Thuan Tran

BUILDING A MUSICAL INSTRUMENT STORE APPLICATION USING REACTJS

Thesis
CENTRIA UNIVERSITY OF APPLIED SCIENCES
Information Technology
June 2024

ABSTRACT

Centria University of Applied Sciences	**Date** May 2024	**Author** Thuan Tran
Degree programme Bachelor of Engineering, Information Technology		
Name of thesis BUILDING A MUSICAL INSTRUMENT STORE APPLICATION USING REACTJS		
Centria supervisor Heikki Ahonen, Nina Hynynen		**Pages** 33

The purpose of the thesis was creating a responsive e-commercial web application containing musical instrument content that satisfies the needs of businesses and customers to selling and purchasing products via online store sales. Thus, saving a great amount of time for both parties.

The thesis was divided into three different parts. The first part consists of the prerequisites to develop the application, including languages chosen and tools to handle the source code. The second part covers the planning phase for the web design. For the third section, the content focuses on the client-side with real-time development processes using ReactJS, HTML, CSS, React Bootstrap, with Node.js runtime environment is in use for data testing and handling the server loads and requests.

In the end, the product will be implemented with basic CRUD functionalities on the front-end side. For future development, the application aims to deliver better performance to users. The application's goal is to host a web service and database server in order to support scalability, high availability, and user-friendly experience to customers and businesses.

Key words
CSS, HTML, JavaScript, ReactJS, React Bootstrap, Responsive Web Design, RESTful API

CONCEPT DEFINITIONS

API

Application Programming Interface

CI

Continuous Integration

CD

Continuous Development

CSS

Cascading Styles Sheets

DOM

Document Object Model

HTML

HyperText Markup Language

HTTP

Hypertext Transfer Protocol

IDE

Integrated Development Environment

REST

Representational State Transfer

RWD

Responsive Web Design

SaaS

Software as a Service

UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

ABSTRACT
CONCEPT DEFINITIONS
CONTENTS

CODES

1 INTRODUCTION

At the end of 2019 and the beginning of 2020, the COVID pandemic broke out, and this event happened to totally change the worldwide trends. Many traditional businesses were in risk of closing for an uncertainty of time. Retailers were forced to close their businesses or limited social activities, such as letting customers wait in queue with a certain distance (Kim, 2020). Therefore, to tackle this issue, businesses decided to switch to online platforms to meet the customers' needs of shopping. For that, buyers had to change their contact shopping habit to online shopping, which resulted in an increase of 26.4% to 4.248 trillion dollars for global retailers in 2020 (Shaw, Eschenbrenner & Baier, 2022).

The objective of the thesis is to create a responsive e-commerce application selling musical instruments on the client side using the React Bootstrap CSS framework and React UI library. This website aims to deliver a user-friendly interface to end users. Once the application has been finished, it will be hosted on a domain, solving the problem for businesses and customers. After the project has been launched, customers who live in far-away areas can purchase goods directly without reaching the store in person.

The thesis is divided into three different sections. The first part will be the introduction of the context, purpose of this thesis and application construction. The second part covers the theoretical framework, which includes the technologies implemented with version controls tools to handle the source code. The third part is going to depict the implementation plan along with the code illustration, thus make the idea of building musical instrument application be transparent. The final product will be deployed on the client-side without backend system and database server. In the end, further development with application, including improving performance, backend development, and database server hosting topics will be discussed after the project work has been done.

2 WEB DEVELOPMENT TECHNOLOGIES AND TOOLS

At the moment, there are various methods to tackle the solution of web development. There are regular updates of frameworks and libraries published and modified to allow the automation for the process of creating web applications. Thus, this advantage saves a lot of time for the application development (Dzhangarov, Pakhaev, Potapova, 2021). However, choosing the suitable stack could be difficult and it either leads to good or bad results. Moreover, to handle the application with the source code, there needs to have software development tools to do so. Technologies and tools are subjects that will be taken into consideration for the application development in the second part of the thesis.

2.1 HTML & CSS

HTML is a language defining the meaning and structure of web applications (Mozilla, 2024). It is developed to facilitate the definition, presentation, and processing of information on the user interface (Ranjan, Sinha & Battewad, 2020). In addition, HTML is used to add formatting and organisational information to improve the display of content on the UI in any forms, such as images, videos, texts (DuRocher, 2021). In HTML, to display a simple text on the interface, the content will be wrapped within an element with encased in angle brackets opening and closing tags with slash (DuRocher, 2021). For special cases, some elements do not have contents and no closing tag and they are considered void elements (Mozilla, 2024).

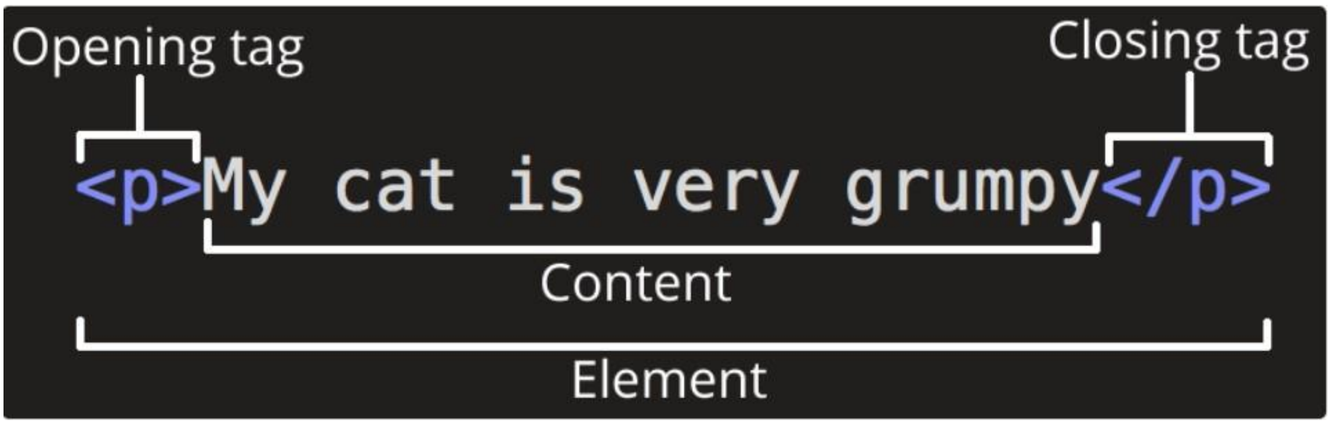
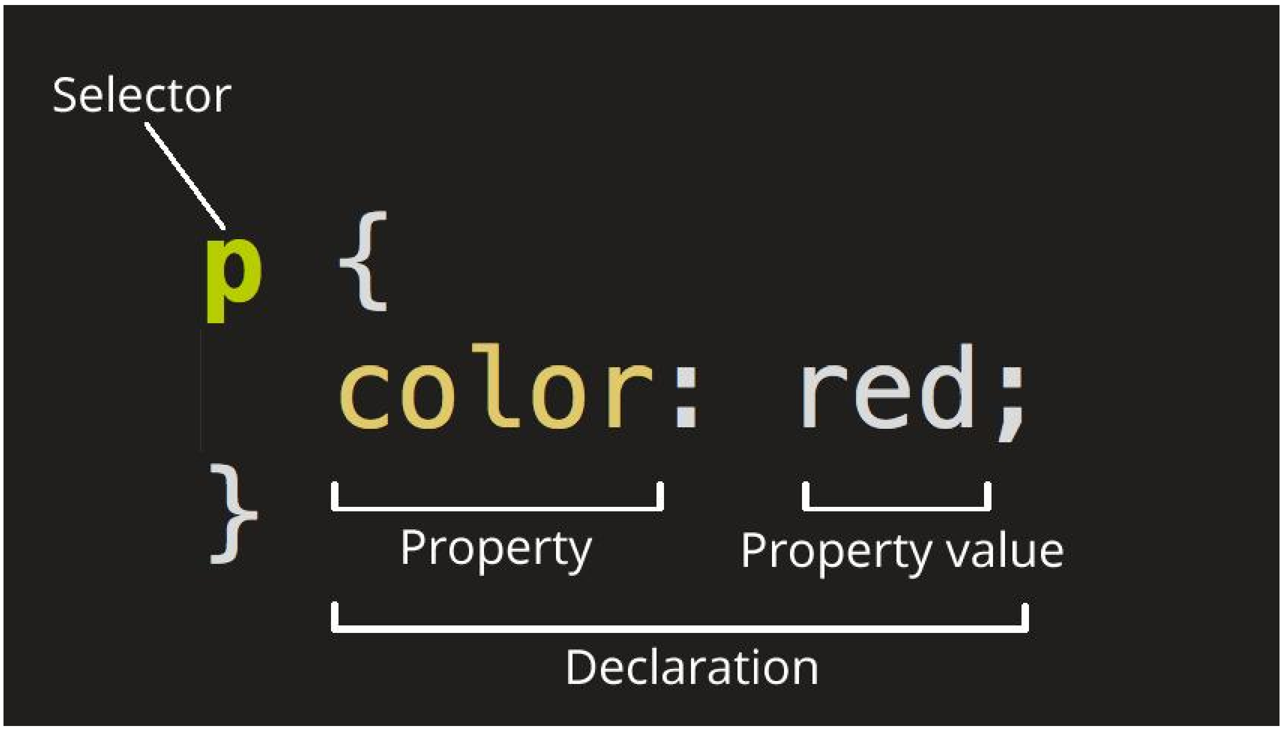


FIGURE 1. HTML sample paragraph element. (Mozilla, 2024)

In addition to HTML, CSS styles and layouts making the web application colourful, visually appealing, and usable (Mozilla, 2024). Formats, colours, and navigation bars are created with styles by the utilisation of CSS language so as to make the users look and feel the web interactive and friendly (DuRocher, 2021). There are three methods to use CSS in a web application. The first way to do it is adding inline inside the opening tag in HTML by adding style attribute and custom, but the inline style is not highly recommended due to its scalability, maintainability and make the HTML document structure unorganised (freeCodeCamp, 2022). The second method is internal style by creating style tag in the head section (freeCodeCamp, 2022). The current application focuses on using external CSS separation based on single responsibility principle in order to lessen the code inside each module with readability and flexibility (jain, 2017).



```
p {
color: red;
}
```

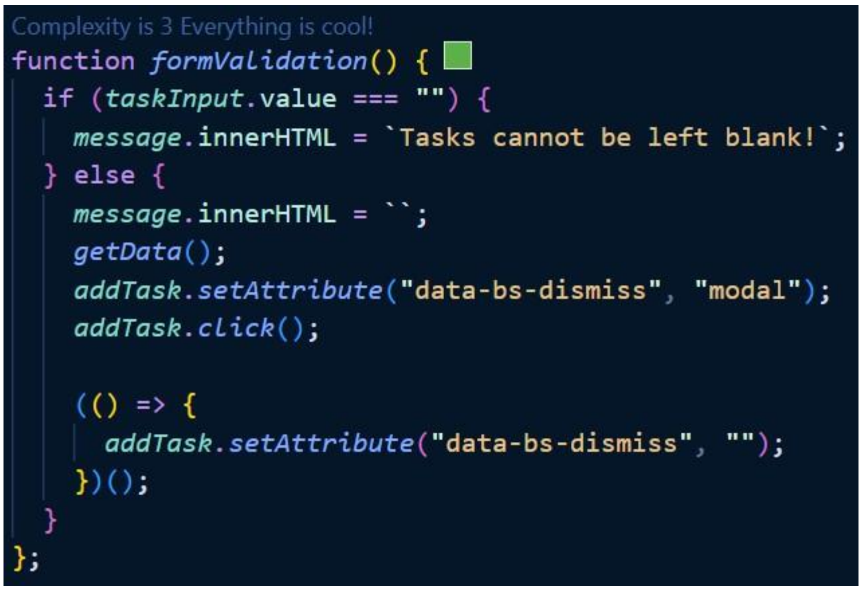
FIGURE 2. A sample CSS code (Mozilla, 2024)

2.2 JavaScript

The website with the display with lively effects are not enough. It needs to have dynamic interaction via mouse clicks. When a website needs to handle form submission, dynamic styling, and information searching, and JavaScript is implemented to handle the events. JavaScript is a client-side language designed to add interactivity into the website. The language is also known dynamic, loose typed compared to TypeScript, which is a strict type language (Robbins, 2018).

As in CODE 1, the function is implemented in JavaScript language to process with form validation. The function in the photo is checking the strict equality condition of taskInput value whether it is an empty task and the message returned the task should not be left blank or moving on with the else condition by retrieving data from getData function of user's tasks. Then the function sets the attribute of "data-bs-dismiss" to "modal" because the user is going to encounter the modal dialog to add the to-do tasks.

JavaScript core is also known as ECMAScript in term of standardisation; however, JavaScript and ECMAScript terminologies can be used interchangeably (Mozilla, 2024). The current version of ECMAScript is the 6th edition, which can be also shortened as ES6.

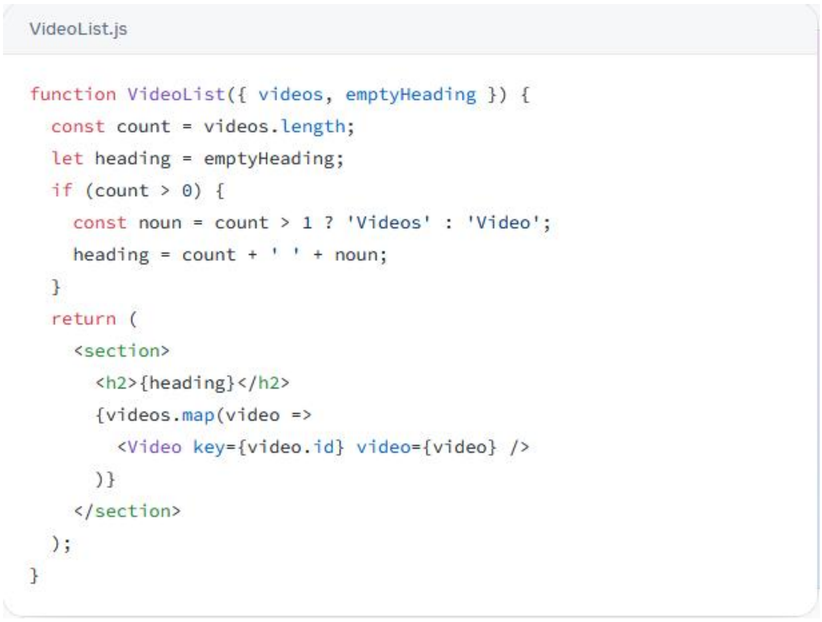


```
Complexity is 3 Everything is cool!
function formValidation() {
  if (taskInput.value === "") {
    message.innerHTML = `Tasks cannot be left blank!`;
  } else {
    message.innerHTML = ``;
    getData();
    addTask.setAttribute("data-bs-dismiss", "modal");
    addTask.click();

    (() => {
      addTask.setAttribute("data-bs-dismiss", "");
    })();
  }
};
```

CODE 1. JavaScript sample code (Shaheb, J., 2022)

2.3 ReactJS

ReactJS, or React, is an open-source UI library developed by Facebook (Rawat, Mahajan, 2020). React is designed to resolve the problem on the View in MVC model, particularly UI rendering and event responses. The library fully renders the application on the UI with simplicity of interaction has been made to avoid subsequent workload from complex mouse clicks (Sonpatki, 2016). For better efficiency, ReactJS makes use of virtual DOM with customised event handling in order to enhance the user experience on cross browsers. CODE 2 depicts a sample code written in JavaScript-based React module that passes two parameters of videos and emptyHeading. The heading renders the number of videos with noun after the if-else condition has checked if the number of videos is greater than one or not. On the screen, the heading is rendered to display the number of videos. In addition, the videos loop through map function that invokes the Video module with sample data containing unique id and the video playlist.



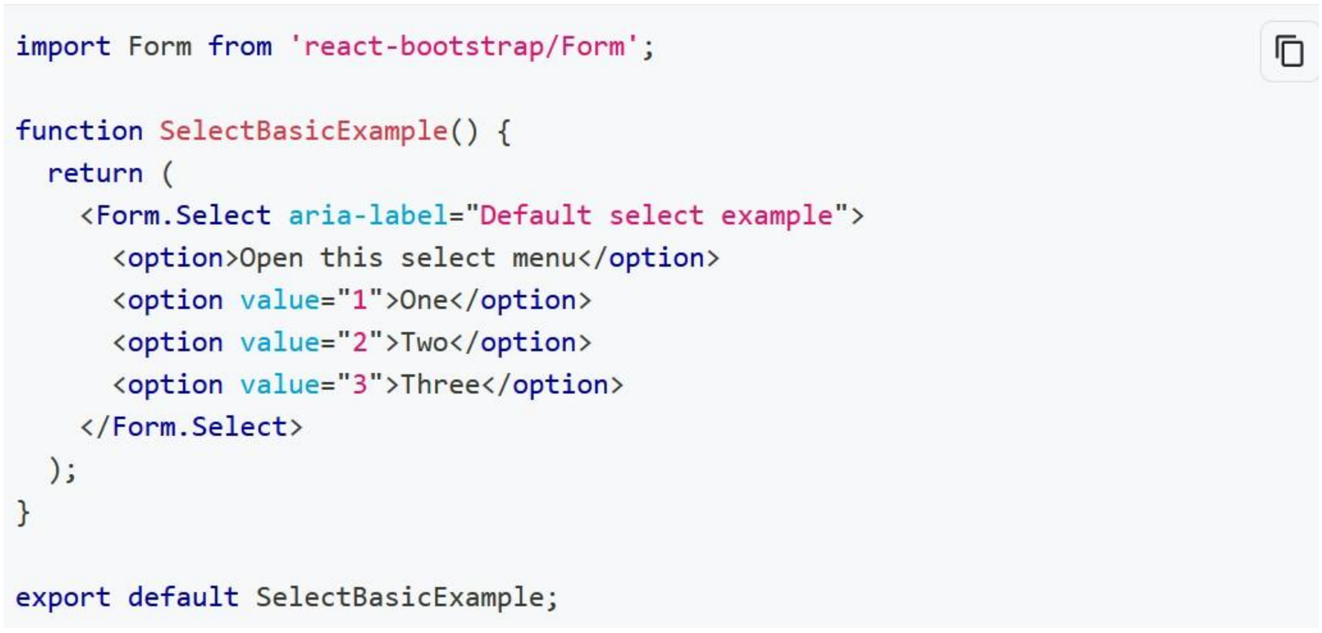
VideoList.js

```
function VideoList({ videos, emptyHeading }) {
  const count = videos.length;
  let heading = emptyHeading;
  if (count > 0) {
    const noun = count > 1 ? 'Videos' : 'Video';
    heading = count + ' ' + noun;
  }
  return (
    <section>
      <h2>{heading}</h2>
      {videos.map(video =>
        <Video key={video.id} video={video} />
      )}
    </section>
  );
}
```

CODE 2. React sample code (react.dev, 2024)

2.4 React Bootstrap

React-Bootstrap is a CSS library toolbox that utilises the front-end components for developers for reusability prospect (Rawat, Mahajan, 2020). The library includes accessible front-end elements, for example, cards, forms, navigation bars, buttons (GeeksForGeeks, 2024). With provided components and functionalities, developers simply make some modifications, and they do not have to consume a lot of time building a component from scratch. Given is CODE 3 that shows the implementation of React Bootstrap library with Form module to display a select dropdown list of three values including one, two, and three with the content is wrapped in between the option tags.

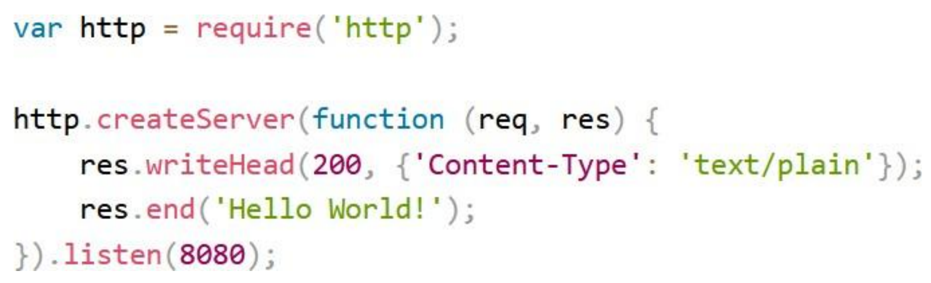


```
import Form from 'react-bootstrap/Form';

function SelectBasicExample() {
  return (
    <Form.Select aria-label="Default select example">
      <option>Open this select menu</option>
      <option value="1">One</option>
      <option value="2">Two</option>
      <option value="3">Three</option>
    </Form.Select>
  );
}

export default SelectBasicExample;
```

CODE 3. React-Bootstrap select component sample code (React-Bootstrap, 2024)

2.5 Node

Node, or Node.js, is a lightweight, cross-platform, and efficient JavaScript runtime environment built on V8 engine of Google Chrome, which can be useful for servers and desktop applications. For web servers, maintaining the performance, scalability, latency, and throughput are core values, and it is not easy to handle the low latency as usual while scaling up the size of the input data (Heller, 2019). In the given CODE 4, Node.js is being implemented on the backend side with http package. Then, http invokes the createServer function to host the localhost with the port number 8080 displaying "Hello World!" content. However, in the project, Node is utilised as an environment to install React library and packages for the development.



```
var http = require('http');

http.createServer(function (req, res) {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Hello World!');
}).listen(8080);
```

CODE 4 – Sample Node.js code printing Hello World (W3School, 2024)

2.6 Git

Git is a distributed version control system, created by Linus Torvalds in 2005 (Microsoft, 2022).For software engineering, Git is used to manage the source code by committing, tracking changes and who contributes to the change (W3Schools, 2024). Via Git, software developers can be able to commit work on their local device, and then synchronise the project work with the copy on the server for recyclingpurpose (Microsoft, 2022). Developers utilise Git to manage the project repositories on any source code storing platforms, like GitHub or GitLab. To push the code, developers can make Staging and Committing processes, then push and finally update the code. Furthermore, a project can be pulled withthe latest version to the local copy (W3Schools, 2024).

Developers can fully utilise simultaneous development with theproject work on the branch management. Git also dedicates to more instant product release with the stable and high-quality code in the main branch on which the developers are working. And it is stated that the code management and updates can be straightforward after all. One of the most crucial advantages of Git is its integration in IDEs help lessen the initialising, staging, committing, and cloning phases. In Visual Studio Code, for example, there is a Source Control category which manages the codechange, track, stage, and commit. For that, developers do not have to spend time typing commands in the console to manage to push the code to source code platforms. (Microsoft, 2022). In FIGURE 3, App.js and index.js modules are going to be uploaded to the GitHub platform after being staged, and they are going to be committed with a message for the release of the product.

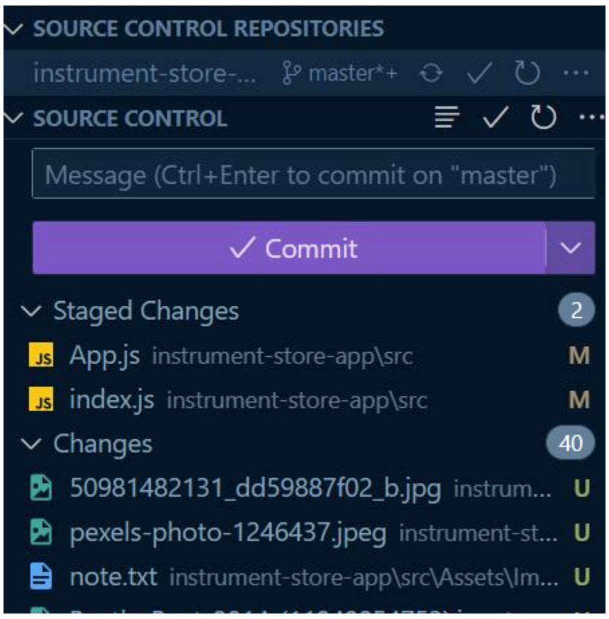

SOURCE CONTROL REPOSITORIES
instrument-store-...
master*+
SOURCE CONTROL
Message (Ctrl+Enter to commit on "master")
Commit
Staged Changes 2
App.js instrument-store-app\src M
index.js instrument-store-app\src M
Changes 40
50981482131_dd59887f02_b.jpg instrum... U
pexels-photo-1246437.jpeg instrument-st... U
note.txt instrument-store-app\src\Assets\Im... U

FIGURE 3. Git integration in Visual Studio Code

Git has a strong community support (Microsoft, 2022). It does not matter where the developer comes from, the project can still be contributed in many different ways and everywhere (W3Schools, 2024). In addition, Git can be utilised as a source code management tool with the purpose of boosting the efficiency of collaboration, automation processes, and traceability with visualisation improvement of the work. GitHub, GitLab, or Azure DevOps are thesolutions that meet the demand of working, committing, and resolving problems in one place (Microsoft, 2022).

2.7 GitHub

GitHub is a code hosting platforms for developers to store the code and the others can contribute to make changes to the repositories (W3Schools, 2024). GitHub is utilised to create, update, store, merge, pull, and contribute on the other repositories. Also, there is always a discussion and GitHub community is going to answer the problem raised, and it is considered social coding stage for developers (Coursera, 2023).

GitHub platform has over 4 million organisations and more than a hundred million of developers who are using GitHub as SaaS. GitHub is straightforward and has user-friendly interface. In addition, developers are able to access to public repositories. In order to support the quality control, GitHub implementssome automation tools for more boring tasks or product, for example, unit testing (Coursera, 2023).

2.8 Visual Studio Code

To develop the application, developers can either write the code on IDE or text editor. In this project, Visual Studio Code is going to be used for the application development as it is an open-source, lightweight,but powerful code editor that can be run on any platforms. The code editor includes the extension and support for a variety of programming languages, for example, JavaScript, Java, and Python (Visual Studio Code, 2024).

Therefore, this application will be implemented with Visual Studio Code environment and required programming languages with source code management tools, including HTML, CSS, JavaScript, React, Bootstrap, Node, and Git, respectively.

3 IMPLEMENTATION PLAN

Having selected the technology stack for the application development, the subsequent phase is to give the project an implementation scheme in details to keep track with the development process. In the plan, the process focuses on outlining the general design of the application and creating visual sitemap to help the user navigate the services and products of their needs. In addition, the plan will also demo a use-case diagram illustrating customer and shop owner interactions. Thus, making the idea of the application become more transparent regarding the functionality. Also, the project files and structures containing source code have to be well-structured for work efficiency and better navigation to save the developer's time to search for modules they work on.

3.1 Frontend design and web structure

In this application, frontend development is mainly focused throughout the process. The main page of the musical instrument website will contain three main sections, including header, footer, and body. The header contains the brand which will be laid at the centre, and it is a hyperlink forwarding to the home page itself. In addition, the header also stickswith the navigation bar with different items redirecting to pages of blogs, products, about, contact, and news pages. In the body section, there will be a carousel hero section displaying the advertisement to the web visitors. Consecutively, there are automatically displaying slides of various products, such as guitar, piano, ukulele, organ, and drum and users can click in the carousel photo to forward to a specific product page. The footer is laid at the end of the application used to display the information and contact details of location, phone number, and email. Furthermore, responsive design using media query in CSS is necessary for different UI scopes. FIGURE 4 will indicate the overall design of the main page towards the description.

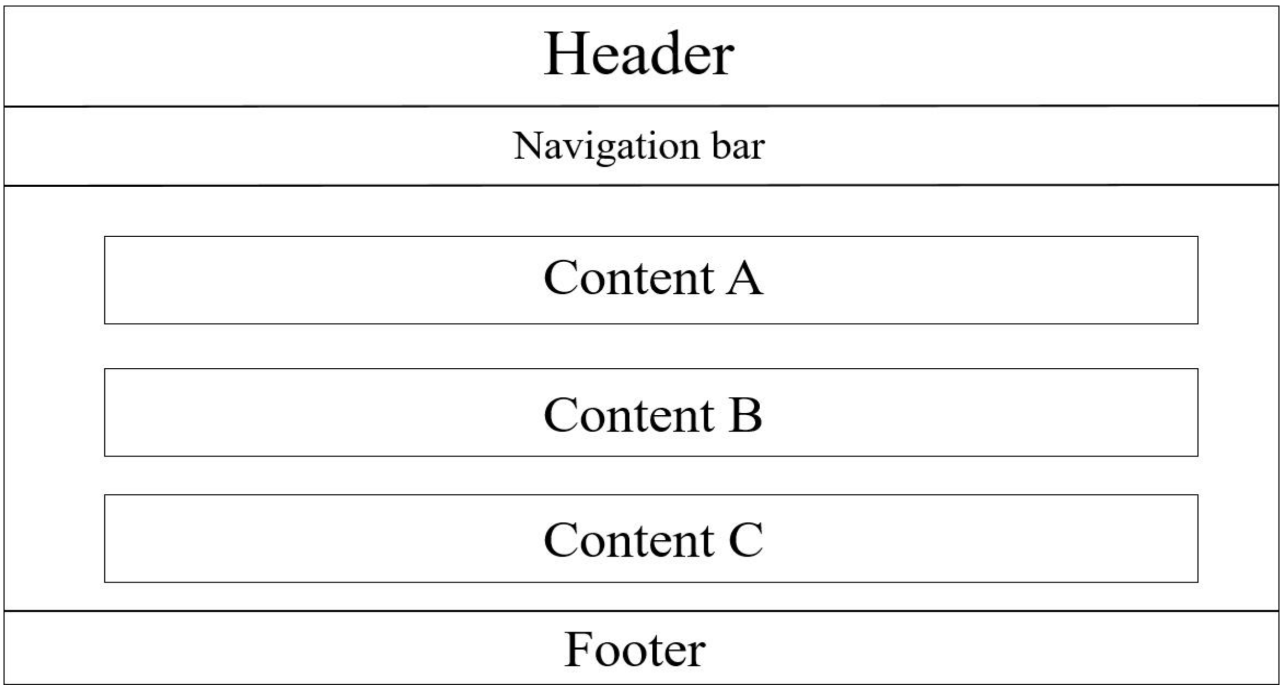


FIGURE 4. Web frontend structure (model adapted from piyushpilaniya98, 2023)

In order for visitors to know where they are during the website visit, it is essential to organise the information into a brief visual sitemap so that the user is able to navigate and find specific information (Chaparro, Bernard, 1999). In this musical instrument application, there are six main pages which are displayed explicitly as items on the navigation bar comprising of Home, About, Products, Contact, Blog, and News. For Products item, users will hover to show a dropdown list of musical instruments which includes drum, guitar, organ, ukulele, and drum.

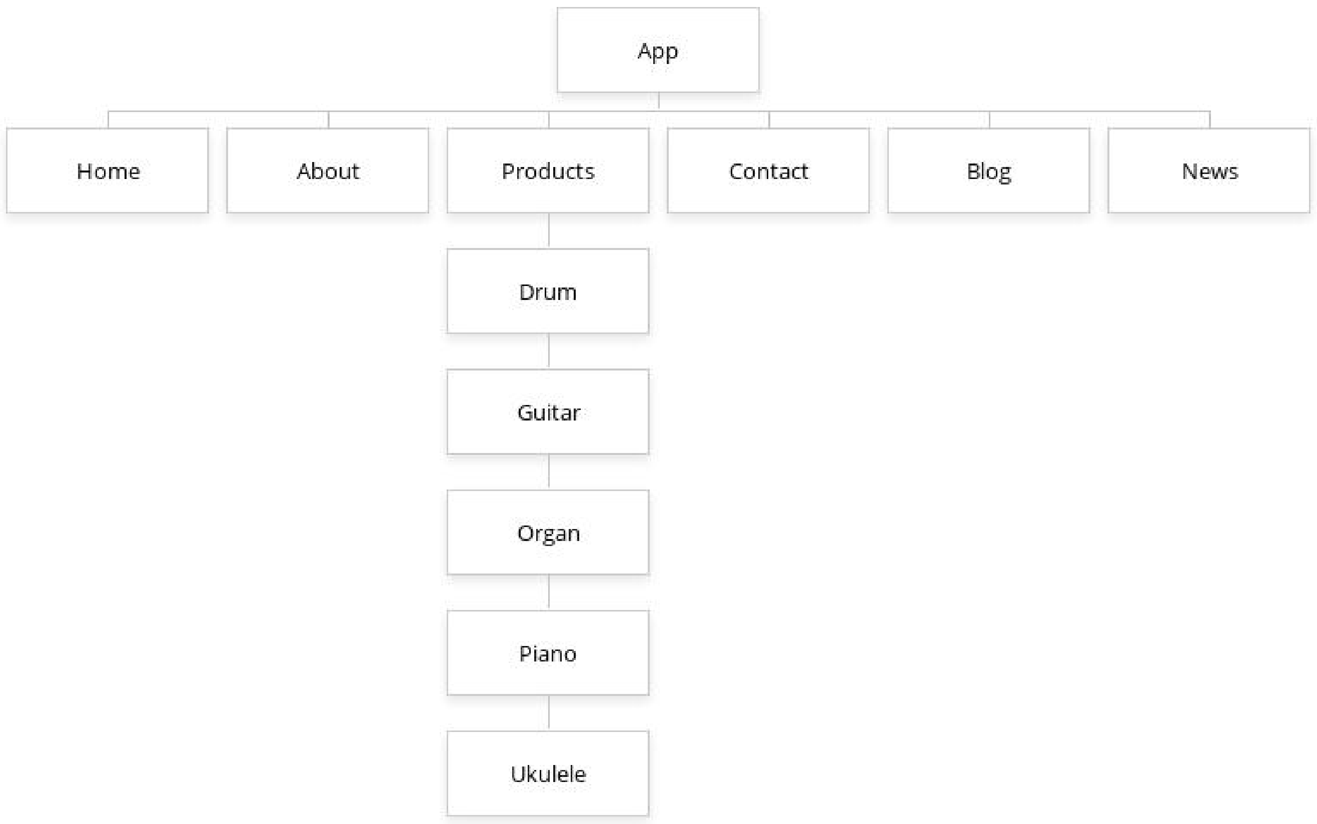


FIGURE 5. Visual sitemap of the musical instrument application (model adapted from Chaparro, Bernard, 1999)

3.2 Planning

It is essential to understand how the application is going to be erected. One ofthe approaches to solve the problem is to create a diagram to have a right track on developing the application which makes it be straightforward, maintainable, and reusable for the future development (Franklin, Planning Your Web Site With UML, 2001). A diagram flow helps the developer to keep pace with the track to progress with the application. During the developing process, there might be someminor changes in order meet the end's meet of users. The diagram is going to indicate two main actors with methods defined, and the flow will mostly focus on the user because the application is developed only on the client-side, and there is not much of interference from the administrator who manages the order from clients. FIGURE 6 depicts the flow of user interactivity with the application and thestore's owner management of the orders.

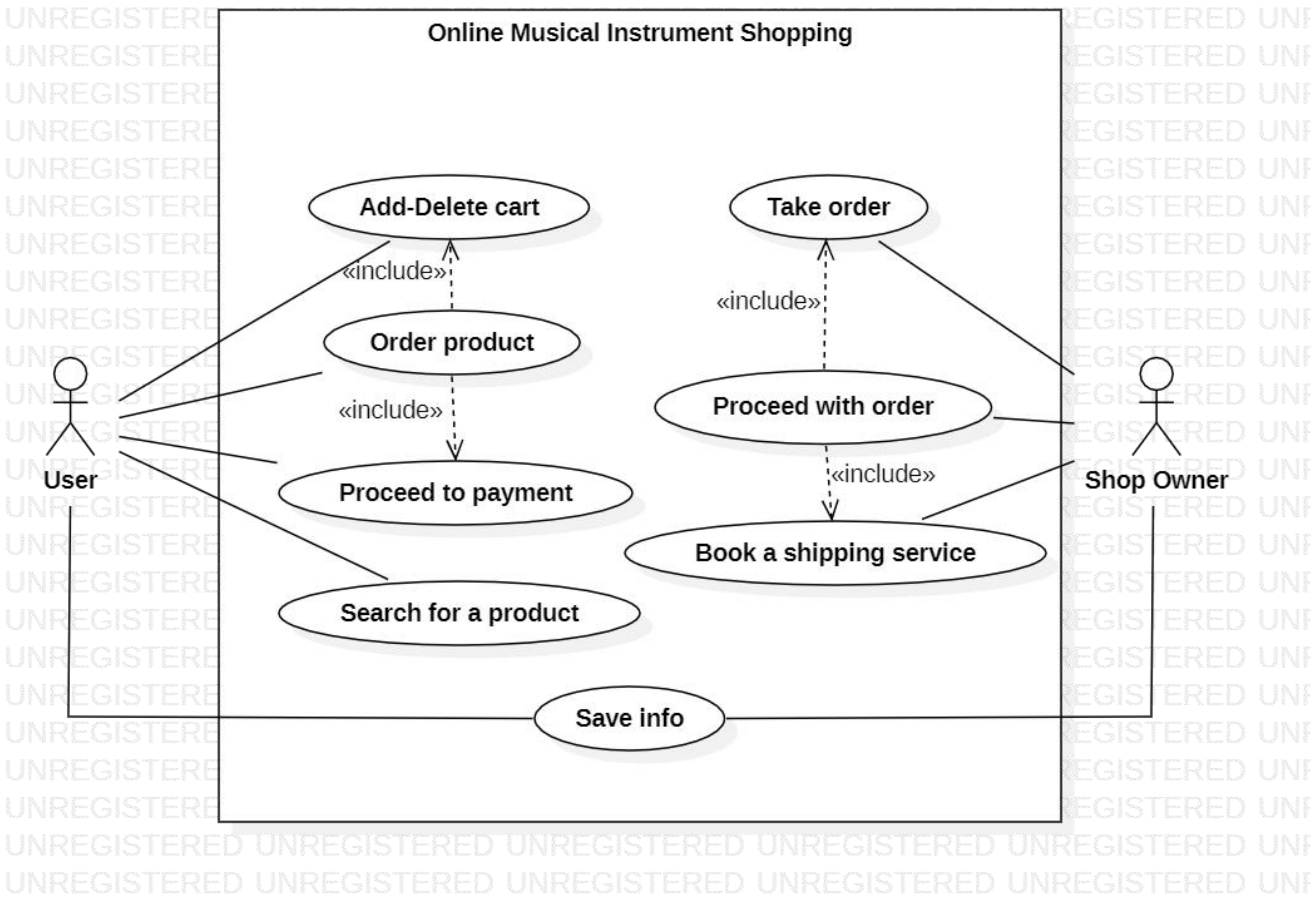
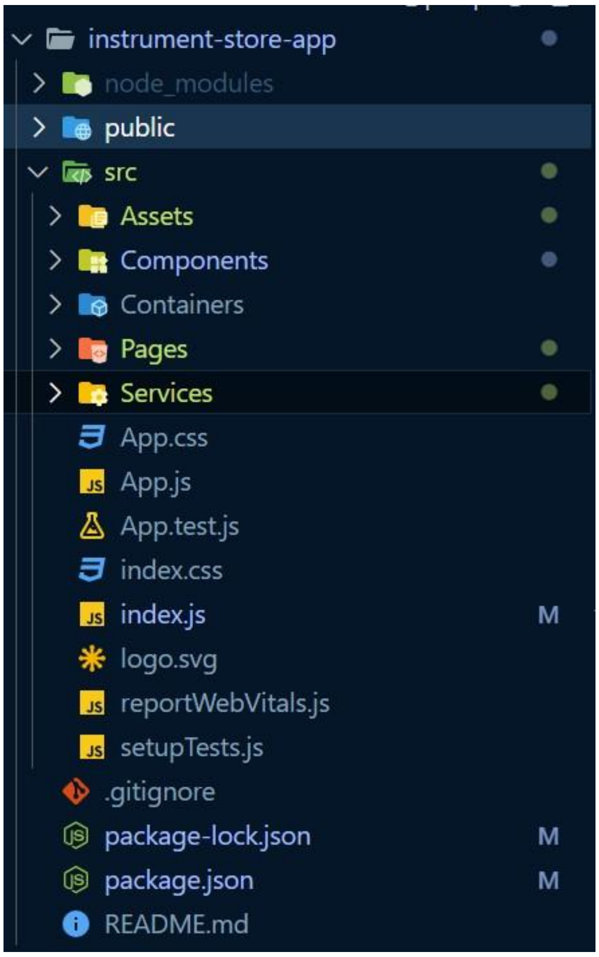


FIGURE 6. Use case diagram of the application

3.3 Project file structure

To make a project become organisational and readable to the developer and other collaborators, files and folders have to be structured in a logical order for revisit, revision, and development (Chien, D., 2023). Having a structured folder creates advantage to locate the iles and work on with great efficiency. In addition, organising the files increases the work productivity, shareability, and consistency.



```
instrument-store-app
  node_modules
  public
  src
    Assets
    Components
    Containers
    Pages
    Services
    App.css
    App.js
    App.test.js
    index.css
    index.js                M
    logo.svg
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json         M
  package.json              M
  README.md
```

FIGURE 7. Project file structure of the application

In the project file as shown in FIGURE 8, “public” folder contains static files which are explicitly displayed on the UI, including HTML index file, logos, icon, and manifest file of JSON format. Most of work will be concentrated in the “src” folder where the source code is stored and managed for the application lives, consisting of JavaScript-based React modules with native CSS and other assets. In src folder, “Assets” folder is used to store the images for the carousels and products. The “Components” folder contains common and reusable pieces of code that makes up a web application without creating a bunch of redundant code. In addition,creating “Components” folder separates and makes the code become organisational, manageable that helps the developer track the modules and make correct imports in the project. The “Pages” folder contains all pages of the application that are displayed directly on the UI. Thefile App serves as the main component which will contain the all imported modules. The JavaScript index file serves as the entrypoint of the application which renders the root component binding with the HTML file (Kods, 2023). Finally, package.json and package-lock.json are two important files which are laid at the root of the project folder. These two files handle the dependencies, metadata, scripts with several configurations.

4 PROJECT IMPLEMENTATION

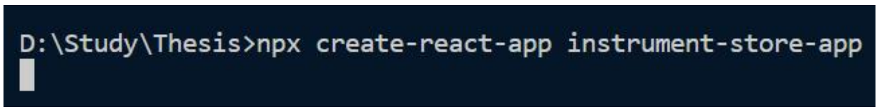
With the choice of technology stack and the planning phase, chapter 4 will focus on the implementation of the application. The application will be developed only on the client-side with ReactJS, Bootstrap with integrated external packages to support the application development to deliver the user-friendly interface and responsive e-commerce platform to users. The subject of the implementation phase starts from the environment setup phase to developing the application pages with explanation providing the service of user’s needs.

4.1 Environment setup

Before building up an application, it is important to choose the appropriate software development tools with environment setup in order to manage and build the application. Visual Studio Code is a lightweight text editor for the application development built with modern interface and a variety of extensions supporting developing applications. In addition, Visual Studio Code integrates Git version control to track, change, and commit the code to code hosting platforms like GitHub or GitLab. To work with commands, terminal is added that works from the root of the project, and developers are able to handle the project with Git commands.

Before creating a React project, some specific extensions need to be installed to support the project, consisting of JavaScript ES6 code snippets, code formatter, ESLint, and HTML CSS Support. ESLint is the most powerful extension in Visual Studio Code which can be used to detect the errors and forcing to standardise the code to improve the code quality. Furthermore, ESLint can help fix the common errors and replace with best practices while coding, and most importantly, the code is maintainable and consistent (Jacob, freeCodeCamp, 2023).

To setup a React environment, Node has to be installed in advance in order to create a React application, and npm is installed with Node to carry, manage, and update the dependencies that the React project is in need. In this project, React is installed in Visual Studio Code via terminal with the command as indicated in FIGURE 8.



```
D:\Study\Thesis>npx create-react-app instrument-store-app
```

FIGURE 8. Installing React project

The npx create-react-app app-name is the most well-known syntax to install a React project. After installing React, developer is instructed to forward to the root folder to start the application with “npm start” command, and given is FIGURE 9 shows the default home page of React application after launching with the following command.

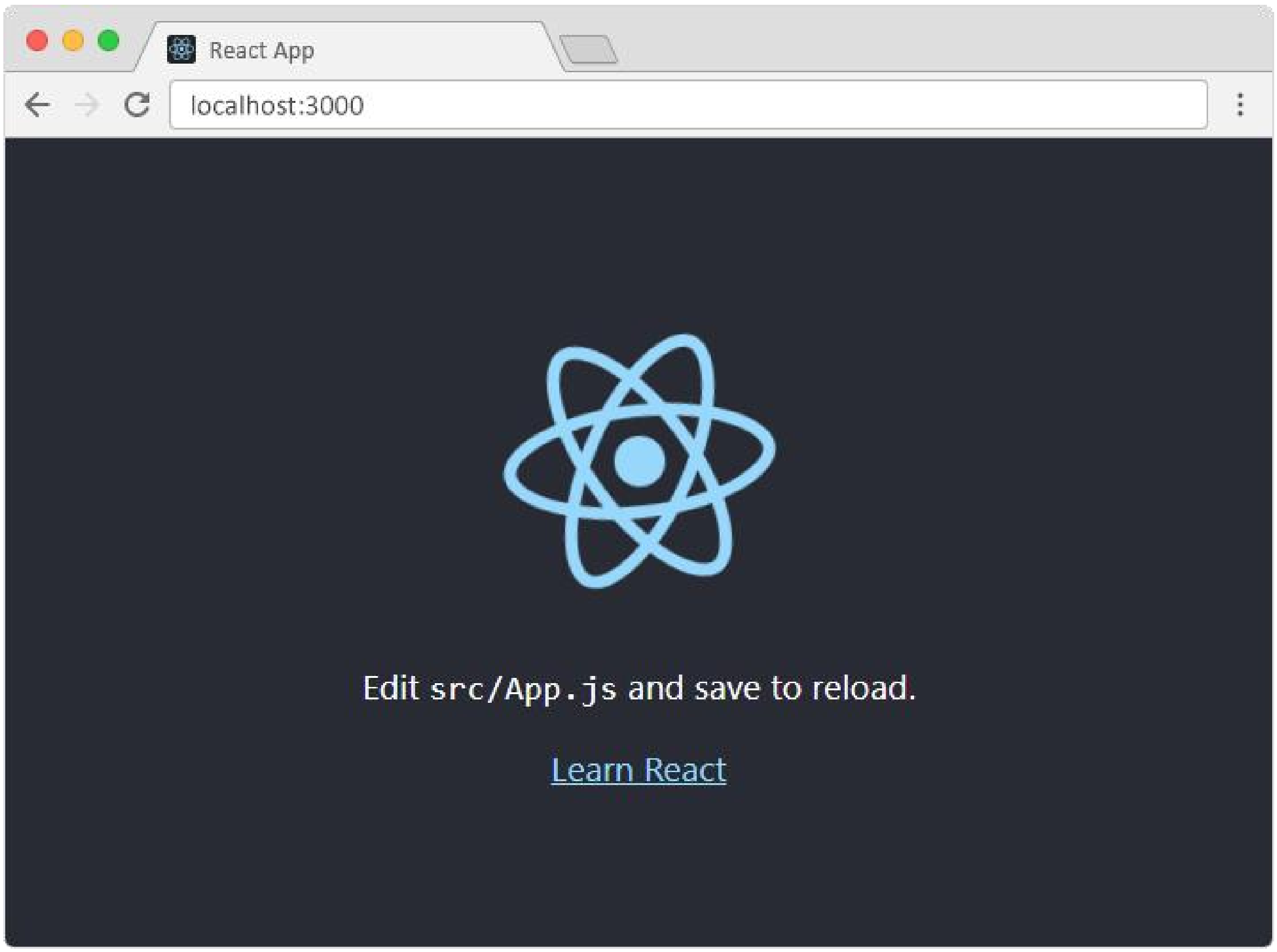
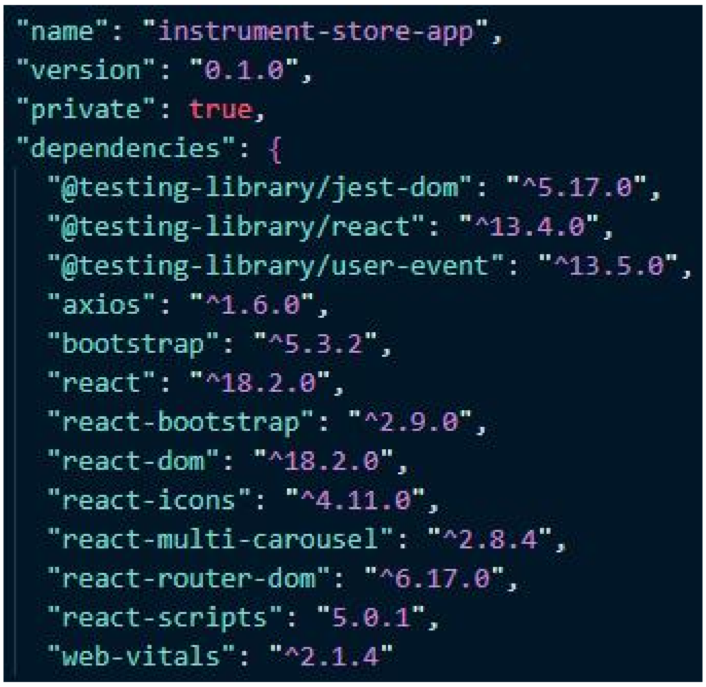


FIGURE 9. React home page (W3Schools, React Get Started, 2024)

To develop the application with more features, different packages are installed to serve the need and boost the development process. As in FIGURE 10, the photo is depicting the packages prepared for the application. “axios” is a JavaScript library used to make HTTP requests from Node to fetch the API data with simple operations, for example, GET, POST, and DELETE (Rawat, Mahajan, 2020). In addition, “react-icons” package includes a set of commonly used icons making incorporation of ES6 to import in React projects (npmjs.com, 2024). React Router DOM package helps increase the dynamic routing feature and easily navigate to different pages. By utilising React Router, a page reloads efficiently, faster than the traditional navigation, and there no refresh is needed after forwarding from page to page. Thus, user experience and web performance are improved significantly (salehmubashar, 2023).



```json
"name": "instrument-store-app",
"version": "0.1.0",
"private": true,
"dependencies": {
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "axios": "^1.6.0",
  "bootstrap": "^5.3.2",
  "react": "^18.2.0",
  "react-bootstrap": "^2.9.0",
  "react-dom": "^18.2.0",
  "react-icons": "^4.11.0",
  "react-multi-carousel": "^2.8.4",
  "react-router-dom": "^6.17.0",
  "react-scripts": "5.0.1",
  "web-vitals": "^2.1.4"
```

FIGURE 10. Packages installed for the project.

4.2 Website implementation

After the extensions and external packages to support to boost the application development, the next stage is to develop the application in practice based on the planning with the general UI/UX design with the support of some external libraries of APIs, routing, and CSS. In the implementation section, the key point is to emphasise the development process on each aspect, including responsiveness on the client-side and API resource utilisation. Most importantly, the CRUD functionalities on the music instrument store will be implemented with provided libraries from ReactJS that handles the shopping activities of users visiting the online store.

4.2.1 Responsiveness on navigation bar and footer

The navigation bar includes five different items that guides the user in between pages in a single page application to About, Product, News, Blogs, and Contact pages. Navigation bar and footer are two components that have different styles on small and large scale devices. The application design is based on Bootstrap grid system divided into six breakpoints on which the components can be modified to fit the device scale.

On mobile devices, the navigation bar turns into a hamburger button with collapsible feature. When the button is clicked, the navigation bar will drop down and display all of the items on the screen. On the contrary, the bar is fully displayed on bigger devices. For the Products item, user can hover to display product list dropdown comprising of Guitar, Piano, Ukulele, Drum, and Organ items. If one of the products is selected, the user will be forwarded to product list page of that specific musical instrument.

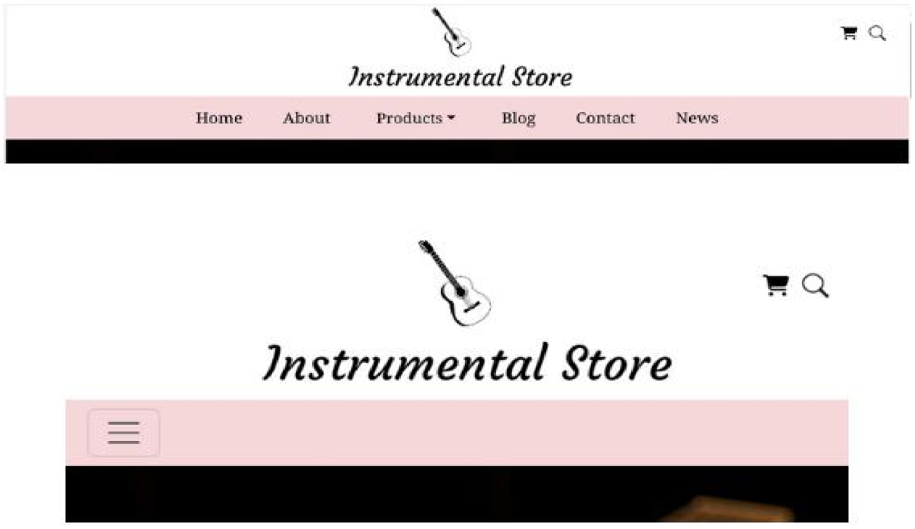
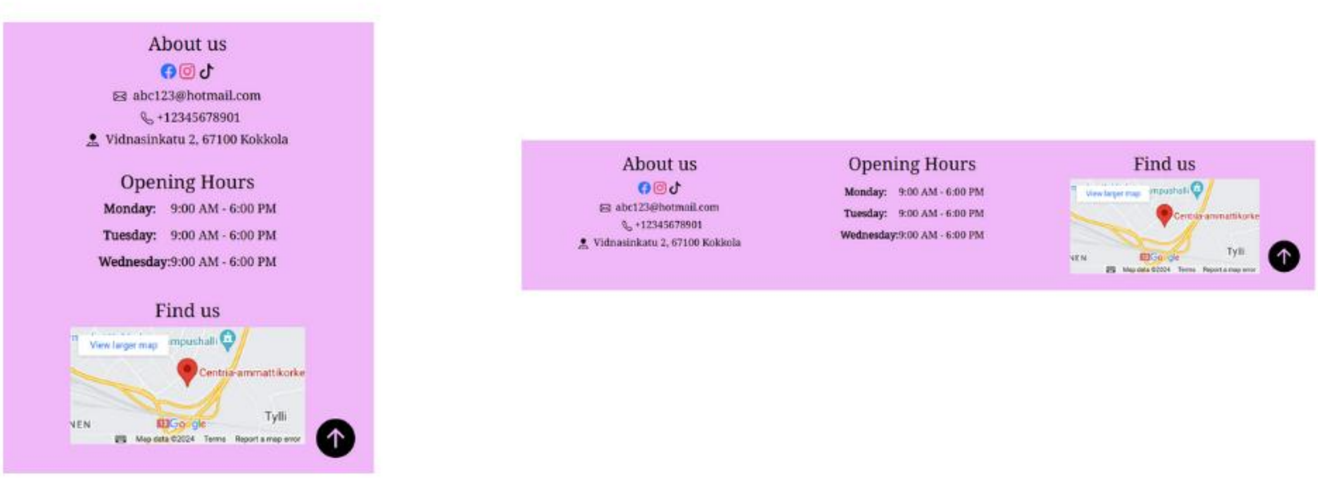


FIGURE 11. Header and navigation bar on large and small perspectives

On the other hand, footer includes all the information about the shop, particularly contact methods, opening hours, and location provision. In the footer section, flexbox feature is implemented to divide the content into three different sections. Flexbox is utilised by using built-in classes of Bootstrap, which saves time to create flexbox classes from scratch without knowing the size. In FIGURE 14, on mobile device, the content is arranged as a block sequentially from top to bottom. For desktop, the footer horizontally displays the content from left to right. In addition, Google Maps API is embedded in the footer for the customers to track the shop location with media queries implemented to fit the device scale.


About us

abc123@hotmail.com

+12345678901

Vidnasinkatu 2, 67100 Kokkola

Opening Hours

Monday: 9:00 AM - 6:00 PM

Tuesday: 9:00 AM - 6:00 PM

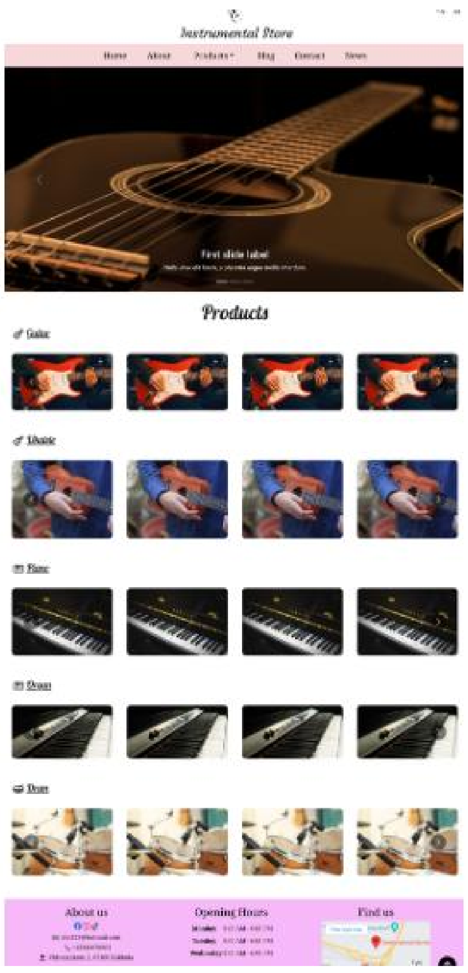
Wednesday: 9:00 AM - 6:00 PM

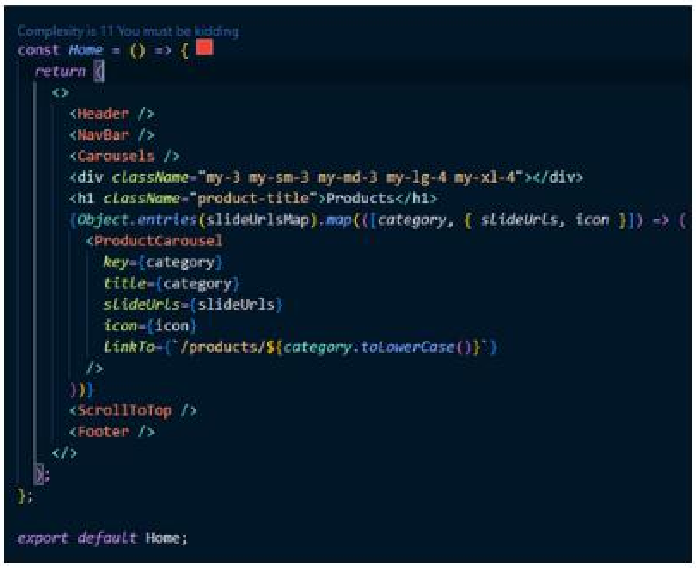
Find us

FIGURE 12. Footer on mobile and desktop devices

4.2.2 Home page

Home page is the main page that comes to the user when they make a visit to the website. The home page serves as the user's entry point of contact and provides general information about the musical instrument content. The content is organised from top to bottom into distinct sections, with a header, navigation bar, body featuring multiple carousels showcasing the products, and footer.





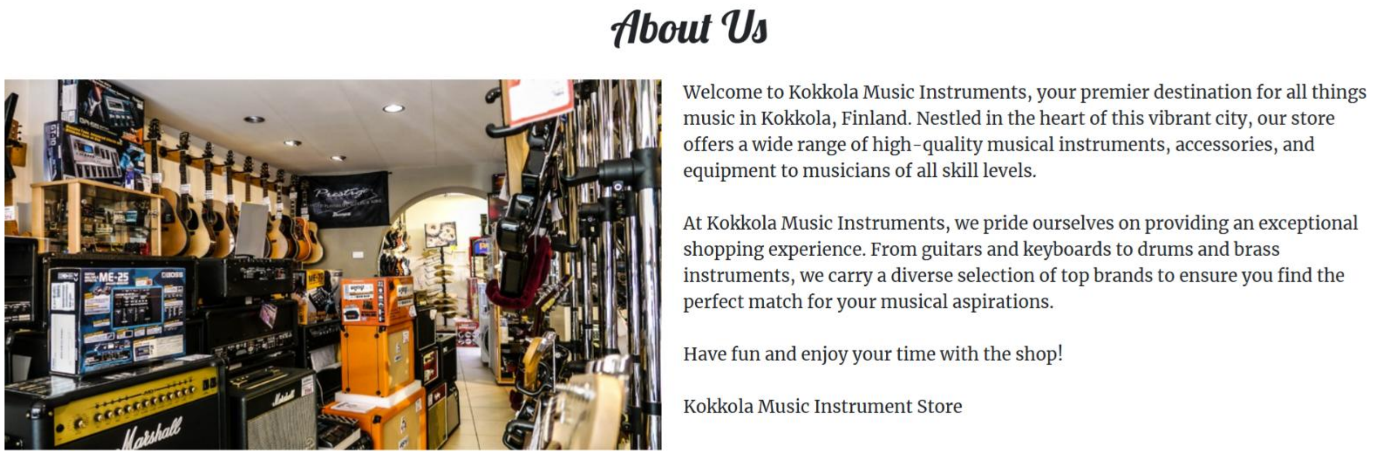
```
Complexity is 11 You must be kidding
const Home = () => {
  return (
    <>
      <Header />
      <NavBar />
      <Carousels />
      <div className="my-3 my-sm-3 my-md-3 my-lg-4 my-xl-4"></div>
      <h1 className="product-title">Products</h1>
      {Object.entries(slideUrlsMap).map(([category, { slideUrls, icon }]) => (
        <ProductCarousel
          key={category}
          title={category}
          slideUrls={slideUrls}
          icon={icon}
          linkTo={`/products/${category.toLowerCase()}`}
        />
      ))}
      <ScrollToTop />
      <Footer />
    </>
  );
};

export default Home;
```

FIGURE 13. Photo of home page

In the home page module, the header includes the store brand with searching and shopping cart buttons which are placed at the centre and top right, respectively. Sequentially, navigation bar is designed to let the user easily forward from this page to another page with react-router-dom library. The header and navigation bar are fixed components which will go anywhere the user scrolls to, and this helps the user navigates to different page without scrolling back to the top. The carousel plays as the hero section attracting the user with specific content. In the code provided in FIGURE 13, ProductCarousel component renders the categories and images from slideUrls. The component iterates over the music instrument categories with icons, titles representing the category. In addition, the user can either click on the title or directly to the slideUrl component to forward to that corresponding music instrument product list page. Furthermore, the component also implements the third-party library of "react-responsive-carousel" to manage to display the product carousels with autoplay, swipe gestures, and responsiveness over the devices.

4.2.3 About page

The About page delivers an introduction about the shop in details with the design is implemented with flexbox divided the page into two sections with one mock photo of the shop and the main content goes to the right side. On the mobile device, the content is displayed as a block from top to bottom. The About page contains a brief introduction with location information provided with the general purposes of the store with the promise of bringing the best experience while doing the shopping and future development of the store.


About Us

Welcome to Kokkola Music Instruments, your premier destination for all things music in Kokkola, Finland. Nestled in the heart of this vibrant city, our store offers a wide range of high-quality musical instruments, accessories, and equipment to musicians of all skill levels.

At Kokkola Music Instruments, we pride ourselves on providing an exceptional shopping experience. From guitars and keyboards to drums and brass instruments, we carry a diverse selection of top brands to ensure you find the perfect match for your musical aspirations.

Have fun and enjoy your time with the shop!

Kokkola Music Instrument Store

FIGURE 14. About page

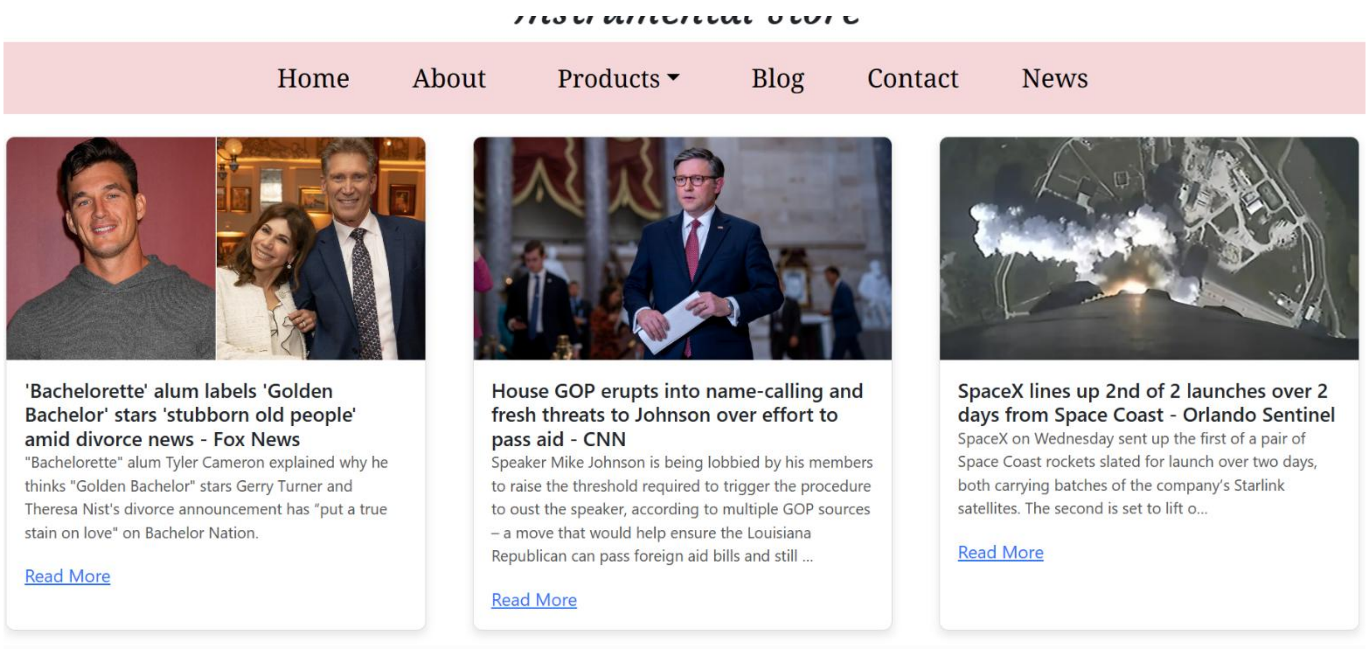
4.2.4 Blog page

The Blog module for the project serves as a vital component for providing the user with informative articles and updates relevant to the music topics. The module utilises axios and it dynamically fetches blog posts from an external API, enriching the store's online presence and offering valuable content to customers.

The Blogs component requests a complete set of blog entries from the JSONPlaceholder API via a GET request. The module at the initial state fetches the data to display the sample blog as a block of cards with short content. The fetched data is effectively handled within the component's state by utilising the useState hook, guaranteeing smooth user interface modifications each time a new post is published or edited. Each blog post is displayed as a card within a grid layout, facilitated by Row and Col components. The Post component, rendered within each Col, dynamically populates content such as the post title, body, and publication date.

4.2.5 News page

Beside reading blogs, user can pay a visit to the News page to catch up with the up-to-date daily news with the usage from News API fetched with instant updates of articles and breaking news headlines from news sources and blogs. To ensure the security of the application, the API and URL keys are separated solely to a file storing environment variables which is laid at the root of the project folder in order to secure the confidentiality and safety of information.



'Bachelorette' alum labels 'Golden Bachelor' stars 'stubborn old people' amid divorce news - Fox News
"Bachelorette" alum Tyler Cameron explained why he thinks "Golden Bachelor" stars Gerry Turner and Theresa Nist's divorce announcement has "put a true stain on love" on Bachelor Nation.

House GOP erupts into name-calling and fresh threats to Johnson over effort to pass aid - CNN
Speaker Mike Johnson is being lobbied by his members to raise the threshold required to trigger the procedure to oust the speaker, according to multiple GOP sources – a move that would help ensure the Louisiana Republican can pass foreign aid bills and still ...

SpaceX lines up 2nd of 2 launches over 2 days from Space Coast - Orlando Sentinel
SpaceX on Wednesday sent up the first of a pair of Space Coast rockets slated for launch over two days, both carrying batches of the company's Starlink satellites. The second is set to lift o...

FIGURE 15. News page

News page is implemented with flexbox design in order to keep the UI friendly and simple. Each news is wrapped inside a card that contains the photo URL, title, and shorter version of description about the articles. And the News module is designed to attract user's interaction via "Read More" button that redirects the user to a new window without closing the application. The fetched data is managed in the React hook which facilitates the real-time updates as new articles becomes available.

4.2.6 Contact page

The website also creates one more category for user to contact the business inquiries for the services and operations. For customers who cannot reach the store in contact, they can use the contact form to connect with the store's owner and leave a message with form submission method.

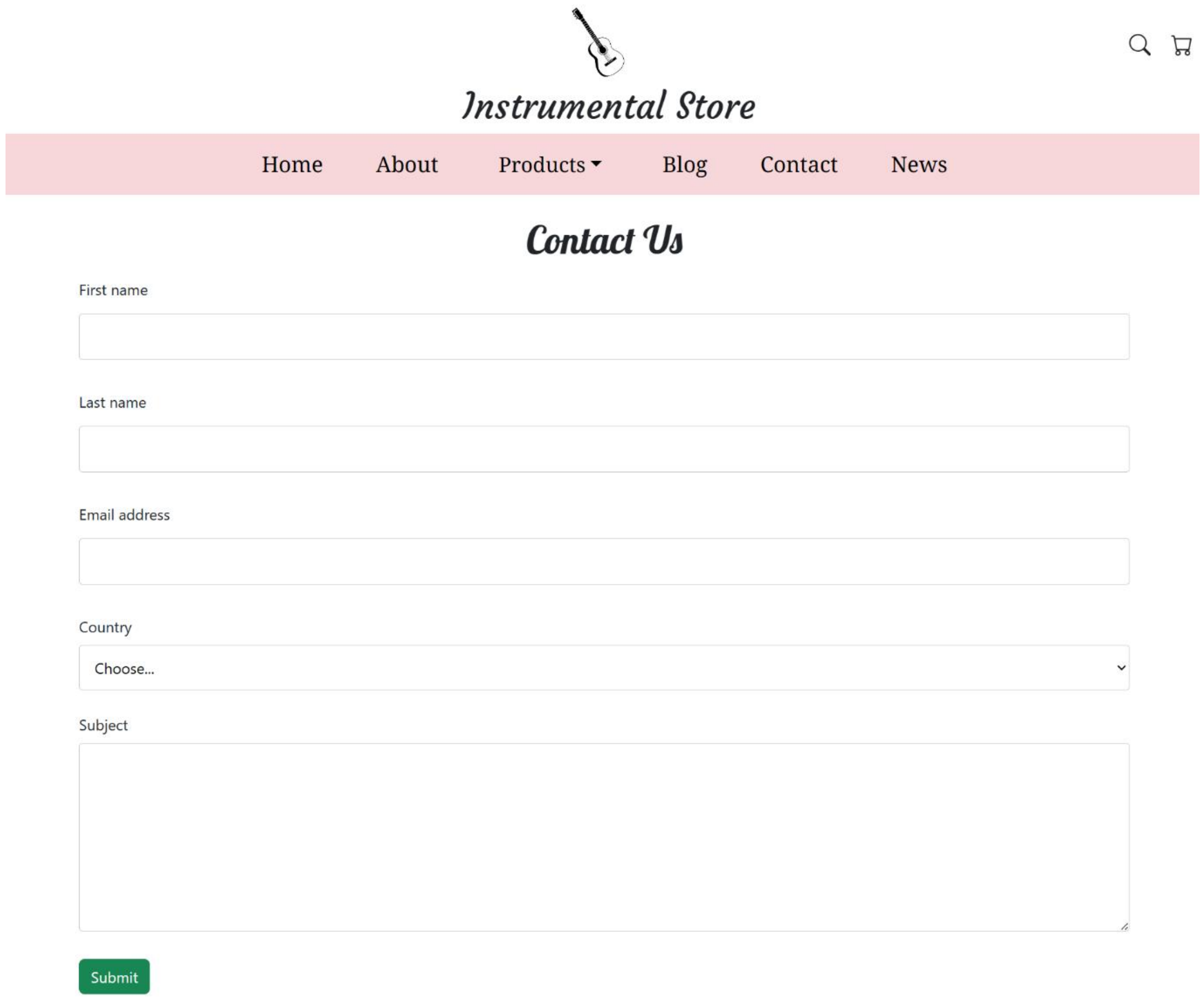
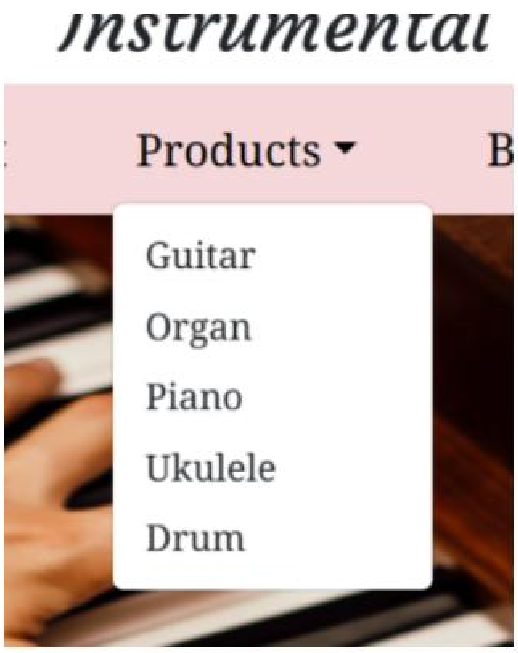


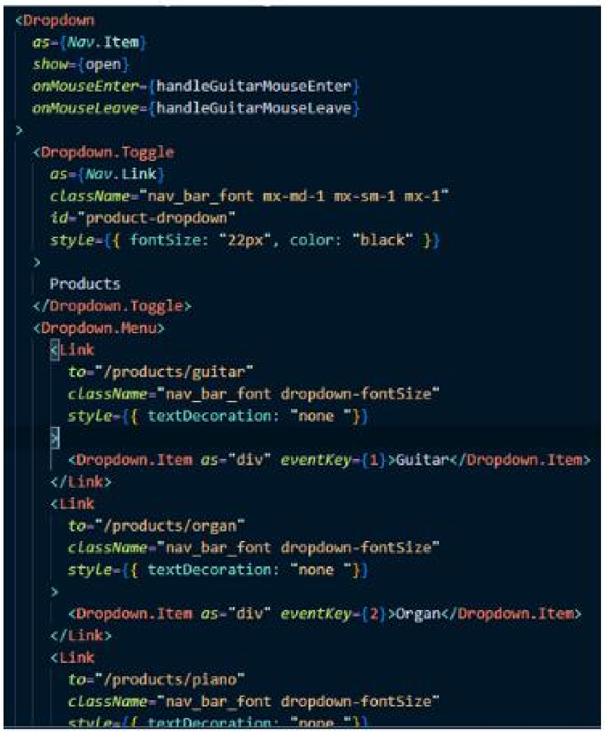
FIGURE 16. Contact form

In order to contact the store, users can make one click towards Contact item on the navigation bar and a form will be displayed to fill in if they need to contact the business for services. In the form, user is asked to fill in the name, email address with the country specified, and finally, a collapsible textbox with the subject is left blank for users to write down the message they want to send to the store. After the form has been filled, user just simply presses "Submit" button to send the inquiry to the store's email address.

4.2.7 Product list and detail pages

On the navigation bar, user can hover the Products item to display the dropdown content of different categories of music instruments which includes Guitar, Organ, Ukulele, Piano, and Drum. The dropdown menu was developed with built-in library of Bootstrap, and inside the menu contains the items of instruments with the user's favour.





```
<Dropdown
  as={Nav.Item}
  show={open}
  onMouseEnter={handleGuitarMouseEnter}
  onMouseLeave={handleGuitarMouseLeave}
>
  <Dropdown.Toggle
    as={Nav.Link}
    className="nav_bar_font mx-md-1 mx-sm-1 mx-1"
    id="product-dropdown"
    style={{ fontSize: "22px", color: "black" }}
  >
    Products
  </Dropdown.Toggle>
  <Dropdown.Menu>
    <Link
      to="/products/guitar"
      className="nav_bar_font dropdown-fontSize"
      style={{ textDecoration: "none "}}
    >
      <Dropdown.Item as="div" eventKey={1}>Guitar</Dropdown.Item>
    </Link>
    <Link
      to="/products/organ"
      className="nav_bar_font dropdown-fontSize"
      style={{ textDecoration: "none "}}
    >
      <Dropdown.Item as="div" eventKey={2}>Organ</Dropdown.Item>
    </Link>
    <Link
      to="/products/piano"
      className="nav_bar_font dropdown-fontSize"
      style={{ textDecoration: "none "}}
```

FIGURE 17. Dropdown list of music instrument

FIGURE 17 is depicting the dropdown menu implemented with Dropdown library. In order to toggle the menu, React hook useState library is imported to handle the state management with mouseOver attributes over the Products item. If the mouse hovers on the item, it will drop down the menu items, and vice versa. In addition, react-router-dom manages to redirect the user to specific product list page with Link library and path specified to the components where the user wants to visit.

The user is directed to a product list page with a block-style display of options and individual product cards for each instrument after clicking on a particular one on the website depicted in FIGURE 19. Each product will have a unique ID, and when a user clicks on one of the product cards, the website will instantly navigate the user to the product detail page, which includes a photo URL from assets and data fetched from the mock data module.

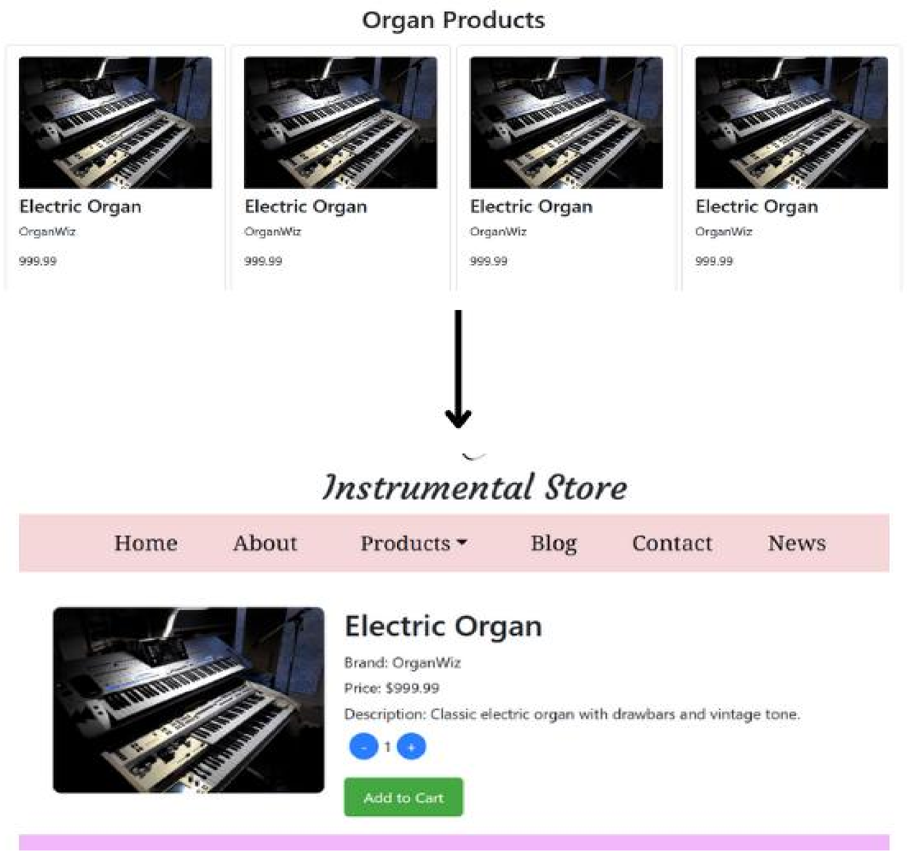
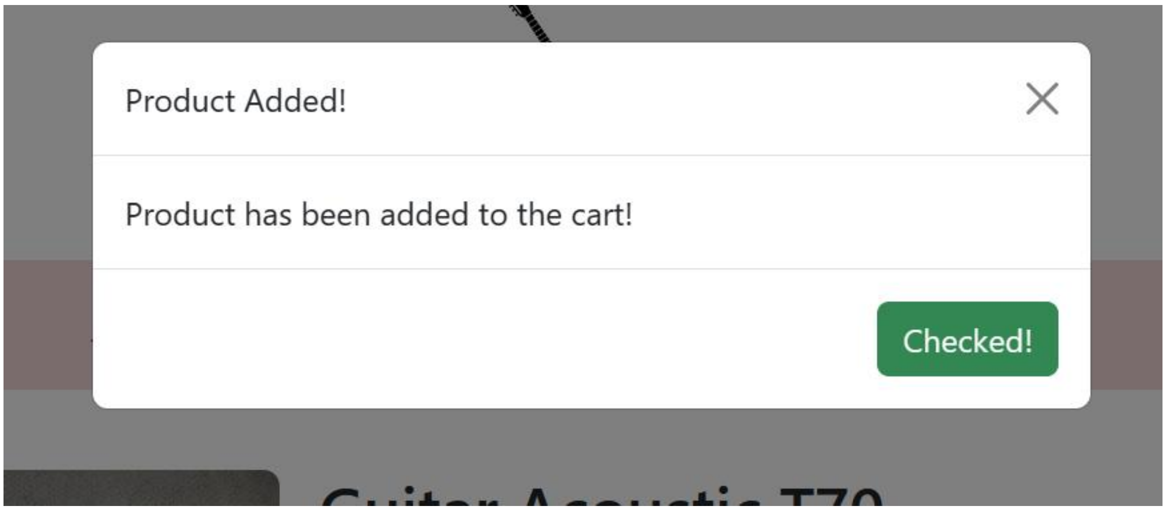


FIGURE 18. Product list and detail page

On the product detail interface, the data displayed on the main section is invoked from the sample data with the implementations of useState, useEffect, and useContext hooks to handle the state across the components. In FIGURE 18, the content includes photo illustration, title with the brand, price, and description of that organ music instrument pattern. User can manage to increase or decrease the amount of product they want to add to the cart, or they can later add up the quantity in the shopping cart after having added the product initially into the cart. React useState hooks are set to update the states of product and quantity across the application. The product detail page has a "Add to cart" button to add the product is connected to the Cart context to add the product to the cart as the context handles the CRUD operations including create shopping cart product, add, update, and delete products. From the product detail page, "Add to Cart" button will notify the user that the product has been added to the cart with a pop-up modal box confirmation.



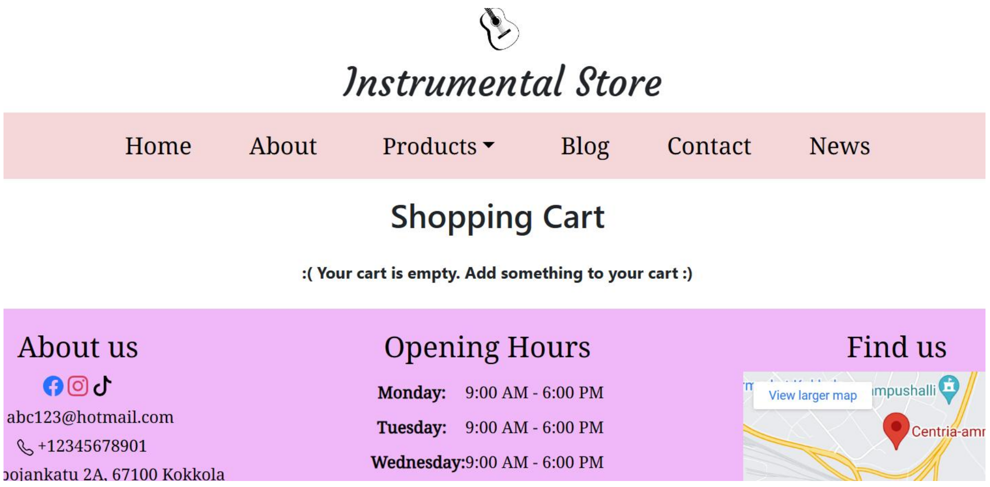
Product Added!

Product has been added to the cart!

FIGURE 19. Product added confirmation in a modal box

4.2.8 Shopping cart

Shopping cart is the place where the user moves to after having added the product to the cart with a certain number of products. The shopping cart has a connection with the product detail page when a product has been added to the cart to proceed to pay the product. At first, the cart is set to be empty because there is no product be added. If the user starts adding product to the cart, the cart storage retrieves the information with a product id from top to bottom if multiple products are added.



Shopping Cart

:(Your cart is empty. Add something to your cart :)

About us

abc123@hotmail.com

+12345678901

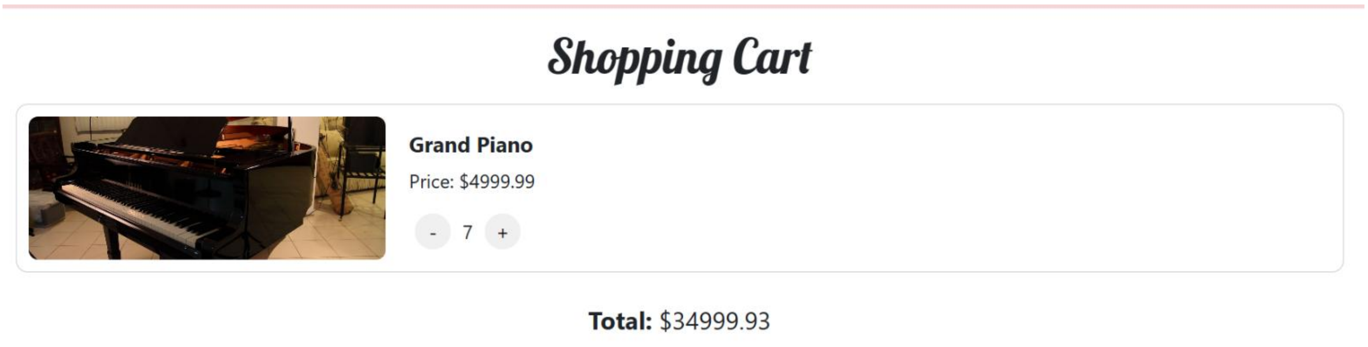
ojankatu 2A, 67100 Kokkola

Opening Hours

Monday:	9:00 AM - 6:00 PM
Tuesday:	9:00 AM - 6:00 PM
Wednesday:	9:00 AM - 6:00 PM

Find us

FIGURE 20. Empty shopping cart


Shopping Cart

Grand Piano

Price: $4999.99

Total: $34999.93

FIGURE 21. Shopping cart with product

As it can be seen in FIGURE 21, once the product has been added to the cart, the shopping cart module will display the product card with the number of items set one as the default value. As the number of products increases, the price will accumulative increase. In addition, users can add as many products as they can, and the following products will be in queue consecutively from top to bottom. In the product card, there are two buttons handling increasing and decreasing the amount of products with the Context manages the CRUD operations. If the product decreases down to zero, the product card of that item will be deleted from the cart. Thus, the cart might have no product and the screen shows a message of adding new items to the cart.

5 CONCLUSION

Overall, the objective of the thesis is to deliver a responsive e-commerce web application that solves the businesses' problem when it comes to selling music instrument. For customers, the web application may be a solution to customers who are living far away as a method to interact and contact the business for further services related to music instruments. The application throughout the project primarily focuses on the frontend development with the support of some programming languages and tools including HTML, CSS, JavaScript, ReactJS, and React Bootstrap with Git to control, track, and commit the code to GitHub platform. The application has also implemented fundamental functionalities by utilising API Context to handle the CRUD operations to add, delete, and update the product to the shopping cart. In addition, external APIs are used to fetch the data to serve the page's demand to bring a good experience to users via reading blogs and news on the webpage.

The web application is erected on the client-side at the first state, and the user interface is designed with simple patterns and there are not too much effects included in to bring the user with best surfing experience. At first, the application aimed to bring Redux technology into the application to handle the state of components across the application. However, the states inside the application are not complicated to utilise Redux, and the useState, useEffect, and Context hooks help save the time and cost to handle states. When it comes to the screen rendering, the application has fundamentally met the demand of creating responsive feature fitting the user's device scale, both on desktop and mobile devices with the implementation of Bootstrap built-in classes and device breakpoint conventions. The users are able to visit different pages on the web, especially they can click on one category in Products item to forward to product list page and users can view and choose to add to cart and order to buy the product based on their selections. For the cart, Context API is developed to handle the CRUD functionalities of adding, deleting, and updating the number of products. Also, the Contact page includes a form to fill in so as to send to the business a message with the feedbacks, services, and other questions related to music business of the store.

For the next development phase, database management systems will be implemented in order to store, manage, retrieve, update and modify a large amount of data dynamically. Furthermore, the backend is going to be developed to handle the logic of forms and payment. The web application is going to be hosted on Vercel or Netlify in the future, and the CI/CD works will be contributed to improve the application's performance with bug avoidance and code failures whilst dedicating to releasing cycles of software development.

REFERENCES

Chaparro & Bernard (1999). *Sitemap Design: Alphabetical or Categorical*. Available at: https://www.researchgate.net/publication/253537215_Sitemap_Design_Alphabetical_or_Categorical. Accessed on 09/05/2024.

Chien, D. (2024). *File Structure.* Available at: https://mitcommlab.mit.edu/broad/commkit/file-structure/. Accessed on 09/05/2024.

Coursera (2023). *What is GitHub and Why Should You Use It?* Available at: https://www.coursera.org/articles/what-is-git. Accessed on 30/04/2024.

DuRocher, D. (2021). *HTML & CSS QuickStart Guide: The Simplified Beginner's Guide To Developing A Strong Coding Foundation, Building Responsive Websites, And Mastering The Fundamentals of Modern Web Design.*

Dzhangarov, Pakhaev & Potapova (2021). *Modern web application development technologies*. Available at: https://www.researchgate.net/publication/352276087_Modern_web_application_development_technologies. Accessed on 30/04/2024.

Franklin, S. (2001). *Planning Your Web Site With UML.* Available at: https://people.apache.org/~jim/NewArchitect/webrevu/2001/05_18/developers/index01.html. Accessed on 30/04/2024.

Heller, M. (2022). *What is Node.js? The JavaScript runtime explained.* Available at: https://www.infoworld.com/article/3210589/what-is-nodejs-javascript-runtime-explained.html. Accessed on 30/04/2024.

Jain, A. (2017). *SOLID Principles and CSS (Part 1)*. Available at: https://medium.com/@ansujain/solid-principles-and-css-part-1-e73aeeb4bdc5. Accessed on 30/04/2024.

Kim, R. (2020). *The Impact of COVID-19 on Consumers: Preparing for Digital Sales.* Available at: https://ieeexplore.ieee.org/abstract/document/9076858. Accessed on 30/04/2024.

Kods, Y (2023). *Understanding the Key Files in a React App.* Available at: https://medium.com/nerd-for-tech/understanding-the-key-files-in-a-react-app-1729391ce88b. Accessed on 30/04/2024.

Microsoft (2022). *What is Git?* Available at: https://learn.microsoft.com/en-us/devops/develop/git/what-is-git. Accessed on 30/04/2024.

Mozilla (2024). *CSS: Cascading Styles Sheet.* Available at: https://developer.mozilla.org/en-US/docs/Web/CSS. Accessed on 30/04/2024.

Mozilla (2024). *HTML: HyperText Markup Language.* Available at: https://developer.mozilla.org/en-US/docs/Web/HTML. Accessed on 30/04/2024.

Mozilla (2024). *JavaScript.* Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript. Accessed on 30/04/2024.

Shaw, Eschenbrenner & Baier (2022). *Online shopping continuance after COVID-19: A comparison of Canada, Germany and the United States*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9379614/. Accessed on 30/04/2024.

Ranjan, Sinha & Battewad (2020). *JavaScript for Modern Web Development.* Noida, India: BPB Publications.

Rawat & Mahajan (2020). *ReactJS: A Modern Web Development Framework.* Available at: https://ijisrt.com/assets/upload/files/IJISRT20NOV485.pdf. Accessed on 09/05/2024.

Robbins, J. (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics.* California, United States: O'Reilly Media.

Salehmubashar (2024). *What is react-router-dom?* Available at: https://www.geeksforgeeks.org/what-is-react-router-dom/. Accessed on 30/04/2024.

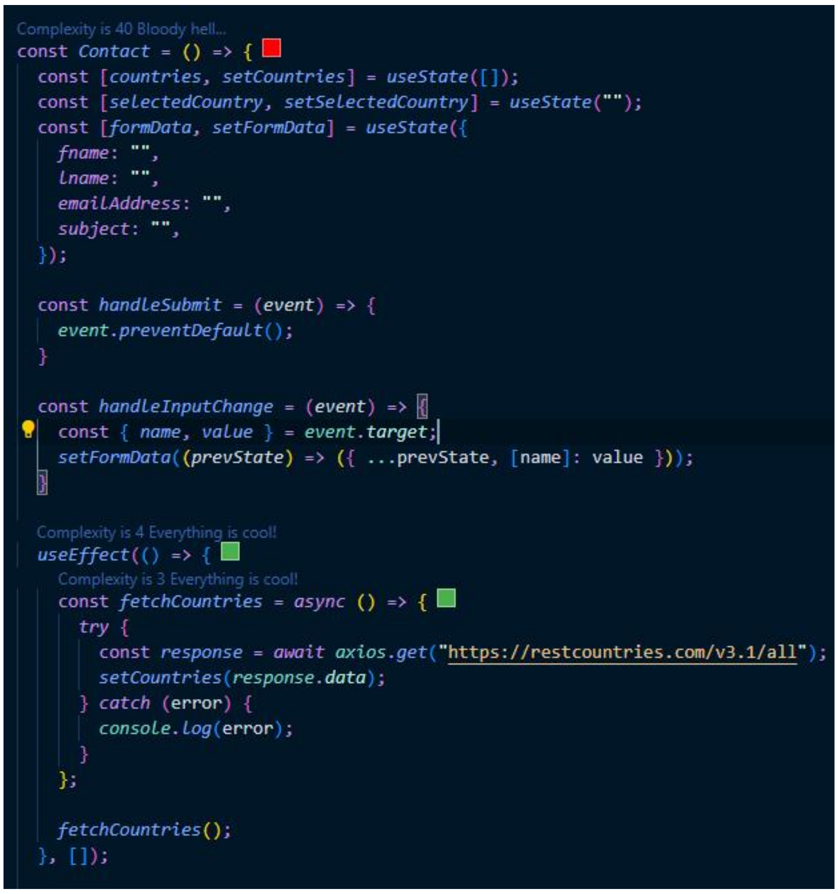
Shaheb, J. (2022). *Learn CRUD Operations in JavaScript by Building TODO APP.* Available at:

https://www.freecodecamp.org/news/learn-crud-operations-in-javascript-by-building-todo-app/. Accessed on 30/04/2024.

Sonpatki, P. (2016). *ReactJS by Example – Building Modern Web Applications With React.* Birmingham, England: Packt Publishing.

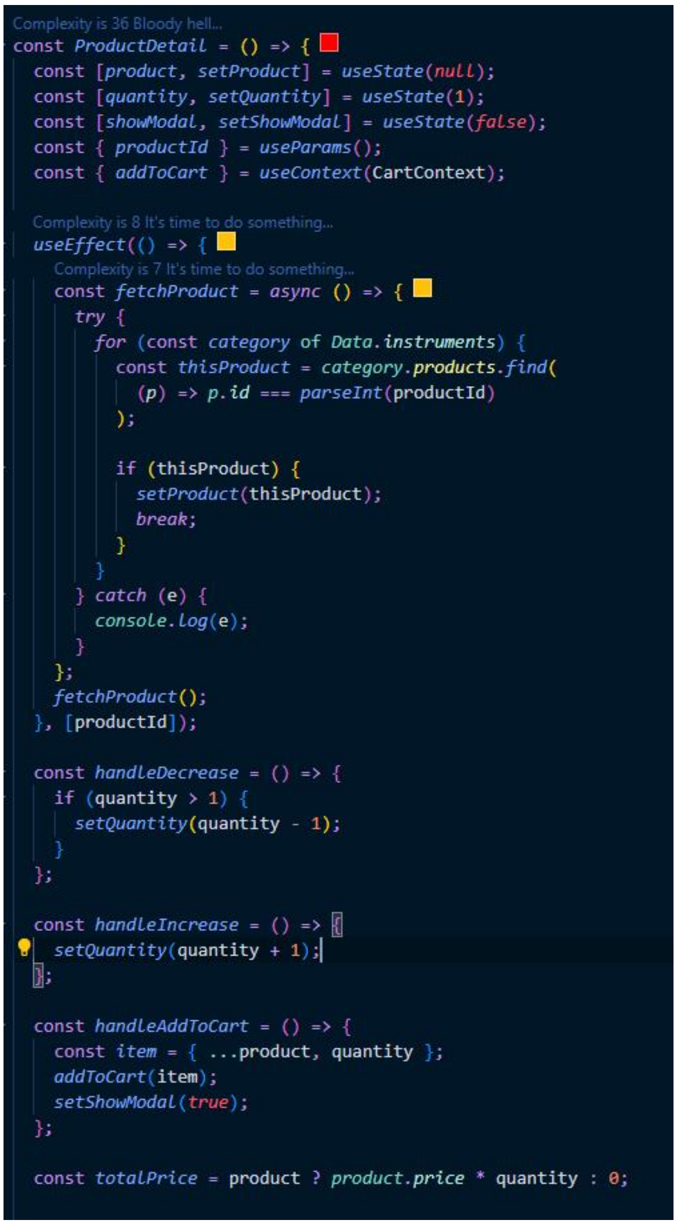
W3Schools (2024). *Introduction to Git and GitHub.* Available at: https://www.w3schools.com/git/git_intro.asp?remote=github. Accessed on 30/04/2024.

APPENDICES



```
Complexity is 40 Bloody hell...
const Contact = () => {
  const [countries, setCountries] = useState([]);
  const [selectedCountry, setSelectedCountry] = useState("");
  const [formData, setFormData] = useState({
    fname: "",
    lname: "",
    emailAddress: "",
    subject: "",
  });

  const handleSubmit = (event) => {
    event.preventDefault();
  }

  const handleInputChange = (event) => {
    const { name, value } = event.target;
    setFormData((prevState) => ({ ...prevState, [name]: value }));
  }

  Complexity is 4 Everything is cool!
  useEffect(() => {
    Complexity is 3 Everything is cool!
    const fetchCountries = async () => {
      try {
        const response = await axios.get("https://restcountries.com/v3.1/all");
        setCountries(response.data);
      } catch (error) {
        console.log(error);
      }
    };

    fetchCountries();
  }, []);
```

CODE 5. Contact form code.



```
Complexity is 36 Bloody hell...
const ProductDetail = () => {
  const [product, setProduct] = useState(null);
  const [quantity, setQuantity] = useState(1);
  const [showModal, setShowModal] = useState(false);
  const { productId } = useParams();
  const { addToCart } = useContext(CartContext);

  Complexity is 8 It's time to do something...
  useEffect(() => {
    Complexity is 7 It's time to do something...
    const fetchProduct = async () => {
      try {
        for (const category of Data.instruments) {
          const thisProduct = category.products.find(
            (p) => p.id === parseInt(productId)
          );

          if (thisProduct) {
            setProduct(thisProduct);
            break;
          }
        }
      } catch (e) {
        console.log(e);
      }
    };
    fetchProduct();
  }, [productId]);

  const handleDecrease = () => {
    if (quantity > 1) {
      setQuantity(quantity - 1);
    }
  };

  const handleIncrease = () => {
    setQuantity(quantity + 1);
  };

  const handleAddToCart = () => {
    const item = { ...product, quantity };
    addToCart(item);
    setShowModal(true);
  };

  const totalPrice = product ? product.price * quantity : 0;
```

CODE 6. Product detail code