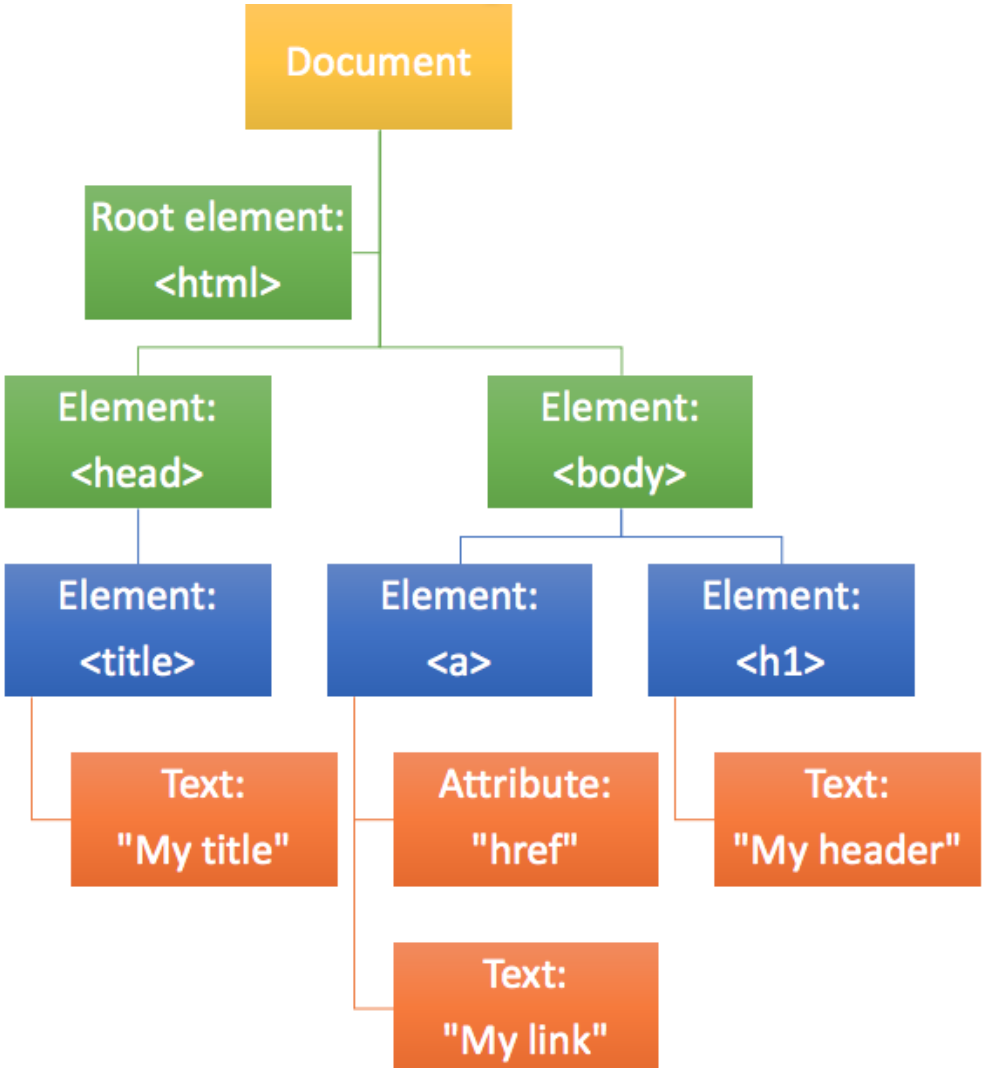

Figure 3. DOM of a Webpage (JavaScript HTML DOM 2020).

If a developer chooses not to use React for a web application, the website will use HTML to update its DOM. This can lead to a slow experience for complex dynamic websites as the entire DOM reloads every time the page is refreshed.

On the other hand, React creates a virtual DOM (VDOM) which is actually the copy of the real DOM. Then, React computes the differences between two DOMs and manipulates only the changes instead of reloading the whole page. This makes updating a website take less computing power and processing time, especially when compared to other front-end technologies. (Chęć and Nowak 2019, 119)

2.6 Node.js

Node.js is a JavaScript runtime environment that is used for developing server-side and networking applications. Node.js is free and runs on various platforms such as Windows, Linux, Unix, Mac OS X.

Because a Node.js app runs in a single thread, this allows Node.js to eliminate the waiting time without creating a new thread for every request. Then Node.js can continue with the next request from users, which is very memory efficient. However, the application will be asynchronous programming model, which can be unfamiliar with some developers who use linear blocking I/O programming. (Wang et al. 2017, 520)

Node.js runs the V8 JavaScript engine and executes code very fast. It is able to handle thousands of concurrent operations. When there is a blocking thread, it saves the computing power by moving to the next API request and resumes the previous connection when the response comes back. Moreover, the Node.js community is very strong. There are now more than 400,000 packages for developers to use for their projects. (ibid.)

The unique feature that makes Node.js special is that it enables front-end developers to write the back-end code without learning a completely new programming language. With its powerful libraries, Node.js can create, read, write, and delete code on the server and in the database. (ibid.)

2.7 JSON

JavaScript Object Notation (JSON) is a standard format for data interchange based on JavaScript object syntax.

When exchanging data between a browser and a server, the data can only be text. However, JavaScript objects are not text. That is why JSON is used to convert these objects to text for transmitting data in web applications.

Although JSON was derived from JavaScript, it is language-independent because many modern programming languages can parse and translate JSON. This property makes JSON one of the most popular data-interchange format. (Carter 2018, 195)

JSON supports two structures

- A collection of name/value pairs such as an object, a hash table, a struct.
- A list of values such as an array, a list (Figure 4).

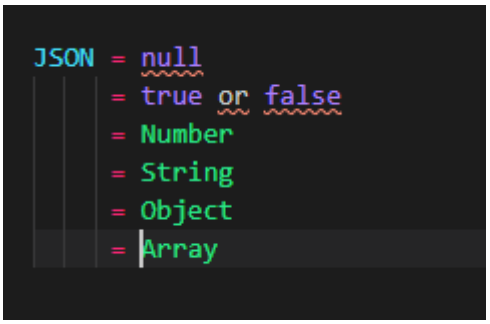

Figure 4. The general syntax of JSON.

2.8 Netlify

Netlify is a platform that supports web developers to publish their website on the internet. It offers not only domain names but also hosting packages. Its service is free of charge for basic accounts or very cheap for premium ones. By unifying the elements of the modern decoupled web, Netlify uses a special deploying technique to make users' projects become performant, secure, and scalable websites and apps. (Netlify 2020)

It now has half a million users such as businesses and developers, and it is building millions of web projects globally. (Netlify 2020)

2.9 Visual Studio Code

Visual Studio Code is a code editor which runs on desktops. It is developed by Microsoft, and it is available for Linux, macOS and Windows. It provides users with built-in support for various programming languages such as JavaScript, Node.js, C++, C#, Java, Python, PHP, Go. Developers can change the theme, write code, install extensions for more useful functionalities, and use GitHub or Git version control system. Visual Studio Code is a free open source. (Microsoft 2020)

2.10 ES7 React/Redux/GraphQL/React-Native snippets

This tool is an extension for Visual Studio Code which offers React/Redux snippets with Babel plugin features.

By pressing command + shift + P (MacOS) or Ctrl + shift + P (Windows and Linux), ES7 Snippet provides users with a list of short commands for longer snippets of code. Running the commands is easy by typing them in the editor and then pressing enter. (Dsznajder 2020)

2.11 Responsive web application

Responsive web design (RWD) is a method for designing web pages so that they can be loaded properly on various differerent devices and screen sizes. A responsive web application (RWA) is often the combination of flexible grids and images. These responsive design patterns support the page elements sizing to shrink or enlarge on different screens.

RWA can adapt to any device such as a desktop, a laptop, a tablet, or a mobile phone because page sizing is rendered in relative units like percentages, instead of pixels or points. Flexible images are processed in the same way so that they are not displayed out of their containers to fit a smaller screen.(Frain 2015, 39)

3 OVERVIEW OF THE WEB APPLICATION

The project is a shopping web application for an online fashion store. This chapter describes the general web functionality for different users, database, front-end and back-end side of the project.

3.1 User groups

There are two types of users: administrators and customers. This project is built to enable different users to perform several actions as follows:

Table 1. The different user actions.

Administrators	Customers
View the list of products with images	View the list of products with images
Sort the list as ascending or descending order	Sort the list as ascending or descending order
Filter the products by size	Filter the products by size
Add preferred items to cart	Add preferred items to cart
Remove the chosen items from cart	Remove the chosen items from cart
See the total amount of money cost	See the total amount of money cost
Add new products	Not valid
Add images	Not valid
Delete products	Not valid

3.2 Database

This thesis uses JSON's data with different types

- Number: a signed decimal number
- String: a sequence of zero or more Unicode characters.
- Boolean: either of the values true or false
- null: represents the absence of a value
- Array: a list of no or many values, each of which can be of any above type.

The data of this project also consists of Object which is a collection of key–value pairs. Keys are strings and the values of objects can be assigned any above types.

3.3 Back-end size

The JSON server is a Node.js module that provides users with an interface like other REST APIs. JSON Server will get data from a JSON file of the main project folder and deploy it into a RESTful database.

3.4 Front-end side

The client-side is built with React, HTML5, CSS3 and Bootstrap 4. While HTML5 defines the structure of a webpage, CSS3 and Bootstrap 4 are used to provide consistent styles of color, size, font, spacing and layout.

Moreover, React tracks the differences between the current UI and the previous UI by comparing its virtual DOM versions. Since React updates only those changes in the real DOM, this algorithm will help to avoid wasted DOM manipulation operations in the browser, which can lead to slow and inconvenient user experience.

4 IMPLEMENTATION AND RESULT

This chapter explains the main steps of development process and the outcome obtained. A simple, scalable website with high performance was created with two pages. The Home page allows users to complete several actions such as seeing the list of products with images, sorting the list in an ascending or descending order, filtering the products by size, adding preferred items to cart or removing them. The About page contains the developer's information.

4.1 The environment was installed

- Install Node.js version 13.12.0
- Install JSON server by executing the command
  $ json-server public/dataBase.json –port 3001
- Install Visual Studio code and ES7 React/Redux/GraphQL/React-Native code snippets

4.2 Structure of the project

The project has 3 main parts (Figure 5):

- HTML: index.html
- CSS: index.css and App.css
- JavaScript: App.js and other components

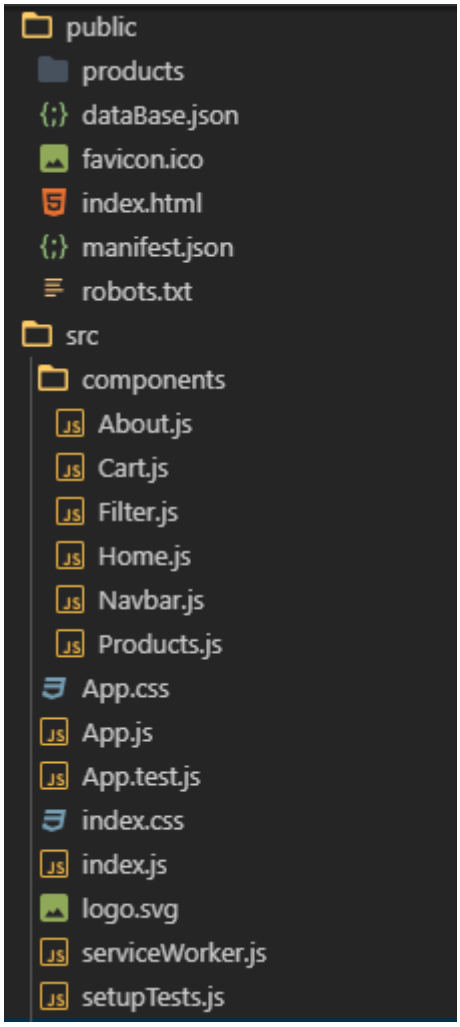

Figure 5. The structure of this project.

- Index.html file: contains the external CSS sources from Bootstrap 4
- Index.css file: declares my custom CSS style
- App.css file: React default CSS style

4.3 Basic HTML of the website

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

    <link href="css/creative.min.css" rel="stylesheet">
    <link href="vendor/magnific-popup/magnific-popup.css" rel="stylesheet">


    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.
```

Figure 6. The main part of the HTML file.

Figure 6 is the fundamental part of the HTML file. All the main functions are defined in .js files. The <!DOCTYPE html> declaration defines this document is written in HTML language. The file contains the external CSS sources used for this project. The web app is given the id of “root”.

4.4 Basic CSS of the website

```css
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  background-color: #ffebee !important;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}

h1 {
  color: #7cb342;

}

#zoom {
  width: 180px;
  height: 261px;
  transition: width 2s, height 2s;
}
```

Figure 7. The fundamental part of the CSS file.

Figure 7 shows the fundamental part of the CSS file. It declares the style of body, code, h1 and an element with id=”zoom” for transition effect. This is a combination of React default style for a web application and the author’s own custom style.

4.5 App.js file

```
import React, { Component } from 'react'
import Navbar from './components/Navbar'
import { Route, BrowserRouter } from 'react-router-dom'
import Home from './components/Home'
import About from './components/About'

class App extends Component {
  render() {
    return (
      <BrowserRouter>
        <div className="App">
          <Navbar />

          <Route exact path='/' component={Home}/>
          <Route path='/about' component={About} />


        </div>
      </BrowserRouter>
    )
  }
}

export default App;
```

Figure 8. App.js file of the website.

Figure 8 illustrates the code for building the structure of the website. The navigation bar is created with two pages called Home and About. Home page enables customers to see the list of products with images, to sort the list in an ascending or descending order, to filter the products by size, to add preferred items to cart or to remove them. Images are displayed as thumbnails. However, customers can zoom in the images by hovering a mouse on them or clicking a link to see a full-sized image.

React Router Dom was imported to add the path for each page, making the navigation bar work properly.

4.6 Components

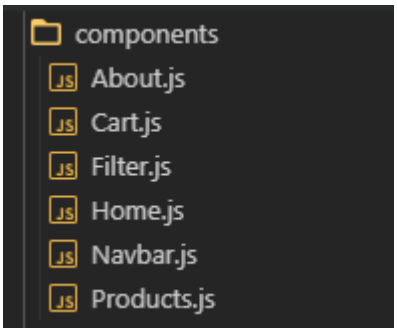

Figure 9. Component files.

- Component About creates a page of the developer's information
- Component Cart contains button of removing items from cart and calculates the total amount of money cost.
- Component Filter creates drop-down lists for sort and filter functions
- Component Home creates the main page.
- Component Navbar displays the navigation bar.
- Component Products creates buttons to add items to cart and allows users to see specific products.

The component Home contains different functions to :

- Fetch the data from http://localhost:3001/products
- List the products
- See each product with full-sized images
- Add products to cart
- Count the items chosen
- Remove products from cart
- Sort the price in an ascending or descending order
- Display the products by different available size

4.7 Rendering the web application

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

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

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Figure 10. The fundamental part of file index.js.

File index.js is responsible for rendering App.js by a specific ReactDOM() method.

4.8 Database – JSON

Create a file with name dataBase.json. This file contains the data of the project.

When exchanging data between a browser and a server, the data can only be text. However, JavaScript objects are not text. That is why JSON is used to convert these objects to text for transmitting data in this web applications.

The JSON structure consists of one product object which has twelve data sets assigned. Each data set includes six properties as follows (Figure 11):

Figure 11. Database Structure.

4.9 Back-end module

4.9.1 Installing the JSON server

JSON Server is available as a NPM package. The installation can be done by using the Node.js package manager:

$ npm install -g json-server

By adding the -g option the package is installed globally on the system.

4.9.2 Running the server

Running The Server by executing the command

$ json-server public/dataBase.json –port 3001

The following HTTP endpoints are created automatically by JSON server:

- GET /products
- GET /products/{id}
- POST /products

- PUT /products/{id}
- DELETE /products/{id}

4.10 Interface layout

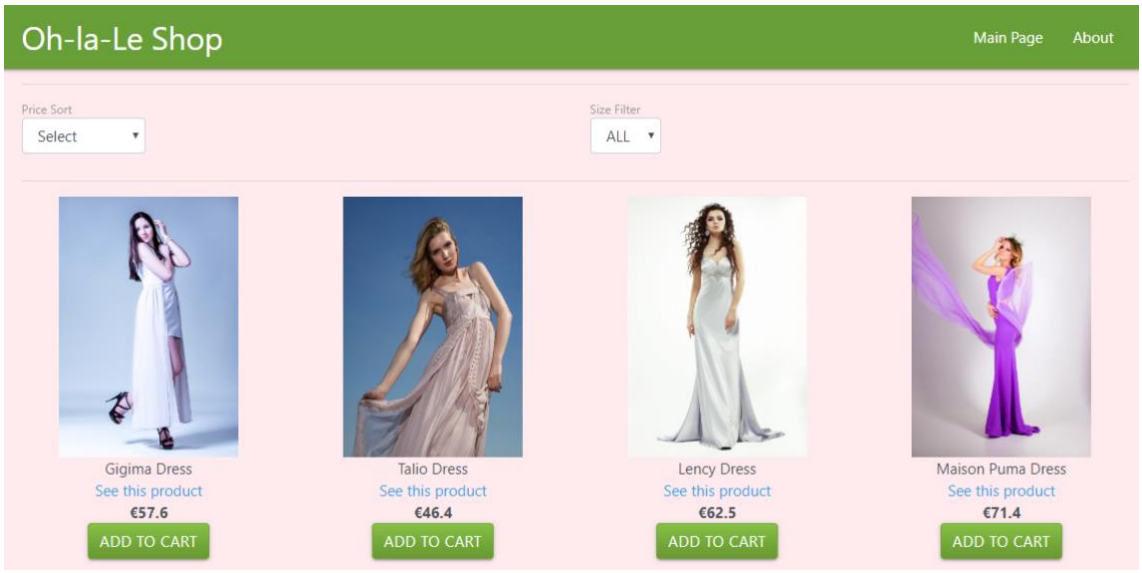

Figure 12. The main page of the website.

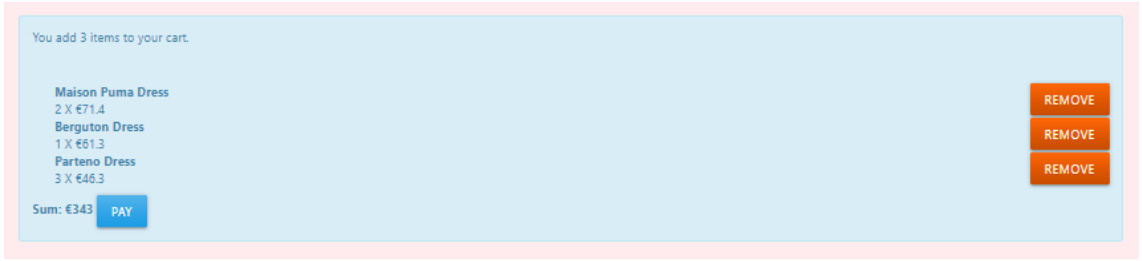

Figure 13. The shopping cart.

Figure 14. The About page.

5 PUBLISHING THE WEBSITE TO THE INTERNET

This chapter defines the domain name and hosting platform chosen for the project. The current address of the website is https://oh-la-le.netlify.app. Netlify is the company that provides hosting service.

5.1 Choosing a domain

The Domain Name System (DNS) is a naming system that defines addresses to web pages. As every machine on the Internet is given a unique identifier called IP address, this address allows computers to successfully communicate with others in the given network.

However, it is very difficult to remember the IP addresses of all websites because these addresses are sets of numbers. That is why a domain name is necessary as words are easier to remember than numbers are. To visit a website, users only need a domain name, and then DNS translates it into a machine IP address, making the connection to the correct website.

The domain name is unique for each website, and it can be bought from the domain service provider. As the web is hosted on Netlify platform, its address is https://oh-la-le.netlify.app.

5.2 Choosing a web host service

After having compared various web hosting service providers, I chose Netlify. There are many reasons which prove it to be one of the most approriate web hosting company for this project. Netlify is free for basic accounts or very cheap for premium ones, easy to establish, secured and fast enough to run a responsive web app. It also meets all the requirements for running a React application.

6 TESTING

Testing is a crucial part of quality assurance to check whether the website is functional, responsive and compatible with most of modern browsers.

For better user experience, the web content is also checked without grammar mistakes.

The responsive test was performed on different devices such as Android, iOS and Windows smart phones. The result was satisfactory as the website works well on all tested mobile phones. All the images, buttons, and the layout are displayed as expected.

The functionality test was done by checking all the links, seeing products with thumbnails and full-sized images, sorting the list in both ascending and descending orders, filtering the products by size, adding preferred items to cart and removing them. All the functions work properly. The test was successful as requirements.

The browser compatibility was tested in several modern web browsers, such as Safari Firefox, Google Chrome. The result was good because the website can be run with all of these browsers. Users can visit this website no matter what browser they use (Figures 15, 16 and 17).

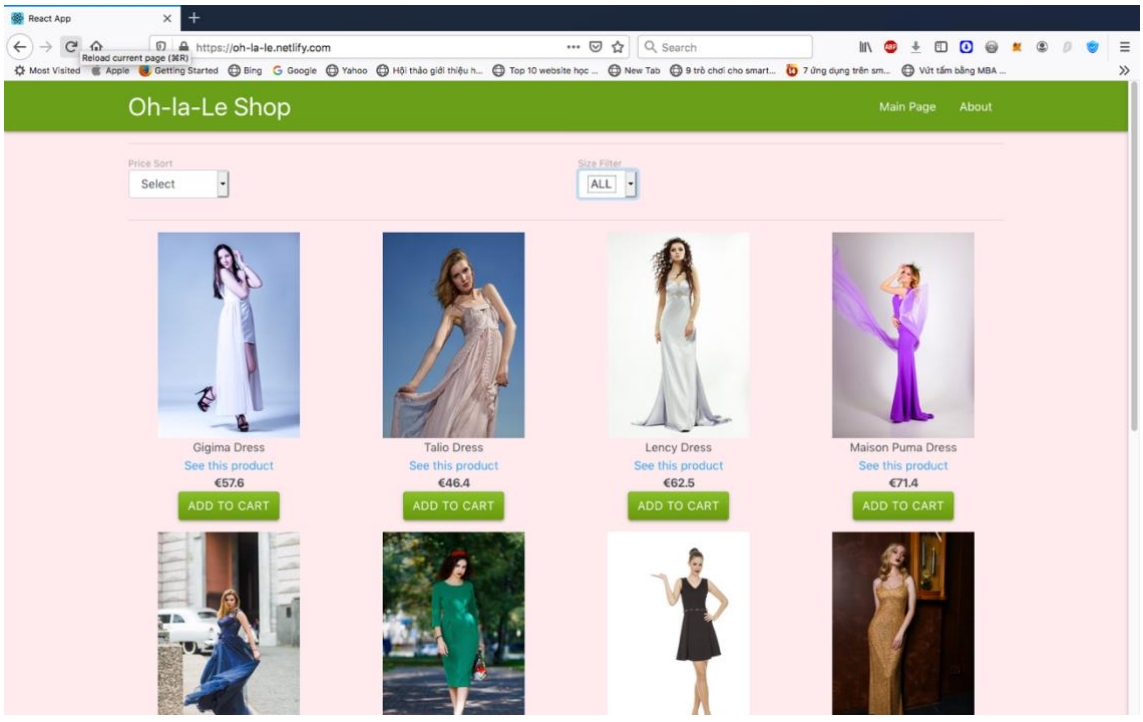

Figure 15. The webpage tested in Firefox.

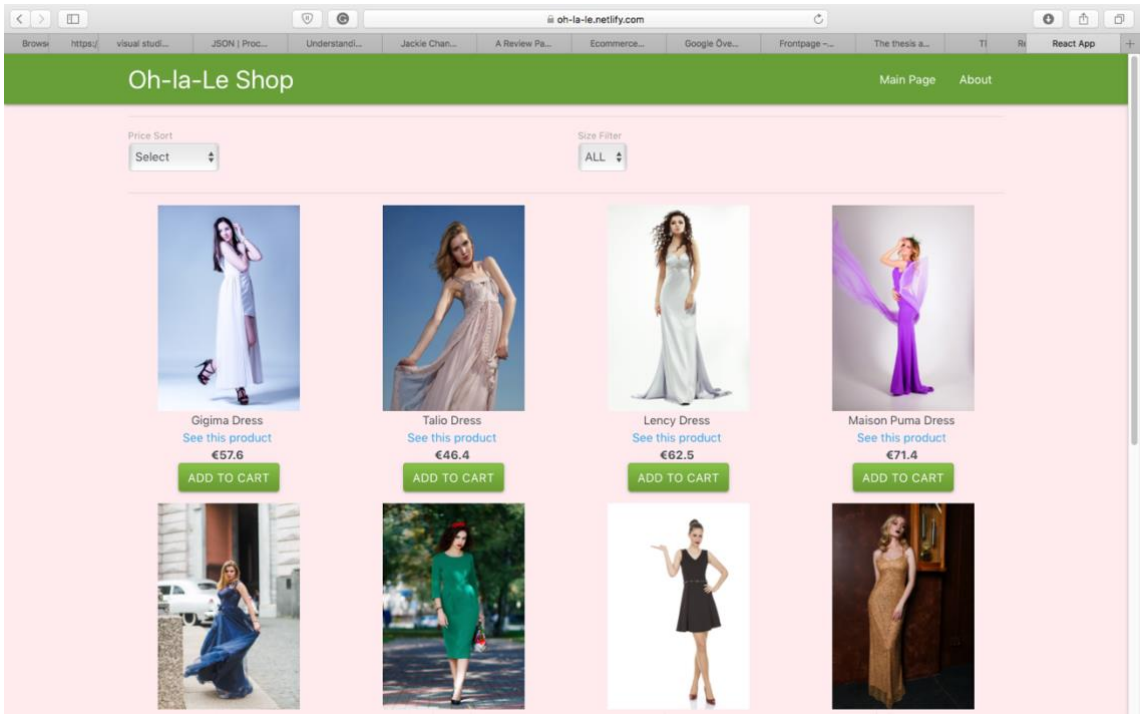

Figure 16. The webpage tested in Safari.

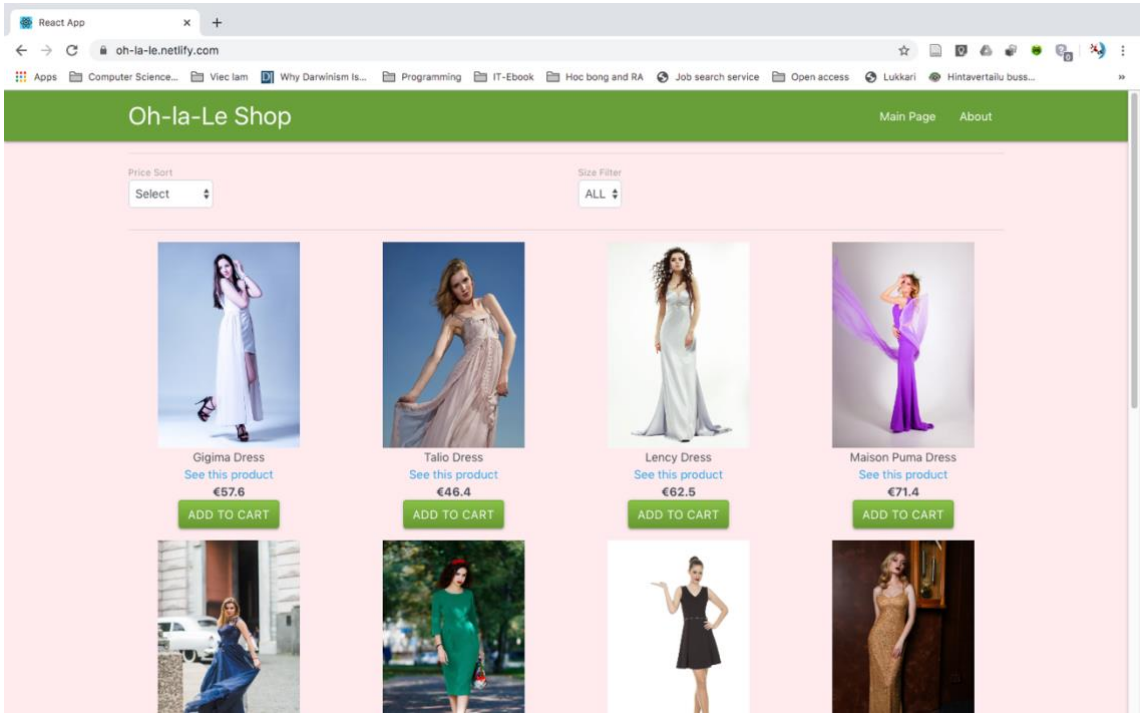

Figure 17. The webpage tested in Google Chrome.

7 DISCUSSION

With technological advancements, the development of a website is assisted by various platforms with different programming languages such as JavaScript, PHP, .NET and Java. A variety of web builders like Wordpress, Wix, OpenCart provide users who do not have any IT skills of coding with ready-made website templates. However, these templates are often not able to be edited freely in the way that developers want. Another major disadvantage is the high service fee for building and operating process.

Therefore, React was chosen for this project. Advantages of JavaScript language are its speed, simplicity, server load, and interoperability. The speed of JavaScript is better because it does not need to render unnecessary outside resources and backend server. Moreover, JavaScript can also be inserted into any web page easily since it is fully integrated with HTML and CSS.

React is a popular JavaScript lightweight frameworks to build a web application because of its productivity. React creates a data structure cache stored in memory. Every time the code is modified, React recognises the changes made and then updates it to the browser. This special feature will improve the performance of the webpage.

Due to the limited time and human resources, payment and invoices are not the targets of this project. However, these properties are recommended as further development in order to improve the economic efficiency of the website. The shop should also inform its subscribers with new promotion and discount programs. Optimizing the search engine is another suggestion as website visibility and network marketing are playing more important roles in running an e-business. In addition, more languages, especially Finnish, need to be available to non-English customers for their convenience.

8 CONCLUSION

The aim of this thesis was to develop a responsive, user friendly, scalable ecommerce website with high performance using React. The website works well as requested, which makes this thesis project successful.

This website was built to enable users to do several actions like seeing the list of products with thumbnails and full-sized images, sorting the list in an ascending or descending order, filtering the products by size, adding preferred items to cart or removing them.

React, HTML5, CSS3, Bootstrap 4, Node.js, and JSON were technologies used to built this project. After developing process, the project was deployed on the Internet. Netlify was chosen as the hosting platform.

The responsive test, functionality test, and browser compatibility were performed on different platforms. The result was good as the website met all the requirements.

The web application is simple, elegant and functional with important features for an online store. It provides clients with a solution to develop their business at a very low cost.

REFERENCES

Carter, P. A. 2018. Understanding JSON. In *SQL Server Advanced Data Types*, Peter A. Carter, 181–200. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-3901-8_6.

Chęć, D. and Nowak, Z. 2019. The Performance Analysis of Web Applications Based on Virtual DOM and Reactive User Interfaces. In *Engineering Software Systems: Research and Praxis*, ed. Piotr Kosiuczenko and Zbigniew Zieliński, 830:119–134. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-99617-2_8.

DiPierro, M. 2018. The Rise of JavaScript. *Computing in Science & Engineering* 20: 9–10. https://doi.org/10.1109/MCSE.2018.011111120.

Dsznajder. 2020. ES7 React/Redux/GraphQL/React-Native snippets - Visual Studio Marketplace. https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets. Accessed April 14.

Frain, B. 2015. *Responsive Web Design with HTML5 and CSS3*. Birmingham: Packt Publishing Limited : [distributor] Bertrams : [distributor] Ingram Book Company.

Japikse, P.; Grossnicklaus, K. and Dewey, B. 2017. React. In *Building Web Applications with Visual Studio 2017*, 329–387. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-2478-6_9.

JavaScript HTML DOM. 2020. https://www.w3schools.com/js/js_htmldom.asp. Accessed April 14.

Javeed, A. 2019. Performance Optimization Techniques for ReactJS. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 1–5. Coimbatore, India: IEEE. https://doi.org/10.1109/ICECCT.2019.8869134.

Johannes, D.; Khomh, F. and Antoniol, G. 2019. A large-scale empirical study of code smells in JavaScript projects. *Software Quality Journal* 27: 1271–1314. https://doi.org/10.1007/s11219-019-09442-9.

Krause, J. 2016a. Active Components. In *Introducing Bootstrap 4*, Jörg Krause, 171–205. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-2382-6_8.

Krause, J. 2016b. Structure of the Page. In *Introducing Bootstrap 4*, Jörg Krause, 33–44. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-2382-6_3.

Krause, J. 2016c. Introduction to Bootstrap. In *Introducing Bootstrap 4*, Jörg Krause, 23–32. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-2382-6_2.

Microsoft. 2020. Visual Studio Code - Code Editing. Redefined. https://code.visualstudio.com/. Accessed April 14.

Ndia, G. J.; Muketha, G. M. and Omieno, K. K. 2019. A Survey of Cascading Style Sheets Complexity Metrics. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3405783.

Netlify. 2020. Netlify: All-in-one platform for automating modern web projects. *Netlify*. https://www.netlify.com/. Accessed April 14.

Subramanian, V. 2019. React-Bootstrap. In *Pro MERN Stack*, Vasan Subramanian, 315–376. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4391-6_11.

Tumuluri, R.; Dahl, D.; Paternò, F. and Zancanaro, M. 2019. Standardized representations and markup languages for multimodal interaction. In *The Handbook of Multimodal-Multisensor Interfaces: Language Processing, Software, Commercialization, and Emerging Directions - Volume 3*, ed. Monash University, Sharon Oviatt, Björn Schuller, University of Augsburg and Imperial College London, Philip R. Cohen, Monash University, Daniel Sonntag, et al. Association for Computing Machinery. https://doi.org/10.1145/3233795.3233806.

Vakulenko, Y.; Shams, P.; Hellström, D. and Hjort, K. 2019. Service innovation in e-commerce last mile delivery: Mapping the e-customer journey. *Journal of Business Research* 101: 461–468. https://doi.org/10.1016/j.jbusres.2019.01.016.

Wang, J.; Dou, W.; Gao, Y.; Gao, C.; Qin, F.; Yin, K. and Wei, J. 2017. A comprehensive study on real world concurrency bugs in Node.js. In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 520–531. Urbana, IL: IEEE. https://doi.org/10.1109/ASE.2017.8115663.

Wang, J. and Si, Z. 2019. Technical Research of Clearing Floating Effect in CSS Cascading Style Sheets. In *Advances in Graphic Communication, Printing and*

Packaging, 543:359–364. Lecture Notes in Electrical Engineering. Singapore: Springer Singapore. https://doi.org/10.1007/978-981-13-3663-8_48.