Ngoc Bui

Improving user experience and functionality for auction application

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

23 March 2022

Abstract

Author:	Ngoc Bui
Title:	Improving user experience and functionality for auction application
Number of Pages:	31 pages
Date:	23 March 2022
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	Mobile Solutions
Supervisors:	Patrick Ausderau, Senior Lecturer

The purpose of this thesis was to improve the satisfaction of users when using the auction application. It allows Nordea and its partner to resell their refurbished equipment in Nordic countries. Users are divided into three types: normal users, admin and partners. The work would include fixing user interface (UI) and adding functionality in order to smoothen user experience and control data. Furthermore, it is also a possibility to learn and get familiar with the new language as well as code structure for future maintenance.

The project structure includes a web repository and two server repositories which are written in TypeScript. There are several critical tasks that need to be focused on; for example, how to get correct input of area code for phone number or how to solve a scraping auction. Besides, there are some other issues related to UI display which are not urgent but add more value to performance if they get fixed.

Generally, the given tasks have been completed which satisfies the stakeholder. Though the new functionalities are achieved, there is still always a place for further development. Those are built as a minimum viable product for testing. If there is a frequent usage with a vast number of users, obviously they will be improved in the future.

Keywords: React, TypeScript, Heroku, PostgreSQL, Docker

Contents

List of Abbreviations

API: Application programming interface

AWS: Amazon Web Services, a cloud platform used to host and deploy applications

CRUD: Basic operation of storing persistently which includes create, read, update and delete

DOM: Document Object Model, a developing representation for HTML and XML documentations

ECMAScript: European Computer Manufacturers Association Script

ES: European Computer Manufacturers Association Script

HTML: Hypertext Markup Language, a code used to build the user interface of the web

HTTP: HyperText Transfer Protocol, a protocol to communicate with servers

HTTPS: HyperText Transfer Protocol Secure, a more secure method to communicate with servers

IaaS: Infrastructure as a service

JS: JavaScript, a popular coding language

JSON: JavaScript Object Notation, a data format in human readable-text

JSX: JavaScript XML, a method to construct rendered components

OKD: Origin Kubernetes Distribution

ORM: Object-relational mapping

OS: Operating system

PaaS: Platform as a service

SaaS: Software as a service

SQL: Structured Query Language, a written command used to interact with databases

TS: TypeScript, a JavaScript with syntax for types

UI: User interface

VDOM: Virtual document object model

XML: Extensible Markup Language, a set of codes or tags used to define a text in a digital file

XHTML: Extensible HyperText Markup Language

1 Introduction

Nordea Finance is a branch of Nordea which provides financing solutions for both corporate and private customers. The service can be processed promptly at your premises or Nordea branch. Customers can find company offices in every Nordic country with variable business sections and target groups. (Nordea, 2020)

In 2020, with the merging of SG Finans to Nordea Finance Equipment, Nordea expanded their services into different applications. One of the products most worth mentioning is Auction which is a platform for Nordea and its partner to resell their equipment under a form of auction or direct sale. The application was launched in 2021 in Norway and it is in the process of deploying to other countries.

The auction application was deployed recently and it has a small number of users. They consist of private or corporate customers, partners and administrators. Customers can login and place bids for a certain product, they can also set bids automatically and be informed each time a bid was raised. Partners can propose their auction and get approved by administrators. Finally, admins are people who process auctions and sales, they are able to manipulate the state of auction, create reports and give permission for partners.

The purpose of this thesis is to enhance functionality and fix responsive design of the components. It is a possibility to explore the code base for future maintenance. Due to the limited schedule, this thesis will focus mostly on improving the admin page in order to help administrators manage and process data better. Other user interface polishing tasks will happen throughout the application.

2 Material, method and technologies

2.1 Material and method

This chapter will introduce the technical part of the application and the procedure of achieving desired outcomes. The Auction was created as a platform for the company and its partner to sell used machineries, tools or vehicles. It uses React for developing front end, Node and PostgreSQL for server databases. After developing on a local machine, the code is pushed to the testing environment in Heroku. When the application works as expected, it will be deployed to production for the end user.

Regarding code structure, the application contains three repositories, one for web and the others for database. Due to the scope of the app, it is required to separate users and auction-objects databases. They are written in TypeScript (TS) and styling code is maintained by using the File Watcher plugin in the code editor. In order to connect databases and the web, Docker is used to pack and build the application.

The Auction has been developing for over a year and the predecessors used Trello as a kanban board. The tool has every information about what has been done, what should be done, bugs and changes. It was recommended to start with building some components to get familiar with the code at the beginning and then gradually move to functional features since both are equally vital. Improving details can be seen from the card in Trello with different critical levels or can be added if detected during experiencing the application.

2.2 TypeScript

2.2.1 Introduction

Since JavaScript (JS) has been popular for many years within the developer community, many coders have had interest towards it. However, it is not an object-oriented language and has a complicated structure. TS was an open-source programming which was introduced by Microsoft in 2012 and persistently grown to recent times. It allows the compilation of different coding languages into JS. (Dubey, 2018)

TS is considered as a superset of ECMAScript (ES) 2015 as shown in figure 1 which compiles JS. The file extension ends with .ts for normal TS file or .tsx if it contains JavaScript XML (JSX). This language would not be totally different from JS, but works as a type check since it adds more structure in maintaining the correct type of variables. In other words, TS is JS programming with type declaration and giving errors at runtime. (Freeman, 2019)

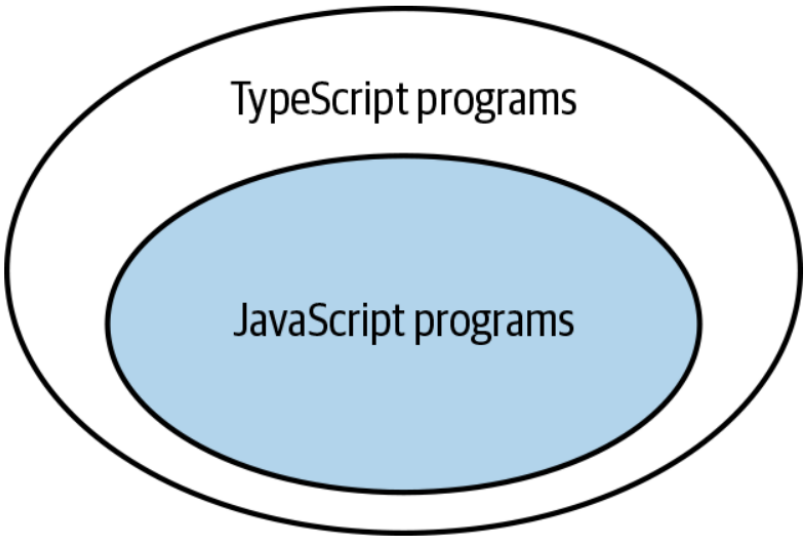

Figure 1. TypeScript is superset of JavaScript

TS can be used to develop backend and frontend which is also the case of this thesis's project. Since ECMAScript 6, a numerous number of new features have been added such as type annotation, interfaces, generics and async/await.

Furthermore, many code editors have supportive plugins for styling and code completion. For instance, Visual Studio Code or Microsoft's Visual Studio offers free support whereas in JetBrains, users need to have the ultimate licence.

2.2.2 Using TypeScript over JavaScript

It is certain that JS is one of the most popular languages to create a dynamic application. However, TS is growing fast as an object-oriented programming language and has been adopted by the JS community. Some reasons to add type checking into JS are that coders can dodge undefined error, refactor and build scalable projects with less effort. Research has indicated that 15% of errors from JS could be found by this language. (Bolgurtseva and Dreimanis, 2020)

Apart from mitigating errors, TS also has many added features that support modern development. For instance, Interfaces, Generics, Inheritance and Method Access Modifiers are used to access and define objects. It would be more readable when dividing code into classes and interfaces. Additionally, TS is extremely strict in datatypes which will instantly throw errors if they do not match the defined one as demonstrated in listing 1.

```
1 let testnumber: number = 6;
2 testnumber = 'myNumber'; //This will throw an error
3 testnumber = 5; //This will succeed
```

Listing 1. TypeScript type checking

A d.ts is a file that declares typing of each data and they are not compiled but used to check type of information. It provides a current and visible definition for application programming interface (API) due to automatic update and frequent maintenance. Furthermore, the file structure can have only one major .ts file which redirects to others and then delivers with the .js version. For that reason, developers can easily modify a single codebase for different browsers and platforms.

```
1 class overLoading
```

```
2       addStudent (id: number);
3       addStudent (class: string);
4       addStudent(value: any) {
5       if (value && typeof value == 'number') {
6             alert ('First overload - ' + value);
7       }
8       if (value && typeof value == 'string') {
9             alert ("First overload - " + value);
10      }
11  }
```

Listing 2. Overload function

Lastly, TS is usable through different browsers, devices and operating systems which does not require any virtual machine or runtime environment to process. An overload function, which is represented in listing 2, offers programmers a variable implementation of function regardless of parameter. (Sharma, 2018)

2.2.3 Limitation

Despite a number of advantages, TS contains some drawbacks which need to be taken into consideration. First of all, it is time-consuming when a company would like to change from JS to TS. Though it would be beneficial in a long-term development, coders need to make some investment in learning the typing and structure at the beginning. (Altexsoft, 2020)

In addition, since TS cannot be compiled directly to browsers, it requires a JS compiler to implement this process. Though processing time would be instant and automatic, it adds an extra step to the whole operation. Besides, if the project needs additional libraries, TS will require to have a definition file which is sometimes unqualified. (Vorontsova, 2019)

Thirdly, readability is one of the biggest advantages of TS; however, it would expand the code with type definition and obviously take more time in the developing process. Annotations would as well add more lines to the code base, so TS files are often more extensive than JS. In reality, TS could not solve every primary issue of JS but deceiving users with the concept of type checking. (Tate, 2020)

With both pros and cons, TS with great support from Microsoft has been favourable by many developers recently based on the report from State of Frontend 2020 (Kubiak, 2021). Though there are some disadvantages, it is not deniable that this language adds huge value to the code quality and development process, especially in large scale projects. With a few more lines of code, TS would help improve performance by decreasing error and rapidly releasing.

2.3 React

According to Meta, React was created for the company's usage at the beginning as a frontend development tool. Eventually, this JS library became popular and grew promptly after a software conference in 2013. Due to its popularity, programmers can have a diverse library and supporting community. React is an open-source cross-platform that can be used not only in web but also mobile with React Native and desktop. React Native has become popular within the company since the code base can be used in many different operating systems like iOS or Android. In addition, React is famous for its single page application which is supported in many browsers. (Borrelli, 2019)

There are many reasons that encourage a programmer to use React among other libraries. First and foremost, the reusable components save time for developers to focus on developing other functionality; therefore, it saves time and ensures the identicality of similar components. Besides, programmers can track the flow of data in every segment since it is unilateral binding. Intuition and declaration are two features that are worth mentioning in React. Coders can easily interact with UI and big data changes automatically modify the chosen display. (Deshpande, 2019)

On the other hand, though it has a huge supporting community, there is little official documentation compared to a huge number of updates and releases. Another consequence of the speedy deployment is that developers need to

constantly learn new things. It can be stated that React has an overwhelming number of goodside that can cover its disadvantages. (Baranowski, 2021)

2.3.1 Virtual DOM

Document Object Model (DOM) consists of Extensible Markup Language (XML), HyperText Markup Language (HTML) and Extensible HTML (XHTML) document under a tree form. Coders can manipulate the state, style and structure of the page through DOM. It can be interacted with by JS or TS since the term focuses on object representation (MDN, 2021). Node tree is created in components by `render()` method and can be manipulated with actions. A lighter version of DOM which is identical but allocated in memory is Virtual DOM (VDOM).

VDOM is a concept in programming where the user interface display is stored in memory and linked to real version through libraries like ReactDOM. The comparison of those terms can be visualised in figure 2 which shows that they are slightly identical. Reconciliation is a process which coders assure that DOM will place UI to correct state. It includes some actions which are required for building applications such as event handler, attribute control and DOM update. The pros of virtual DOM are that it allows coders to update a minor part without changing the whole component. (React 2022)

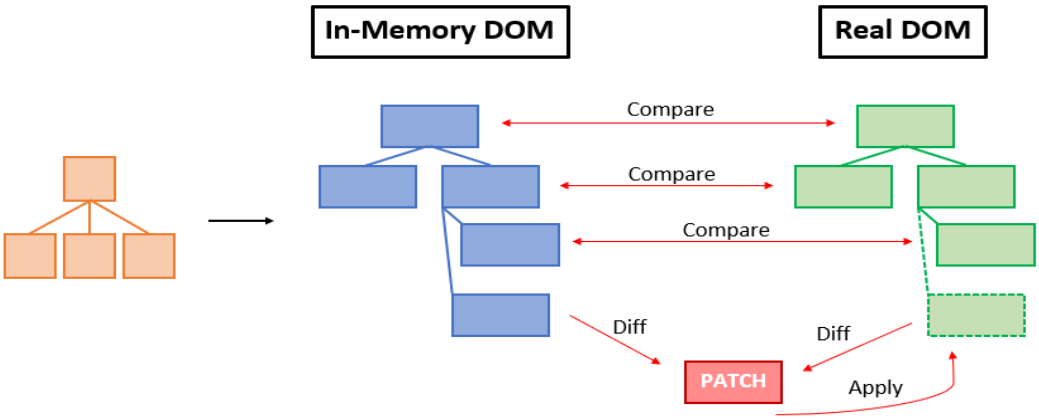

Figure 2. DOM mechanism

React generates the virtual DOM from scratch and keeps two versions in memory. Creating and rendering the three do not consume lots of time; therefore, improve performances of applications. Besides, it is the best way to interact with elements compared to other JS's methods. Once React amends DOM, code will amend components. Lastly, when data is updated, it will create the entire new interface hence saving a huge amount of memory. Generally, virtual DOM is totally similar to the actual DOM, but it takes less time to update which shortens compiling time. It provides a straightforward method to handle browsers when data is changing without keeping any observables in memory. As a result, it creates value in performance and productivity when developing applications. (Rigel Network, 2017)

2.3.2 Components and Props

Theoretically, components are reusable code that represents the user interface; for example, header, navigation bar or modal can be considered as a component. It is also a JS function which takes props and returns elements. There are two types of components: function and class which operate the same way. The first one has been popular recently due to the emergence of React Hooks which make the code shorter in managing state. However, some developers favour the traditional way that components are described by their own state in class.

Props connect components within a three and can be referred to as properties of each component. It is designed that parameters can receive different values but the prop itself is fixed in the project. For example, a title takes a name string as prop, whenever the component is used, it needs to pass a title name to prop. As a result, we have an identical title with their own value in different places.

There is a slight difference of using props in function and class. In former components, they will be passed from function argument while in latter, they are rendered by default and start with prefix `this.props`. Components get data from parents through props or generate their own with state. For some large

components with many functionalities, developers can nest small components inside the main one. Moreover, props passed to components are unlimited, coders can have as many as required and the value type is diverse. Apart from string, it can be integer, boolean or even a function and passed to props in a curly bracket. They also allow default values which means if nothing is given to the component, this value will be used. (EaseOut, 2021)

There are some special prop types called ‘children’ and ‘key’ where the first one represents every element passed in the body of the component. An example for this type is when using Hook Context where components with global variables wrap children inside; consequently, data can be used in every file without passing along the component tree. ‘Key’ is often used when passing elements of an array to children components, it helps React render lists more productive. (Pavlutin, 2021)

In conclusion, components are reusable files and take props as properties. It is advisable to divide complicated components into smaller one and put them under a separate folder. Function components have become widely used due to its transparent display and less code required to achieve the same purpose in class. (Yamazaki, 2020)

2.3.3 JSX

According to React, JSX is a JS extension and used to present a user interface. It creates elements which are then rendered by DOM by render function. JSX will turn into normal JS after compilation; therefore, it is possible to put it in function for complicated implementation. It is the same as HTML with similar tags such as `<h1>, <p> or <div>` and those are organised under JS components. In order to enhance error handling and execution, it is recommended to use JSX instead of plain JS in React. (React 2022)

```
1 const getGreeting = user => {
2      if (user) {
3            return <h1>Hi {formatName(user)}</h1>
4      }
```

```
5       return <h1>Hello fellow</h1>
6 }
```

Listing 3. JSX is an expression (React 2022)

Expression like variable or property can be included into JSX by curly braces as shown in listing 3, it will compile the formula and return with the result to display. Naming attributions like class name of element should be in camelcase and empty tags end with />. Input will be automatically altered into strings in JSX to avoid injection incidence.

2.3.4 State management

Along with the rapid development of React since 2012, many different methods have emerged to manage state changing. One of those would be React hooks or Redux library and recently released Recoil. Depending on scale and requirement of the project, state management techniques will be chosen.

Firstly, Redux is the traditional way which is a react-based library and used mostly in e-commerce applications. The basic structure of Redux would include a store where all state is contained, an action file which stores every function used to control state and a reducer that combines many actions to return new results from initial value. Redux can work independently in any framework and update UI as a return of state's change. The working flow of redux and the updating way of information are illustrated in figure 3.

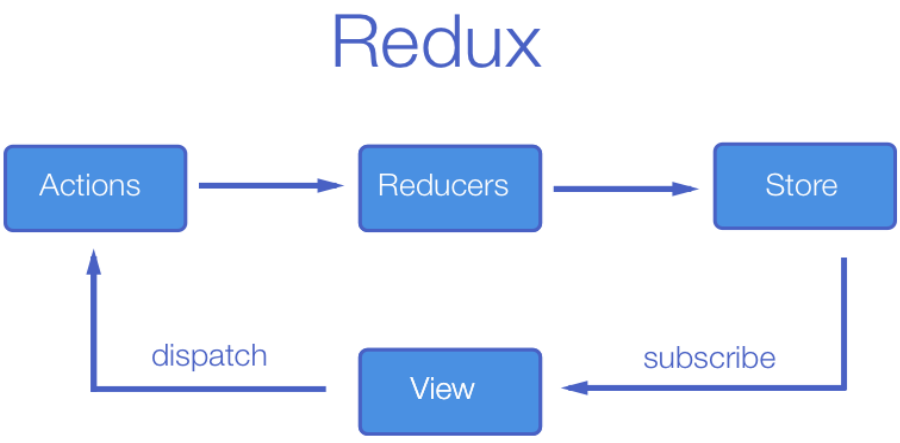

Figure 3. Redux mechanism (Jain, 2019)

Redux provides a united place where all state is controlled and changes are traceable. Therefore, it is advisable to use Redux when there are a huge number of states in application and it starts to be complicated to update at the same time. The combination between React and Redux has shown a positive result as it has a huge support community and optimising performance. Local Storage persists state making storing objects; for instance, shopping carts, more comfortable. Besides, the browser has a debug plugin for Redux which makes state controlling more efficient and visible.

Secondly, another feature that React has introduced to the community is Hooks. Unlike Redux, Hook controls state in functional components which operate independently as a class component. `useState, useEffect, useReducer` and `useContext` are the most popular which do not need to install an external library. An example of `useReducer` will be displayed in listing 4. The code base would be cleaner and smaller with Hooks than Redux. However, Hooks can be used to manage applications with a small amount of state without complicated updating logic.

```
1 const [state, dispatch] = useReducer((state,action) => {
2       const {type} = action;
3       switch(action {
4               case 'action name':
5                       const newState = //do something
6                       return newState;
7               default:
8                       throw new Error()
9       }
10 }, []);
11 }
```

Listing 4. useReducer example (Mihaela, 2019)

State is updated by `useReducer` which takes initial condition and reducer function as arguments. `useContext` will be responsible to update state instantly and can be called in every child component instead of passing them down the component tree.

Lastly, Recoil is a new, at time of writing in 2022, tool for state management which was built by Facebook and React. One of the highlighting features could be that it is easy to learn due to its simplicity and React-like approach. Within a couple years from deployment, this library received more than ten thousand stars on Github and was gradually adopted by developers. (Dev 2021)

In a nutshell, every method has pros and cons, depending on project architecture, coders can choose the most appropriate procedure. This project will use React Hooks to manipulate state changing since it is mostly used to store data fetching from API.

2.4 Backend

2.4.1 Node

Node, which was launched in 2009, was built on the V8 JavaScript engine in C, C++ and JS. It is used to create server-side applications which contain every access to build a completed applicable project. The main functionality of Node that makes it preferable by the community are its global approach and modules. Developers can use the same language for both front and back end. An application inside a Node ecosystem can have access to some crucial global variables. Module is represented for one functionality which needs to be exported to be able to be used in other locations. Modules can be imported by using function `require()`. Apart from writing modules, Node has many built in ones that support communication with network, terminal and computer systems. For instance, a web server has HTTP or HTTPS, File System or Path for file and OS. (Codecademy, 2022)

Node implements events in order with a call-back function in JS runtime. This function whose first argument is error handler will be called after the asynchronous implementation is done. The call-back function is implemented after the file is read, if there are errors, it will return a null otherwise print out the data as presented in listing 5.

```
const fs = require('fs');
fs.readFile('./file.js', (error,data) => {
     //error is null if there is no error but Error object if it is
     if (error) {
           throw error;
     }
     //the file data will be passed into callback if there is no error
     console.log(data);
});
```

Listing 5. Error replacement in callback function

According to Node.js User Survey Report (2018), Node has improved productivity and satisfaction for developers; besides, it reduced cost and enhanced performance of applications. The reasons behind those complements are because developers can use the popular JS in Node.js. They do not have to learn a new language, but only the convention and structure. Nowadays, TS is also an available choice to work with Node. Similar to JS, it has a huge supporting community and reusable material. It is mostly used in full-stack development and scalable projects by maintaining the speed of data communication and compatible with containerized microservices. Lastly, developers are offered with over 600 000 free packages in Node which support development greatly. (Dziuba, 2021)

On the contrary, server built in Node.js will be less efficient in create, read, update and delete (CRUD) applications since its best features are not applied. Besides, applications requiring relational queries or massive calculation will reduce the performance of this platform. While Node uses a central processing unit core, it will block all traffic if the server has a lot of computation. However, using relational databases in Node is not impossible nowadays with Sequelize object-relational mapping (ORM).

Overall, Node is a dynamic tool for constructing server-side applications such as data streaming or real-time apps. Based on a survey of Stack Overflow in 2020, over 50% of responses claim that developers use Node.js in development (Larson, 2020). One of the biggest cons is that it can not process many threads at the same time, but every line has to be implemented asynchronously.

However, developers can have many instances running simultaneously in parallel without syncing. The process is called worker thread which executes several tasks and queues the request if there are any lines available. (Gimeno, 2019)

2.4.2 Docker

According to its documentation, Docker is an open-source platform which assists applications to be developed and run on different systems. Traditionally, deploying a web application requires a server to be installed on Linux, then the second one is needed if traffic is more crowded. Nowadays, Linus Torvalds has developed software-based servers which operate on a runtime environment. This tool is called a container and one docker can run many containers simultaneously. (Docker, 2021)

Containers carry every necessary requirement to implement the application such as dependencies and settings. It would speed up the development time within developers since they do not need to concern what has been installed on the host, but only the containers. Docker has everything available for developers to test or deploy and the process is the same regardless of production environment which is in cloud, on premise or hybrid.

Containers have three main categories that are builder, engine and orchestration. Distrobuilder or Dockerfile is one example of a builder tool used for constructing the containers. Engine is the program control operation of those containers, it could be docker command or dockerd daemon. Lastly, orchestration refers to the technology used to run many containers consisting of Kubernetes and Origin Kubernetes Distribution (OKD). (Opensource.com, 2022)

Containers implementing procedures are similar to virtual machines but in a more detailed way. They separate applications and dependencies and all containers use the same operating system such as Linux or Windows. Many developers favour Docker because it improves efficiency in resource usage; for

instance, a virtual machine would consume more memory than a container. An application which is kept in a container would be light for the system and be able to run in a short time. It is also cost-wise since there are not many OS instances required to handle a single load of work. (Yegulalp, 2018)

In addition, time-savings and commutability are one of Docker's huge benefits for users. With Docker, applications can be delivered to production rapidly because it handles updates of new features and processes new versions in less time. Besides, as mentioned before, containers keep everything inside, so it is easy to transport among different environments. Developers would only need to install Docker in order to run containers. (Yegulalp, 2018)

Last but not least, Docker is a saving tool for microservices which was built from variable smaller services. Though it is not compulsory, it adds more value to the development cycle and handling microservices procedure. However, there are a few points which need to be considered when using Docker. It is not a replacement of virtual machine and can not convert application’s design into microservices. Furthermore, even though it would be more secure to put apps inside containers, it does not ensure entire security from other breaking attempts. Docker would be a right choice for projects whose requirements are rapid response, quick updates and more freedom in development. (Yegulalp, 2018)

2.4.3 PostgreSQL

PostgreSQL or Postgres refer to an advanced open-source managing platform for relational databases. The Ingrea project was developed at the University of California and then evolved to Postgres at the beginning. In 1996, it was built to operate with Structured Query Language (SQL); therefore, the name was changed to PostgreSQL. It is programmed to implement a huge load of data with a great number of users at the same time. MacOS was the first system that PostgreSQL was designed for, nevertheless, it was available on Windows and other platforms as well. (Postgres, 2017)

Unlike NoSQL which stores data in fields of JSON forms, PostgreSQL uses tables and those are linked with each other with foreign keys. Another feature that makes this program sophisticated is schema. Since it includes core information of the database such as types and tables, schema needs to be prepared particularly at the beginning. It will be difficult to change after the schema has been created. (Borozenets, 2021)

Additionally, required data can be called from different tables at the same time according to SQL formation. One of the reasons that programmers would like to choose PostgreSQL for large sized projects is because of its security feature. Transparency and accuracy are prioritised in SQL systems. Every data transaction can carry as many changes as possible and they will be either rejected or received in order to maintain its accuracy. Besides, if data does not belong to any tables nor fit any criteria, the system will stop its access to the database. (Borozenets, 2021)

Data types and indexes are also main characteristics of PostgreSQL. It offers a wide range of data types that is more suitable with unstructured or exclusive data. Therefore, this software could be a choice for big projects with complicated and clustered databases. Responding time is concerned by many developers to enhance the user experiences. PostgreSQL offers a diversity of indexing choices since it was used to get queries steadier. (Chen, 2021)

In a nutshell, PostgreSQL is compatible with any big, scoped project where a lot of data has to be updated constantly and retrieved from many different sources with queries. For those whose structure is adjusted frequently, this tool would not be considered for managing databases. Nevertheless, since it was built by a university team, it lacks professional support, but the online community is respectable as well. Furthermore, PostgreSQL’s user interface is pgAdmin which offers a more convenient way to manipulate the database of an application.

2.5 Cloud services

Cloud service provides efficient and economic access to resources without hardware on premises. It is managed by a third party cloud computing and services provider which allows applications to be accessed on the network. Cloud services include infrastructure, platforms, software and technology which also represents four main solutions. By using the cloud, companies can easily scale their project, reduce expenses and improve flexibility. Since cloud provides every needed program, companies do not need to have extra personnel and be able to buy added licence along with the project size. Moreover, cloud providers have different subscriptions depending on the needs of the project. Developers can access many different services on cloud such as software or storage and do not need to concern about updates. When a company decides to change the cloud or stop using it, they can just cancel the subscription. (Red hat, 2019)

With the fast emergence of cloud computing platforms, many developers have been using it in most of their projects due to its mobility and capacity. There are many choices depending on company budget and purpose, the most popular ones can be Amazon Web Services (AWS), Microsoft Azure or Google Cloud Platform. (Dignan, 2021).

Cloud service has four solutions and Heroku is Platform-as-a-Service (Paas) that hosts applications and allows developers to have large-sized projects. Heroku is aiming to target customer-oriented applications, the deployment process is handled with all required servers and softwares (Rusev, 2019). Heroku’s user interface is friendly for both technical and normal users with a well-organised dashboard to manage, keep track and deploy applications. Besides, using cloud does not require own infrastructure and it is supporting for many languages. Developers do not need to consider installing libraries, configuring security methods or operating systems which is also a downside compared to other platforms. Though the community is small, the supporters are very active in sharing and tutoring newcomers. In addition, DevCenter which

is Heroku's developing support forum contains informative documentation and most of the solutions of the problem.

Multi-language support adds a value to Heroku, in case it does not support a specific language, developers can always use build-pack or customise the existing ones. Heroku offers PostgreSQL application instant access and high security as well as those add-ons that support storing data and managing data. Scaling projects can be done horizontally or vertically, based on process types and app instances. The first option refers to expanding the infrastructure while the other means increasing the capacity of the current engine. Heroku also provides a useful report and error of the deployed application. (Almeida, 2019)

On the other hand, some drawbacks of Heroku are frequent unreachability, high latency and not supporting intensive computing applications. Another option for cloud could be AWS which originated from Amazon. AWS offers a wider range of services including Infrastructure-as-a-service (IaaS), PaaS and Software-as-a-service (SaaS) with cheaper price for basic subscription. Therefore, it would be more applicable for medium and big organisations compared to Heroku. Nonetheless, AWS's algorithm is much more complicated than Heroku for a beginner and companies obviously need an expert for maintaining applications.

Thirdly, Azure is a hybrid cloud platform consisting of IaaS and PaaS solutions. In order to use Azure, users need credit card information and the providers offer some free credit each month. However, many services require extra payment apart from testing. They have a free package for students with less credits and zip size. Generally, Microsoft Azure and AWS have similar features and both work well with heavy computing projects. A cloud expert will be essential if the company would like to use these platforms.

In a nutshell, each platform has its own pros and cons, depending on business purpose and technical level, developers can choose the best option for their project. Heroku lets companies build, deploy and keep track of their application in a most efficient way. Due to less complicated procedures, it is more

applicable for small organisations which want to experiment on their products before making investment decisions. Like other platforms, users can pay on demand which is a cost-saving point for enterprises with limited budgets.

3 Implementation and findings

Those tasks have been selected based on need and difficulty level in order to help new developers get familiar with the codebase and application. Starting point will be fixing UI, adding components and then creating a full functionality from database to frontend.

3.1 Handle deleting user

The administrators have received some requests from the users to erase their accounts in the database recently. The problem is how would the data be displayed when a user wants to have their accounts deleted because user data is related to many other tables. For example, bidding history or the winner of the auction are crucial pieces of information that need to be stored. Since user and object databases are separated, when deleting user information totally, only part of id is kept in the auction object database. This helps maintain the proper function of the application while still fulfilling the request of the customer.

Practically, a conditional function has to be applied whether there is any user information found, it will return the id. The solution seems to be easy to conduct but it requires a deep understanding of how those data connect to each other as well as the props passing to components.

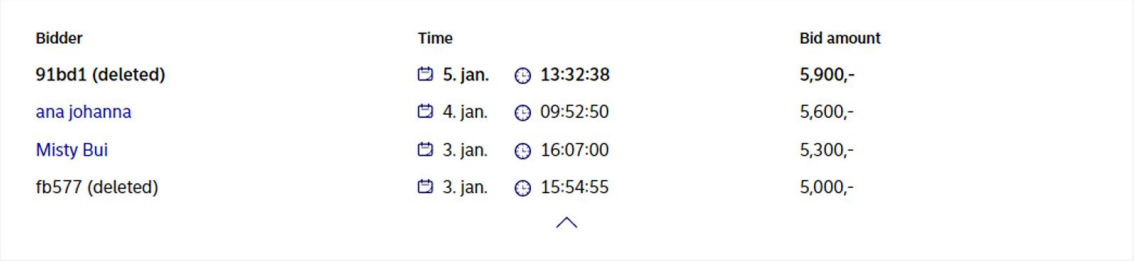

Figure 4. Example of deleted bidder display

Deleted users will be displayed with their identity number in the bid history which is illustrated in figure 4. According to regulations, if users do not have any bid, their data can be removed totally; however, if they have placed or won an auction, it is allowed to keep information for five years. Users can request to delete their data by sending email to administrators as shown in the figure 5.

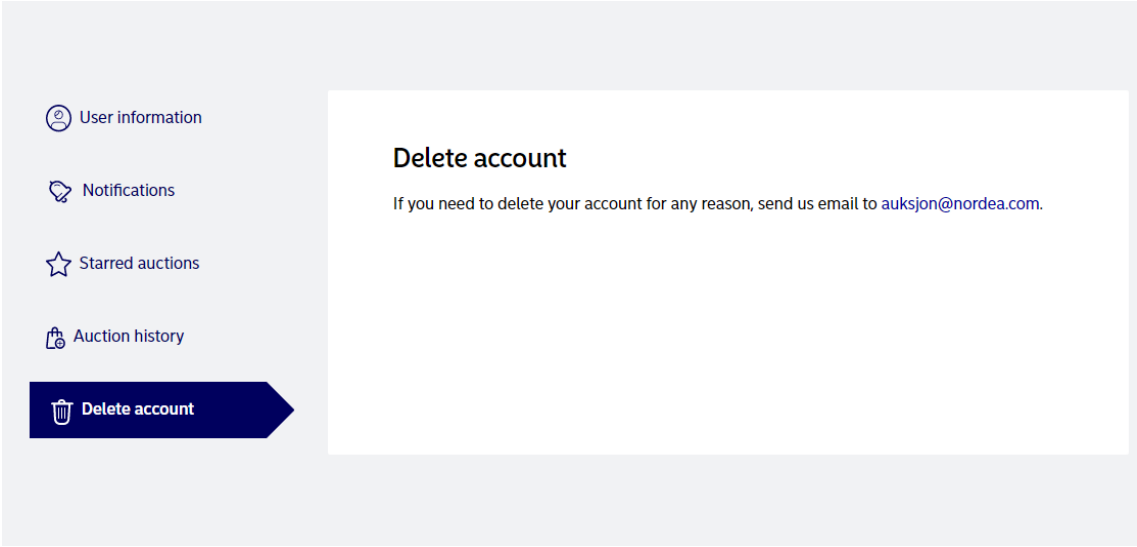

Figure 5. Delete tab in My Account

Instead of creating a submitting form, the request can be handled manually when administrators receive email since there are a few users who would like to delete their data. This solution helps save lots of time of developing but still satisfies the users.

3.2 Add country code dropdown

When registering an account to login the application, area code is required for the phone number since users are from different countries. If the phone number is incorrect, they could not receive notifications and there are many cases that users input wrong by accident. Therefore, the solution is to create a dropdown with fixed area code for each country and Nordics will be displayed on top since those are the main using market.

A third party library could not be used but the component had to be built from scratch instead. It is not advisable by the company to fetch API from a free endpoint, so the database was built in json for every country code. At the beginning, the fixed area code dropdown was used in the application; nevertheless, it is extremely tricky to customise the visual to be the same as the designed one. Another solution is to use that dropdown but reduce some requirements from the designer. The latter option was first implemented , but it did not meet the identical display of the company and poor performance. Therefore, after some discussions, it has been decided to build another dropdown component for phone number input as shown in figure 6 which fits the requirement of the task.

Figure 6. Area code dropdown

In order to match design and functionality, the dropdown contains three separate components and are aligned by flex. The country selection can be opened or closed when clicking on the left dropdown whose default value is Norwegian code. The right is an input box where users can only insert numbers. The countries container is a scrollable list which goes through the array and displays each data. A line is added under the fourth country by css styling to distinguish Nordics and the rest.

3.3 Handle scraping auction object

There are a few requests on how to handle the object which can not be sold even after republishing and staying for a minimum of six months in the auction. At the moment, this object will stay in the unsold list eternally and that requires some actions to move it to history instead. The idea is to create a button to scrap these objects and process this type of auction to history where sold and other sales are stored.

The implementation consists of creating a new table to store scraping information such as object id and date. Initially, there are a few more fields like who is responsible for the action and the reason for destroying the object. However, in order to simplify the procedure and make use of existing functions, the admin needs to submit a description where reason and their identity are stated before proceeding further. Those information will be kept under notes part of each auction. After pressing the confirmed button, the auction will be moved to the processed list and it is irreversible.

The challenge of the task is writing queries to return more columns from the new table. The original function has already been joined from many tables and a lack of understanding the way they relate to each other causes difficulty in creating new query. Furthermore, the old jest test has to be fixed in order to work properly as well as a test for the new functionality. The authorization for different users such as admin and partner has caused many trouble in rewriting the test since they share the same function but scrapping is only for admin.

```
1 pg.oneOrNone<object>(
2      `WITH object_documents as (
3            SELECT ARRAY(
4                  SELECT json_build_object(
5                        'document_id', document_id,
6                        'name', name
7                  )
8                  FROM document
9                  WHERE object_id = $[objectId]
10                 ) as documents
11           )
12           SELECT object.*, object_documents.*, destroyed_object.id
13           FROM object
14           JOIN object_documents ON object.id = $[objectId]
```

```
15            LEFT OUTER JOIN destroyed_object ON destroyed_object.id =
object.id
16            WHERE object.id = $[objectId]
17      `,
18      { objectId },
19      );
```

Listing 6. Example of query

The query in listing 6 is an example of getting nested data from different tables such as general object, document and destroyed object table. There is a number of certain information required from each table, especially the document. Therefore, the files belonging to the same object would be extracted as a json in the returned result.

3.4 Add form for leasing company information

When an object is sold as an auction or normal sale, it will be processed to a next phase where admins keep track of payment method and documentation. If the item is then leased to another company, a form is required to get and display information. The implementation also includes a new table in the database and a form in the web.

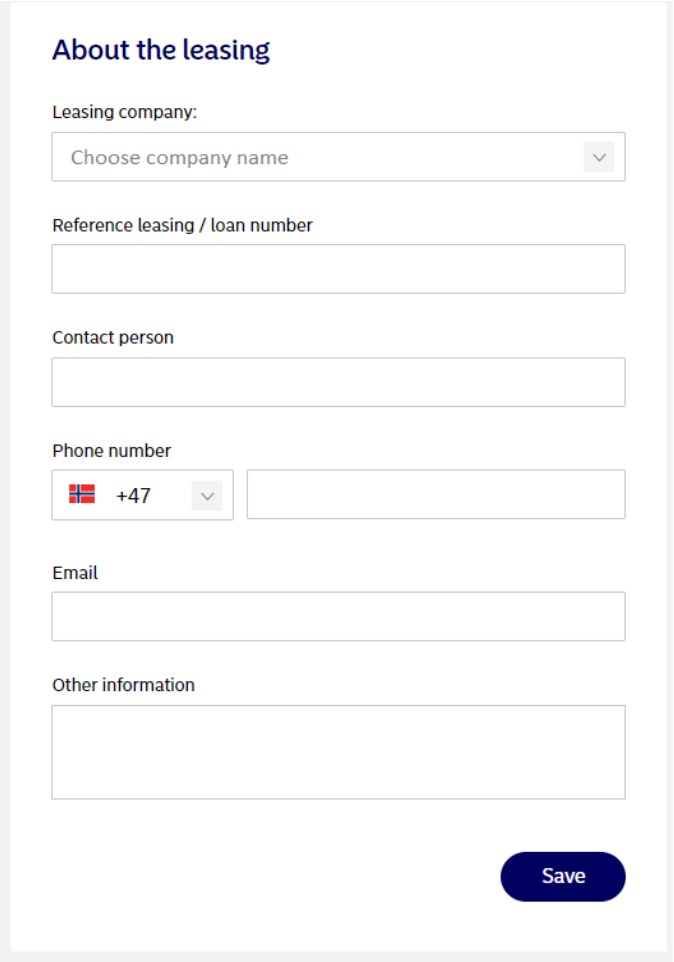

Figure 7. Leasing form

The information of the leasing company will be saved from the form displayed in figure 7. It is not compulsory to fill in every field of the table. Initially, information is fetched from the database, concrete data will be shown instead of blank if the company has already been registered to the system. After saving, the details will appear below the form and the user can always edit them afterwards.

The implementation would need performance in both front and back end. At the beginning, the table which contains information of the leasing company is created with the identification number of the object as primary key. Accordingly, an object will be leased to only one company and its details can be updated on any occasion. Lastly, it is straightforward in creating a form in the application since it is built on existing components such as input text, dropdown and button.

3.5 Add English notification for email and SMS

While exploring the application, it is noticeable that every message or email sent to the user is in Norwegian. The issue is understandable when the initial market is in Norway; however, for further expansion in the future, English or other Nordic language would be required and customised for each country. At the moment, English notification would be appropriate for improvement.

There are two types of notifications that users can choose either by sms or email. They can receive notice for outbidding auction or one day before their followed auction ends. In addition, when users register or are invited to be partners, the admin will send them several notifications and contracts. Therefore, in order to expand to other countries, English text is vital for both ways of communication.

The implementation consists of translating language on mailing services and saving user's preference whenever they change language on the main page. In order to save the language of the user, the database has to be modified to store the value. Specifically, a language column will be added to the user table with default value is 'no' as Norwegian. If the user selects English from the language dropdown, the data will be changed to 'en' in the database.

Based on the language code retrieved from the server, corresponding notifications will be sent to the user. For each type of notification, there will be two versions for English and Norwegian. The notification in sms can be written directly to the server whereas for email, similar templates for English have to be added in mailing service. The improvement offers the possibility to launch the application to different countries since English is a popular language in the world. In the future, the similar technique can be applied for other languages such as Danish or Swedish.

3.6 Challenges

Even though AWS has many advantages over Heroku, it would not be applicable for this project because of the shortage in developers and time limitation. It applies the horizontal scaling strategy where codebase is deployed to several environments for testing before going to production. Since the project code is pushed to Github whose connection with Heroku, whenever merging to master, it will deploy automatically a build for that branch. Overall, based on scale and other tools used in the project, Heroku is the wise choice at the moment.

Secondly, it is common that a developer will confront a number of problems when maintaining an existing application. First and foremost, the technology gap has contributed to a huge issue in this project. The project can run only from Node version 10 or lower while the newest version now is 17. The similar problem happens with the PostgreSQL version. The latest version was installed at the beginning; however, it does not allow running the command to generate typed files from the database. The reason is PostgreSQL version 14 does not accept md5 anymore while it is required to run the schema command. Consequently, a developer needs to reinstall the database and PostgreSQL many times in order to get the right version. It is challenging since the database postgreSQL version does not fix itself when installing a new one.

The difference in operating system also caused some issues while running applications. The predecessor developer wrote an application in Mac which has a different execution than in Windows terminal. There are some commands that could not be run in git bash which then demand to have an Ubuntu subsystem for Windows. Besides, it consumes lots of time to understand the system and purpose of some functions because of confusing naming and poor commenting in the codebase. The repositories are nested with each other throughout the system and the folder naming is ambiguous that cause uncertainty in finding the correct file to implement the task.

Lastly, it is necessary to balance between stakeholder, designer and technical possibility. Stakeholders would like to add many functionalities to the application and designers maintain the identical display for the company. Due to time limits and critical needs, it is essential to simplify the demand but still meet the requirement. Communication is a key to success when many parties have to discuss a lot with each other to come up with the most appropriate solutions. Despite those challenges, certain outcomes have been achieved with positive feedback from stakeholders.

4 Conclusion

First of all, the main purpose of the thesis was to improve user experiences by adding more features and to get familiar with all aspects of the application. The study was a success in exploring and understanding the code structure. Though there are still many hidden corners, it is feasible to locate the area which needs to be fixed whenever issue arises. Moreover, the deployment and debugging on Heroku has become a familiar action which supports resolving issues from user smoother. For instance, if a user has trouble logging in, some errors are visible on the dashboard which indicate the reason for failure.

Secondly, one of the crucial successes is to fulfil those tasks and learn about working flow in the organisation. It is not only about coding but the way to operate with other teams, such as considering requests from stakeholders and building a user interface based on sketches. There are several times that stakeholders change their demand and the components have to be rebuilt. Those have brought chances to learn more in agile development when a functionality is developed in sprint, it is tested and then developed again if needed. It was different from doing school projects where developers can have control on everything without considering other parties in application creation.

The last achievement would be in technical skills as the project has offered an opportunity to learn TS as well as React. It has been challenging to start with a type-defined language after working with JS for a long time. The code returns

many errors at the beginning since the argument has an incorrect type. In addition, writing queries in the backend to return dynamic data is an extensive learning which requires creating a complex formula to join many tables and extract data from others. Preparing test cases for each endpoint in the server is a crucial task in order to maintain the proper operations of fetching data. This is also new knowledge that is valuable to learn and achieve.

On the other hand, there are still possibilities for improving the applications. Since those new functionalities are rarely used, the developer can not take all demanding features to production. Instead, it would be similar to a testing version and preparing for a larger development in the future. For example, the table used for destroying objects has only two fields at the moment. However, if it would be decided that many objects can be scrapped in the future, more fields will be added to determine who and why they are disposed. Another improvement can be a check for phone number input. Users can add only numbers currently but there is no length limitation; therefore, a phone validation would be necessary to improve user experiences.

In a nutshell, a huge learning and improvement in technical and soft skills has been attained. Since time was limited, tasks had to be done promptly and deployed to production. However, most of the critical issues have been solved and the administrators are satisfied with the current state of application. There are some suggestions for further development which can be taken into consideration in the future or when the functionality has extensive usage and requires expansion.

References

Almeida, Fernando. 2019. 13 Reasons Why You Should Use Heroku in Your Next Project. Online. <https://dzone.com/articles/13-reasons-why-you-should-use-heroku-in-your-next>. Accessed 25 January 2022

Altexsoft. 2020. The Good and the Bad of TypeScript. Online. <https://www.altexsoft.com/blog/typescript-pros-and-cons/>. Accessed 25 January 2022

Baranowski, Wojciech. React JS: Advantages and Disadvantage. Online. <https://massivepixel.io/blog/react-advantages-disadvantages/>. Accessed 20 January 2022

Bolgurtseva O., Dreimanis G. 2020. Why You Should Choose TypeScript Over JavaScript. Online. <https://serokell.io/blog/why-typescript>. Accessed 25 January 2022

Borozenets, Michael. 2021. Why use PostgreSQL as a Database for my Next Project in 2021. Online. <https://fulcrum.rocks/blog/why-use-postgresql-database/>. Accessed 20 January 2022

Borrelli, P. 2019. Angular vs. React vs. Vue: A performance comparison. Online. <https://blog.logrocket.com/angular-vs-react-vs-vue-aperformance-comparison/>. Accessed 20 January 2022

Chen, Brandon. 2021. PostgreSQL vs. MySQL: What You Need to Know. Online. <https://www.fivetran.com/blog/postgresql-vs-mysql>. Accessed 23 January 2022

Codecademy. 2022. What is Node?. Online. <https://www.codecademy.com/article/what-is-node>. Accessed 23 January 2022

Deshpande, Amruta. 2019. 7 Advantages of ReactJS for Building Interactive User Interfaces. Online. <https://www.clariontech.com/blog/7-advantages-of-reactjs-for-building-interactive-user-interfaces>. Accessed 20 January 2022

Dev 2021. State Management Battle in React 2021: Hooks, Redux, and Recoil. Online]. <https://dev.to/workshub/state-management-battle-in-react-2021-hooks-redux-and-recoil-2am0>. Accessed 21 January 2022

Dignan, Larry. 2021. Top cloud providers: AWS, Microsoft Azure, and Google Cloud, hybrid, SaaS players. Online. <https://www.zdnet.com/article/the-top-cloud-providers-of-2021-aws-microsoft-azure-google-cloud-hybrid-saas/>. Accessed 25 February 2022

Docker overview. Online. <https://docs.docker.com/get-started/overview/>. Accessed 23 January 2022

Dubey B. 2018. Online. Difference between TypeScript and JavaScript. WWW document. <https://www.geeksforgeeks.org/difference-between-typescript-andjavascript/>. Accessed 19 January 2022

Dziuba, Anna. 2021. Why and When to Use Node.js: A Complete Guide for 2021. Online. <https://relevant.software/blog/why-and-when-to-use-node-js/>. Accessed 19 January 2022

EaseOut. 2021. React Components and Props. Online. <https://www.easeout.co/blog/2021-02-15-react-components-props/>. Accessed 25 January 2022

Facebook Inc. 2019. State and Lifecycle. Online. <https://reactjs.org/docs/state-and-lifecycle.html#adding-lifecycle-methods-to-aclass>. Accessed 20 January 2022

Freeman, Adam. 2019. Essential TypeScript 4: From Beginner to Pro. Apress; 2nd ed. edition

Gimeno, Alberto. 2019. Node.js multithreading: Worker threads and why they matter. Online. <https://blog.logrocket.com/node-js-multithreading-worker-threads-why-they-matter/>. Accessed 25 February 2022

Jain, Priyanka. 2019. Getting Started With Redux. Online. <https://www.c-sharpcorner.com/article/getting-started-with-redux/>. Accessed 25 February 2022

Kubiak, Grzegorz. 2021. Why use TypeScript? Could it be the best way to write frontend in 2021?. Online. <https://tsh.io/blog/why-use-typescript/>. Accessed 25 February 2022

Larson, Quincy. 2020. The 2020 Stack Overflow Developer Survey – 65,000 Devs Share Their Salaries, Top Programming Languages, and More. Online. <https://www.freecodecamp.org/news/stack-overflow-developer-survey-2020-programming-language-framework-salary-data/>. Accessed 25 February 2022

MDN 2021. Introduction to DOM. Online. <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction>. Accessed 20 January 2022

Mihaela. 2020. State Management Battle in React 2021: Hooks, Redux, and Recoil. Online. <https://dev.to/workshub/state-management-battle-in-react-2021-hooks-redux-and-recoil-2am0>. Accessed 25 February 2022

Nodejs User Survey Report. 2018. Online. <https://nodejs.org/en/user-survey-report/>. Accessed 25 February 2022

Nordea. Online. <https://www.nordearahoitus.fi/en/personal/about-nordea-finance.html>. Accessed 19 January 2022

opensource.com. What is Docker?. Online. <https://opensource.com/resources/what-docker>. Accessed 23 January 2022

Pavlutin, Dmitri. 2021. A Simple Guide to Component Props in React. Online. <https://dmitripavlutin.com/react-props/>. Accessed 25 January 2022

PostgreSQL History. 2017. Online. <https://web.archive.org/web/20170326020245/https://www.postgresql.org/about/history/#>. Accessed 21 January 2022

React 2020. Virtual DOM and Internals. Online. <https://reactjs.org/docs/faq-internals.html>. Accessed 20 January 2022

Red Hat. 2019. Online. <https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services>. Accessed 25 January 2022

Rigel Networks. 2017. Online. <https://www.rigelnetworks.com/using-virtual-dom-react-js-top-5-benefits/>. Accessed 25 January 2022

Rusev, Konstatin. 2019. What is Heroku and What is it Used For?. Online. <https://mentormate.com/blog/what-is-heroku-used-for-cloud-development/ >. Accessed 25 January 2022

Sharma, Ashok. 2018. Why You Should Use TypeScript for Developing Web Applications. Online. <https://dzone.com/articles/what-is-typescript-and-why-use-it>. Accessed 19 January 2022

Tate, Darlene. 2020. What is TypeScript? Pros and Cons. Online. <https://medium.com/@BuildMySite1/what-is-typescript-pros-and-cons-8dc5cdc3e78d>. Accessed 25 January 2022

Vorontsova, Marina. 2019. Disadvantages of using TypeScripts. Online <https://soshace.com/typescript-vs-javascript-vs-flow/>. Accessed 20 January 2022

Yamazaki, Shiori. 2020. Understanding Functional Components vs. Class Components in React. Online. <https://www.twilio.com/blog/react-choose-functional-components>. Accessed 25 February 2022

Yegulalp, Serdar. 2018. Why you should use Docker and containers. Online. <https://www.infoworld.com/article/3310941/why-you-should-use-docker-and-containers.html>. Accessed 23 January 2022