



Front-End Theoretical Learning and Practical Application



MACQUARIE University

Version Number	Date
1.1	24/05/2022
1.2	28/05/2022
1.3	30/06/2022
1.4	02/06/2022
1.5(Final Version)	05/06/2022

From: Prithivi Sharma, 45705704



Table Of Contents

1. Introduction(Context, Goals, Motivation).....	3
2. Initial Plan.....	4
3. Improved Plan.....	5
a. Changes.....	5
b. Reasons Behind Changes.....	5
4. Project Material.....	6-20
a. Practical Aspect.....	6-12
i. FreeCodeCamp.....	6-12
1. BootStrap.....	6-8
2. jQuery.....	8-9
3. SASS.....	10
4. React.....	11-12
b. Theoretical Aspect.....	13-20
i. LinkedIn Learning.....	13-20
1. HTML.....	13-15
2. CSS.....	16-17
3. JavaScript.....	17-18
4. UX.....	19
5. React.....	20
5. Keeping Track of Progress.....	21-24
a. Exercises.....	21
b. Small-Scale Projects(Portfolio Webpage).....	22
c. Quizzes.....	23
d. JSX Calculator(Deliverable).....	23-24
6. Deliverables.....	24-25
a. JSX Calculator.....	24-25
7. Reflection.....	26-27
a. What I learned.....	26
b. What I would improve on.....	26-27
c. Where to from here.....	27



Introduction(Context, Goals, Motivations)

This project is based on and inspired by one of the key areas of software development for which I had a strong level of passion and interest, web programming. Due to the extensive amount of knowledge and time required to understand this constantly evolving topic, the first project is based on one key aspect of web programming, front-end development. The main outcome from this project was to understand the basics of front-end development and to apply them through a practical application that emphasises on front-end development with data storage and display. This would involve a learning structure of practical content and theoretical content through online learning to allow a balanced learning on both sides of learning on the main topic and for ease of use and keeping track of my progress.

Goals

Upon learning and applying the skills and lessons learned throughout the period of this CPD project, the main goal was to understand the fundamentals of web programming, beginning with front-end development and be able to display them through my practical work. This would involve the development and successful operation of applications that are focused on front-end development and use some of data collection, storage and display, as seen from the practical deliverable of this project.

The second goal that I wanted to achieve by undertaking this project was to extend my professional portfolio. By doing so, not only would I be able to include my portfolio but also diversify it by showing my skill on other types of software development. As a result, this would allow me to have a portfolio which is well rounded, further reflecting my knowledge on other areas of software development.

Motivations

The main motivating factors that were responsible for me to pursue this particular aspect of software development was mainly due to the massive learning content and its endless possibilities in improving my knowledge as a professional software engineer. One such area that was a strong catalyst for me to focus on front-end development and gave me a strong starting point to learn more about the elements that are key to web programming. This involves back-end and full stack development which can help me build on my skills of data storage and touch on other areas like security and accessibility. Moreover, this topic would also allow to have a starting point with other software development topics that interconnect it like mobile application development and android application development.

As a result, this would allow me to also follow other career prospects where I could be able to learn more and gain professional first-hand experience on the application of front-end development in a professional environment.



Initial Plan

The initial plan to achieve the learning outcome and goals of this project comprised of following the same structure as I had originally set out. My initial plan focused on the entirety of web programming rather than front-end development. Due to my inexperience with the subject I wanted to learn more about, the practical learning of my initial plan would be covered through online learning comprising of exercises that would cover only a certain aspect of front end development let alone the whole of web programming. This can be further supported from my initial decision to complete an online course of responsive web design from freeCodeCamp and the career learning path course, “Become a web developer” from LinkedIn learning.

The deliverables that I had set out in my initial plan comprised of webpage projects like -: Tribute Page, Technical Documentation Page, Product Landing Page and a Portfolio Page. The timeline of learning throughout this time would consist of 3 weeks covering the practical aspects of learning, 2 weeks of theoretical learning and 1 week to complete the deliverables I had set as seen from **Figure 1.1**.

Parts	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Practical						
Theory						
Project						

Figure 1.1 : CPD Project Gantt Chart

Upon submission of my initial plan, I had noticed after the feedback given to me by my lecturer Carl that although this initial plan helped me learn some aspects of my overall learning outcome, I would not be able to cover all areas involved in front-end development but rather try to cover all elements involved in web programming which would take a longer period of time to cover. This in turn, would require me to start again with smaller and more concise topics and deliverables that would not only cover but also show the implementation of all areas of learning I set out to complete upon completion.



Improved Plan

Changes

The most notifying changes that were visible from the initial plan that was set out for the CPD project was in the learning scope and the content I would be covering even though the timeline of studying would remain the same time. This would include the learning content that I would cover on both the practical and the theoretical areas of my CPD project. Instead of responsive web design, the practical aspect would cover the area of Front End Development Libraries which would focus on learning and gaining a strong foundation on elements like Bootstrap, jQuery, SASS and React and the theoretical area of learning would be more concise with the course “Become A Front End Developer”.

The second major change that was also required and done was to the deliverables for my CPD project where I would make further changes to match the learning outcome and goals I had set out and the time constraint of completing it and displaying it for my CPD project presentation. As a result, after consulting with Carl, I reduced the number of deliverables from 5 to 1 which were a part of the Front End Development Libraries as some of these projects were only focused on the inputs from users rather than also delving on interactive and seamless data input, output, storage and display.

Reasons Behind Changes

The main factor that led to a change in the learning material that would be covered in both the practical section and the theoretical section was because they both lacked material to cover all aspects involved in web programming and therefore, were not a complete learning module that would help me cover my learning outcome and goals for this CPD project. This in turn, would allow me to focus on specific areas that are particularly related to front-end development and how to apply those for my deliverables. In conclusion, this would allow me to be accustomed to the base concepts of front end development while also being able to learn about the theoretical elements that are important to consider with practical applications of front end development.

The change of the output of deliverables from 5 to 1 was mainly decided upon because of the time constraints as well as the main objectives of the deliverables as they did not fully meet the learning outcome I had set for my CPD project. As a result, this would allow me to better understand the learning I would receive through my practical and theoretical areas, leading to better implementations on that one project that meets the main learning outcome I had set for my CPD project.



Project Material

Practical Aspect

The practical areas projected on this project would be covered through freeCodeCamp and would focus on the learning and application of front-end libraries directly responsible for being able to implement different styles within a website or application with more efficiency. This would include being accustomed to and working alongside Bootstrap, which contain many different components that help developers style their websites to accommodate different designs. In addition to that, the use of jQuery would also allow me to ensure that my work would be able to produce the required output within all browsers while also helping me with learning how to simplify the interaction within the elements of JavaScript and HTML. Moreover, adding many different forms of logic within CSS and being able to extend them with SASS was another area that would allow me to learn not only how to add new features that are not provided by CSS but also be able to maintain my style sheets for projects involving front-end development and web programming in general. As a result, I would be able to apply these new learnings when continuing with React which was another major learning topic involved with this area, exposing me to JSX which would be the building block for my main deliverable of this project itself.

freeCodeCamp

Bootstrap

Bootstrap is a framework that is used to design responsive web pages and applications. This front-end framework is built on the base languages of HTML, CSS and JavaScript and is responsive, sites. As a result, by learning this framework and its support to other plugins from JavaScript, Bootstrap would help developers make projects that are flexible to different requirements. It would do so through its offering of different premade customisable components that can be added to a developer's user interface without a lot of difficulty. Therefore, learning its usage within the spectrum of front-end web development would help developers immensely in applying it in the professional field of web programming. The many different areas of learning with Bootstrap in this course included learning how to create different types of buttons for a website, creating a grid element and add elements inside and style them without the use of CSS and how to make webpage more interactive with the different classes I can use to create more depth within a certain area of my project.



One of the key areas where I learned more about the importance of bootstrap was its use in the creation of different elements and changing their appearance based on the purpose they serve to the webpage or application that is being made by a developer. One example of this was the implementation of creating a button element and formatting its style to meet its requirement. This can be seen in *Figure 2.1* which show how I had used Bootstrap for the first time to create a button element with default classes which would further expand onto other types of buttons and buttons with additional styling features like icons as displayed in *Figure 2.2* and *Figure 2.3* . With the introduction of these new concepts, I was able to familiarise myself with Bootstrap and what it has to offer in making the user interface of a front-end developer's project more interactive and to better convey what certain elements are used for in the project.



Figure 2.1 : Creating a new bootstrap button



Figure 2.2 : Creating a bootstrap block button

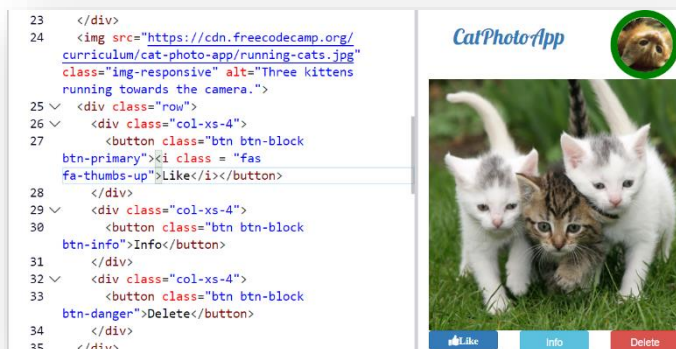


Figure 2.3 : Adding an icon to a bootstrap

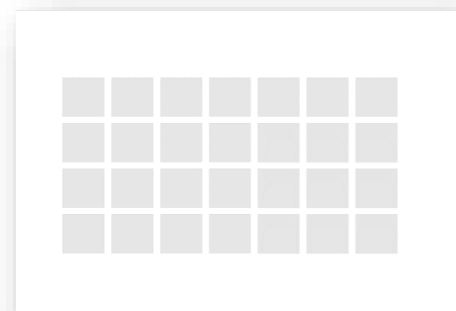


Figure 2.4 : Grid filled with buttons

Another area of learning I had through Bootstrap was how to create new elements, most particularly grids. Using grid elements allowed me to learn more about how to implement a grid element and how to nest other elements as a part of that element. As a result, this helped me reflect on how I could organise and format areas of my project better to provide a user interface that is easier to follow for users. This is further evident from *Figure C* where instead of structuring buttons in a block-like structure, a grid structure was implemented to better position those buttons and in return, make it more visually appealing to users. In addition, the grid structure of Bootstrap allowed me to make my projects more flexible due to its responsive 12 column grid structure, allowing me to add many elements without taking too much space from my webpage, further highlighted in *Figure 2.4*.



The use of different classes of Bootstrap in many other elements to make an area of a webpage or an application to a user played a significant role in helping me get more familiar with the many facets of the implementation of Bootstrap. This is evident from the different functionalities I implemented using

Bootstrap which included creating a row and also creating a Bootstrap well which would create more emphasis towards a particular area of the application or the webpage that is being developed as seen in *Figure 2.5* and *Figure 2.6*. Due to that, I was able to understand the application of Bootstrap on many other elements used for webpages and applications and learn how seamlessly they are integrated and how useful they are in ensuring that the work of a developer is well organised and easy to understand while also being visually interactive.

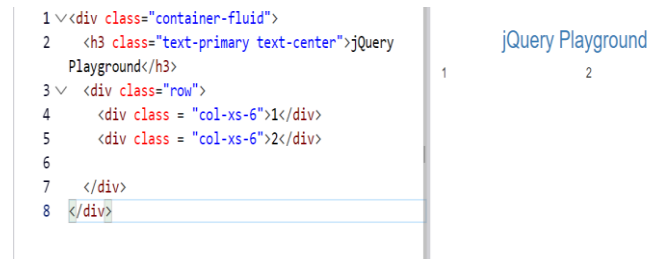


Figure 2.5 : Bootstrap row creation

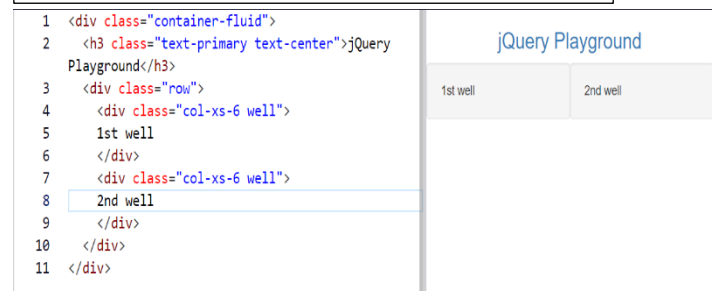


Figure 2.6 : Bootstrap well

jQuery

jQuery is one of the most used libraries within Front-end development and is directly responsible for making the process of developing complex code from Javascript much easier. This can be seen from its ability to offer shorthand solutions for problems in JavaScript that are a lot more complex. This includes its ability to create, edit, delete or clone certain elements from one page. Moreover, the majority of the libraries that are involved in front-end development in one way or another have dependencies on jQuery and require jQuery for their operations, further adding to their importance within the operations of front-end development. In this project in particular, I was able to apply jQuery and learn more about how to use them, especially with areas such as targeting elements through CSS classes and changing operations of CSS and non-CSS properties of elements.



Out of the many actions that developers can execute using jQuery onto their applications and websites, the ability of jQuery to change the behaviour of a particular element through the addition and deletion of CSS classes was the one that allowed me to truly understand the importance of jQuery in increasing the ability of elements to be more interactive. This can be seen from *Figure 3.1* and *Figure 3.2* that display the use of jQuery in creating and removing CSS classes that manipulate the behaviour of one or more elements. This includes adding and/or removing classes that are responsible for changing the way elements are viewed on the website or application. By implementing these actions using jQuery, I was able to see how jQuery is extremely helpful to front-end developers as it allows them to perform complex actions such as controlling the way elements operate with much more ease and lets them make changes without confusing them about where a particular element is situated within their webpage or application.

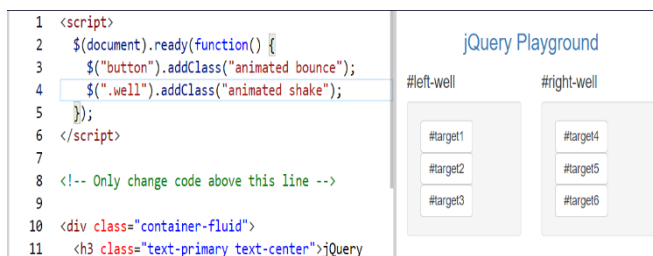


Figure 3.1: Adding a CSS class to a button using jQuery

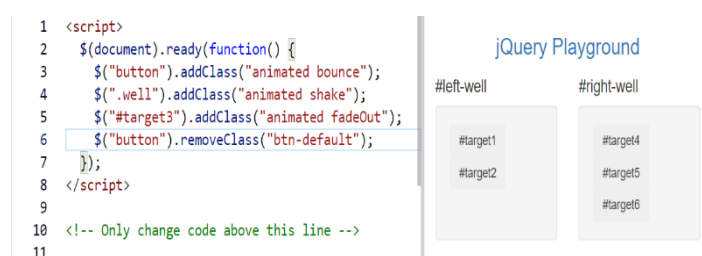


Figure 3.2: Removing CSS classes using jQuery

Apart from CSS classes, the ability of jQuery to alter CSS and non-CSS properties of elements can be further seen through other functions it performs that directly affect how elements are being presented and what they can do. Examples of their use are evident in *Figure 3.3* and *Figure 3.4*, showing an implementation of changing the CSS of an element and disabling the clicking of a button as well as many other functions that can not only edit the content within an element but also be able to remove them entirely. This allowed me to reflect so much more on the importance of jQuery and how it can be used to not only make the work of developers much easier but to also allow them to perform actions with certain elements of their webpage without requiring the creation of a CSS file which normally handles the representation and actions of elements but rather finding an alternative which is less time consuming and easier to modify in the future.

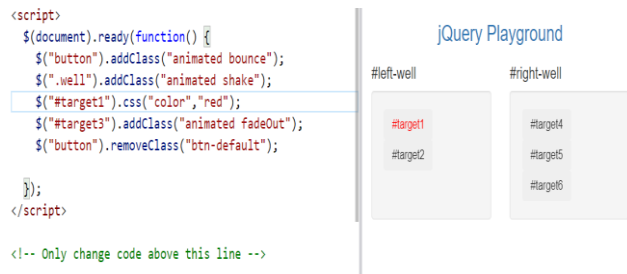


Figure 3.3 : Changing CSS of an element using jQuery

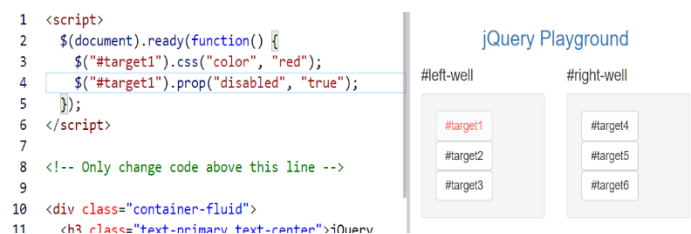


Figure 3.4 : Adding a CSS class to a button using jQuery



SASS

SASS, a pre-processor language that compiles into CSS is another crucial area of learning that is included in this project. It's ability to compile itself into CSS to create stylesheets which are easier to manage and more stylish make it a framework which is integral in helping developers sustain large codebases they have more productively and neatly. It does so with its features to store data in variables, using loops to change multiple elements or to continue adding a style until a condition has been met.

By applying variables to store data using SASS, I was introduced to a completely different way to store information used for styling my project. Their implementation as seen in *Figure 4.1* allowed me to learn a new way to make a big change without needing to reference the same style and making readers of my work confused. Furthermore, by the usage of variables to define a certain feature, I was able to learn a new way to keep my work organised and write less CSS code to cover all elements within my future projects. Similar to variables, mixins were another area that allowed me to organise and reuse my CSS code, allowing me to add that particular area wherever I wanted without repeating it as displayed in *Figure 4.2*

```
1 <style type='text/scss'>
2
3 $text-color: red;
4
5
6 .header{
7   text-align: center;
8 }
9 .blog-post, h2 {
10  color: $text-color;
11 }
```

Learn Sass

Some random title

This is a paragraph with some random text in it

Header #2

Here is some more random text.

Here is another header

```
1 <style type='text/scss'>
2
3 @mixin border-radius($radius){
4   -webkit-border-radius:$radius;
5   -moz-border-radius:$radius;
6   -ms-border-radius:$radius;
7   border-radius:$radius;
8 }
9
10 #awesome {
11   width: 150px;
12   height: 150px;
13   background-color: green;
14   @include border-radius(15px);
15 }
```



Figure 4.1: Use of SASS variables

Figure 4.2: SASS mixin use

Another area where I was able to write condensed CSS code to cover every element I wanted was through SASS loops. Through SASS loops, I was able to change multiple elements and apply a style until a condition has been met. This in turn, not only helped me new ways to organise my project more effectively but also strengthened my knowledge about how to better format the styles of my code. This can be seen from the practical exercises I completed in *Figure 4.3*, where I used a while loop to change the font of 5 different elements until a particular condition was met.

```
1 <style type='text/scss'>
2
3 $x: 1;
4 @while $x < 6 {
5   .text-#{$x}{font-size: 15 * $x;}
6   $x: $x + 1;
7 }
8
9
10 </style>
11
12 <p class="text-1">Hello</p>
13 <p class="text-2">Hello</p>
14 <p class="text-3">Hello</p>
15 <p class="text-4">Hello</p>
16 <p class="text-5">Hello</p>
```

Hello

Hello

Hello

Hello

Hello

Figure 4.3 : SASS while loop



React

React is a JavaScript library that is widely used in the creation of webpages or applications. This can be directly attributed to the many benefits it provides as a library with high speed, its usability through its ability to reuse components. Apart from that, React offers developers better code stability through its top-down structure of data flow and also allows developers to render their work more quickly along with a developer toolset. As a result, this allows developers who have just begun to get familiarised with React to learn it much more quickly. From engaging in the practical learning of React with this practical course, apart from learning about using its components, the biggest areas of learning for me came from its implementation in managing and displaying data, different ways elements are rendered using React components and the different ways to render the state of a component.

The implementation of unidirectional data flow uses state data that is needed to render a state prop is the starting point of the management and display of data through the use of React. Unidirectional data flow acts as a base for other major functions such as `componentDidMount()` which acts more as a function called at the endpoint of an API. This can be seen from *Figure 5.1* and *Figure 5.2* where each of them shows the implementation of unidirectional flow and `componentDidMount()` with some elements of unidirectional flow being used in the execution of lifecycle methods of user interfaces. By being able to compare these two methods upon execution, I was able to learn more about the many different ways data is stored and rendered in React based on which method is required for what type of rendition.

```
1 class MyApp extends React.Component {
2   constructor(props) {
3     super(props);
4     this.state = {
5       name: 'Prithivi'
6     }
7   }
8   render() {
9     return (
10      <div>
11        /* Change code below this line */
12        <Navbar name={this.state.name}/>
13        /* Change code above this line */
14      </div>
15    );
16  }
17 };
18
19 class Navbar extends React.Component {
20   constructor(props) {
21     super(props);
22   }
23   render() {
24     return (
25       <div>
26         /* Change code below this line */
27         <h1>Hello, my name is: {this.props.name}</h1>
28         /* Change code above this line */
29       </div>
30     );
31   }
32 }
```

Hello, my name is: Prithivi

```
1 class MyComponent extends React.Component {
2   constructor(props) {
3     super(props);
4     this.state = {
5       activeUsers: null
6     };
7   }
8   componentDidMount() {
9     setTimeout(() => {
10       this.setState({
11         activeUsers: 1273
12       });
13     }, 2500);
14   }
15   render() {
16     return (
17       <div>
18         /* Change code below this line */
19         <h1>Active Users: {this.state.
20           activeUsers}</h1>
21         /* Change code above this line */
22       </div>
23     );
24   }
25 }
```

Active Users: 1273

Figure 5.1: Unidirectional flow

Figure 5.2: Implementation of `componentDidMount()`



The ability to render nested components using component composition was a significant area of learning for me as I was able to discover the depths of rendering which can be executed by components within a React framework. An example of this is seen in *Figure 5.3* shows React's capabilities in being able to access elements that are nested within another nested component. By being able to find a different way to render components, I was able to learn how to divide my work into different areas. As a result, this would help me separate my project's work and recognise areas of my work responsible for application logic and other area's focused on other functionalities. This would allow me to keep my work organised and also allow me to find faults with a lot more ease.

```
const Fruits = () => {
  return (
    <div>
      { /* Change code below this line */ }
      <TypesOfFruit />
      { /* Change code above this line */ }
    </div>
  );
};

class TypesOfFood extends React.Component {
  constructor(props) {
    super(props);
  }

  render() {
    return (
      <div>
        <h1>Types of Food:</h1>
        { /* Change code below this line */ }
        <Fruits />
      </div>
    );
  }
}
```

Types of Food:

Fruits:

- Apples
- Blueberries
- Strawberries
- Bananas

Figure 5.3 : Nested component rendered

While completing this section of the course, the concept of rendering elements using React components was one vital area where I was able to learn more about the flexibility that React provides when rendering elements, either dynamically or through conditions. This can be further emulated from the execution of the exercises seen in *Figure 5.4* and *Figure 5.5* where elements are rendered through some sort of action or condition being triggered. By being able to discover new ways to render elements, I learned about new and different ways to manipulate what is being rendered to the user and how to achieve that using other means. This in turn, would help me decide what approach is the best when trying to render elements in my current and future projects and allow me to build upon new concepts I have learned.

watch movie, play games, study

Create List

My "To Do" List:

watch movie, play games, study

Create List

My "To Do" List:

- watch movie
- play games
- study

Figure 5.4: Dynamic rendering elements

```
13 }
14 render() {
15   // Change code below this line
16   if(this.state.display == true){
17     return (
18       <div>
19         <button onClick={this.toggleDisplay}>
20           >Toggle Display</button>
21         <h1>Displayed!</h1>
22       </div>
23     );
24   }else{
25     return(
26       <div>
27         <button onClick={this.
28           toggleDisplay}>Toggle Display</button>
```

Displayed!

```
13 }
14 render() {
15   // Change code below this line
16   if(this.state.display == true){
17     return (
18       <div>
19         <button onClick={this.toggleDisplay}>
20           >Toggle Display</button>
21         <h1>Displayed!</h1>
22       </div>
23     );
24   }else{
25     return(
26       <div>
27         <button onClick={this.
28           toggleDisplay}>Toggle Display</button>
```

Figure 5.5 : Rendering elements through the use of conditions



Theoretical Aspect

The theoretical aspect of this project involved the use of LinkedIn learning and would focus on the essential understanding of the fundamental concepts involved in HTML, CSS and Javascript. These courses would refresh and also in some cases, enhance my knowledge on these 3 languages and let me develop solutions with a broader understanding of the key concepts that all of these base languages offer. In addition to getting re-familiarised with topics I had experience with, I was also to learn more about conceptual areas involved in web design and front-end programming through courses focusing User experience and user interface like accessibility and responsive layout. As a result, this allowed me to approach my projects with a newer perspective with consideration to many new factors which I had not associated with the development of front-end projects before. Furthermore, I would also learn more about the other areas which are essential in keeping track of my progress through the use of GitHub when using ensuring that I am able to front-end applications and/or webpages with the most advanced tools and their functionalities at my disposal and being able to make and track changes of my work with much more efficiency and clarity.

HTML

HTML(Hypertext Markup Language) as its name suggests, is a markup language used for front-end development that defines the structure of web pages. This can be seen from the markups provided by HTML, known as tags and elements which give a structure to a web page. Moreover, the use HTML also provides a strong base on many different types of content is built and shared on the web by following a certain structure. This allowed me to be able to get myself refamiliarized with the concepts I have previously learned about HTML while also learning about different concepts I had not known about. This included new learning experiences about different types of content and processes originating from and ranging to media and different form elements as well as with aria labels to provide information about a certain element within a webpage and how URL paths are defined if developers are looking to see how certain changes look on a webpage or for testing purposes. By being able to receive known and newfound knowledge of HTML throughout this course, I was able to strengthen my understanding of HTML as a whole which in turn, I could directly use for future projects where HTML would play a big role in its implementation.



From the learning of different elements that are related to a form of element that developers are already aware of, I was exposed to different ways media as well as navigation are presented with the use of HTML, ultimately improving my approach and knowledge on certain areas that tie into front-end development. This included being able to learn about and see the application of new elements that are related to media such as captions and subtitles that are included in video and audio as applied in *Figure 6.1* as well as being able to embed other forms of media with the help of HTML, further shown in *Figure 6.2*. In addition to that, learning about new elements that are responsible for data intake such as with forms was another where I had an increased level of newfound understanding about the endless capabilities that HTML has. One example is visible on *Figure 6.3* with the introduction of element types such as search and phone number which are built in within HTML that allows developers to be able to receive different types of information without too much effort. From these new areas of learning, I was able to find out about new elements that I can implement within my system as well as being able to incorporate different types of media from other websites onto my own projects through the use of HTML.

```
<track src="https://s3-us-west-2.amazonaws.com/s.cdn.io/10558/moonwalk.vtt"
      kind="captions"
      label="english"
      srclang="en"
      default>
<p>This would be a video of a moonwalk, if your device supported
playing this video.</p>
</video>
```

Figure 6.2: Addition of English subtitles in video

```
<iframe width="560" height="315"
src="https://www.youtube-nocookie.com/embed/0Gr1XSyxZy0" frameborder="0"
allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture"
allowfullscreen></iframe>
```

Figure 6.3: Use of iframe to add media from other websites

```
<form action="received.html" method="get">

  <label for="name">Name</label>
  <input id="name" name="name" type="text">

  <label for="email">Email</label>
  <input id="email" name="email" type="email"
    placeholder="you@example.com">

  <label for="password">Password</label>
  <input id="password" name="password"
    type="password">

  <label for="search">Search</label>
  <input id="search" name="search"
    type="search" placeholder="#128269;">

  <label for="phone">Phone Number</label>
  <input id="phone" name="phone" type="tel">
```

Our Form

Name

Email

Password

Search

Phone Number

Figure 6.3 : Form including different element types



The importance of ARIA roles was another topic which provided me with better insight on additional functionalities that can be added to an HTML file within a project to ensure element that are added within the HTML document of developers are understood and rendered by the browsers accordingly. One main area where ARIA roles might be useful are towards accessibility as they are useful to help users who have disabilities, allowing developers to be able to create websites that are accessible to disabled users as well. Moreover, the importance of ARIA labels is further highlighted from how their use affects the accessibility tree that is accessed with the developer tools within a browser, allowing elements that contain other elements to remain hidden with the use of an ARIA label defining them all in one container, rather than multiple containers as highlighted in *Figure 6.4*. By learning about ARIA labels and how they are used within HTML documents, I was able to learn a lot more about how they have a massive importance in ensuring inclusion within the work of developers and how developers like myself need to consider that when moving forward with future front-end development projects.

```
1 <h1 aria-label="Hello World">
2 <div class="grid">
3   <span>H</span>
4   <span>e</span>
5   <span>l</span>
6   <span>l</span>
7   <span>o</span>
8
9   <span>w</span>
10  <span>o</span>
11  <span>r</span>
12  <span>l</span>
```

Figure 6.4: Aria label implementation

The pivotal role URL paths take can be seen from their use by developers for testing and reviewing their work on big websites when they are working with a copy of it locally. Their use is unlimited as they can be used to redirect to CSS files and Javascript files to retrieve files and/or information. From being able to find differences between absolute and relative URL paths as shown in *Figure 6.5*, I was able to see how relative paths provide developers with more freedom to look at changes they have made and how they can be improved. Upon learning differences between relative and absolute URL paths, I was also able to find out and learn about the different versions that are associated with relative URL as seen in *Figure 6.6*. This in turn, helped me strengthen my knowledge on how to use relative URL when navigating amongst webpages in a website and how to use them to point to other files as well.

Absolute URL

<https://awesomedogs.com/about>

Relative URL

/about

Figure 6.5 : Absolute and relative path differences

```
Awesome Dogs
├── index.html
├── prices.html
├── people.html
├── contact.html
├── blog
│   ├── march-9.html
│   └── march-10.html
├── images
│   ├── logo.gif
│   └── happy-dog.jpg
├── CSS
│   └── styles.css
└── /images/logo.gif
    └── ../images/logo.gif
```

Figure 6.6 : Different forms of relative paths



CSS

CSS or Cascading Style Sheets is a language that is used to style and design a website. It's main involvement within front-end development is from its ability to describe how HTML elements that are made by developers are to be presented onto the screens of their users. After completing the course on CSS, there were many areas involved in layout and application which not only helped me better understand the features of CSS, but also directly influenced how my projects would be presented. The topics of learning where I was able to understand more included the grid layout where I found out about different ways content within webpages can be structured and displayed to users which allowed me to explore the application of background image as well. Along with that, the use of advanced selectors was another area of new knowledge of CSS for me as this allowed me to find a new way to select particular areas of a webpage to make changes to specifically. By learning these new concepts in a language that I am already familiar with, I got to know more about how I can use it to better structure my work, particularly towards the design aspect.

The topic about grid layouts were one of the areas which had a major impact on my knowledge and application of CSS concepts within projects. One of the areas which helped strengthen my knowledge was the different ways grids can be presented and altered to meet a specific design requirement. One example included the ability of developers to change the structure of a grid as seen in *Figure 7.1*. This new discovery allowed me to learn more about how useful grids in allowing developers to structure information in a more succinct manner as seen from *Figure 7.2* where the use of grid properties allowed me to structure the section of my webpage much more clearly and made it easier to follow. From the introduction to two different types of

grids, implicit and explicit, I was able to self-reflect on the capabilities of CSS and learn a lot more about the functionalities that are associated with grid cells and how flexible they are depending on what is being included on the grid. One prime example is implicit grids being able to accommodate grid items that are greater than that of the grid cells, allowing for automatic placement. From learning this new concept, I was able to understand and apply this technique to improve the overall look of my project and learn new ways to design websites with more efficiency as this would allow me to better divide HTML contents within the main website that will be accessed by the user.

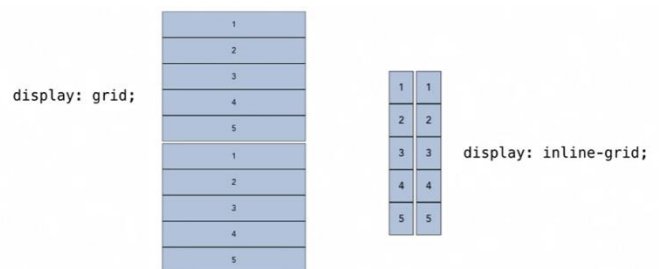


Figure 7.1: Normal and in-line grid display



Figure 7.2 : Rendered element with grid display



Advanced selectors were another area of CSS which I had not discovered when initially learning it. From understanding its use to make matching patterns based on the relationship between nested elements and how that is achieved through descendant and children selectors as shown in *Figure 7.3*, its implementation that allowed me to be able to control many elements styles without a lot of effort. This can be seen at *Figure 7.4*, showing how their use allows many changes to be condensed, allowing for webpages to be altered without requiring a lot of code due to selectors being able to handle style operations of multiple nested elements that are specified in the CSS file. By being able to learn about and implement selectors, I was able to self-reflect on better organising CSS files and optimising their styling operations, especially between elements that are nested.



Figure 7.4: Italic style added to paragraphs below heading type 3

```
.item-details h3 + p {  
  font-style: italic;  
}
```

Figure 7.3: Code to change font style of paragraphs below headings

JavaScript

JavaScript is the third and final language that is defined as the most important language used in front-end development. From its responsibilities ranging from handling the user-side actions that influence web-page behaviour like receiving and storing data as well as identifying certain areas of web-pages for developers so they can understand the structure of their page with more ease. The new elements of learning that I had from undertaking this course was through the concepts such as Event Listeners and Query Selectors. From learning more about these areas of learning that are associated with JavaScript, I was able to learn more about their application and how helpful they are in finding areas within my document you can identify in your work to make changes and adding more functionalities to make my project faultless.

Event listeners were a strong area of learning within the study of JavaScript as throughout this course, I was able to learn more about its advanced functionalities and their implementation within projects. This can be seen as seen from its use in *Figure 8.1* where upon clicking at grid cells, I was able to change their appearance and show how event listeners can help add advanced functionalities on front-end development projects. This use



of event listeners influenced some elements of the deliverable of my CPD project as seen in *Figure 8.2*, further highlighting how useful event listeners are in updating data that is being presented. As a result, I was able to learn a lot and self-reflect about how influential JavaScript operations are in increasing the efficiency of websites and/or applications, resulting in a more immersive experience for users.

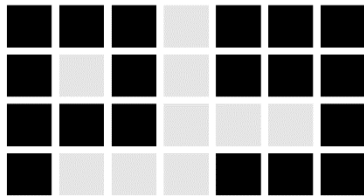


Figure 8.1: Grid cell colouring exercise using event listener

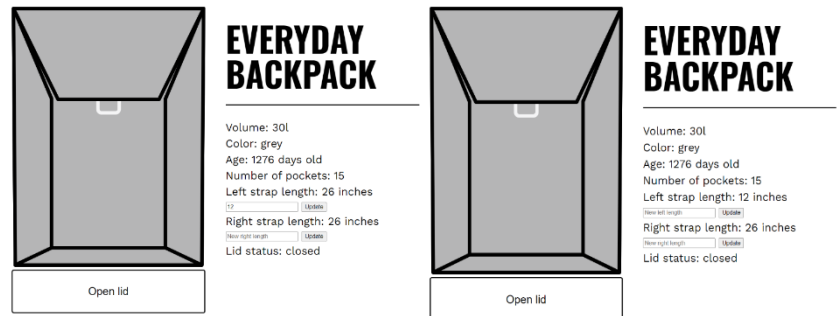


Figure 8.2: Element change before and after using event listener

Query Selectors were another area of self-reflection for me when analysing the different functionalities that JavaScript provides. This was mostly evident from how they were able to identify certain areas of a webpage being rendered to the user in the console section of the developer tools of the web browser.

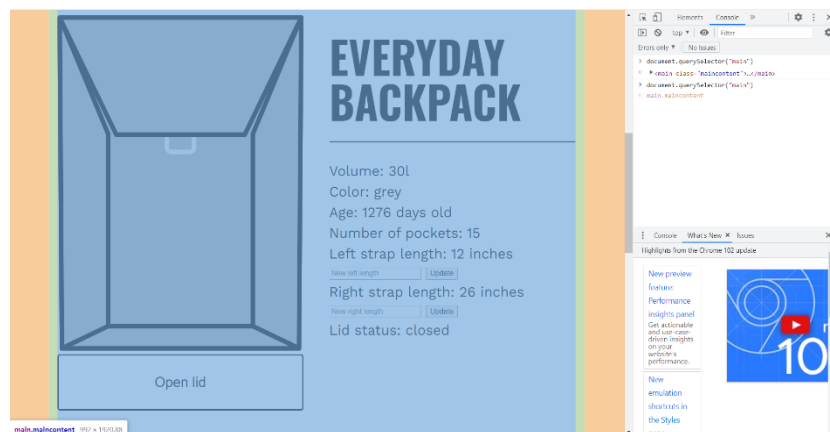


Figure 8.3 : document.querySelector() on browser console

Doing this, as shown in *Figure 8.3* in the console of developer tools of my project allowed me to use the same technique in my front-end project and better structure certain elements to allow users to better follow the information I was trying to convey. This in turn, helped me better organise my work and present it with better order and self-reflect on the different ways to identify where certain elements are located within my project.



UX

The one area which was a completely new concept in front-end development to me before starting this project was User Experience. By understanding the concept of User Experience within the context of front-end development, there were many areas of understanding and evaluation I was able to take-away from this experience. This included the understanding of elements within front-end programming which are used to help users with disability navigate through websites and application such as voice recognition software and screen readers and magnifiers. By being able to explore these two areas in particular, I was able to self-reflect not only on how to be inclusive of my work to other audiences but also on how to better organise areas of my project to accommodate other audiences, further influencing how I approached my deliverable for this project as well.

The common usage of voice recognition in helping users with mobility limitations to navigate through websites and applications was one of the main areas of focus throughout this course. From learning more about their primary functionalities, dictation and command and control, I was able to self-reflect on the importance of ensuring the process of navigation is made easy for all types of users. This was prevalent from learning more about the key responsibilities they each share with dictation handling operations like writing documents, emails and filling in information and command and control being responsible for dragging files, switching programs for users and clicking buttons in their systems or in websites. After looking at how Dragon NaturallySpeaking implements these same controls for users, I was able to self-reflect on where I position important elements of my application as by doing so, I would allow navigation to be easier for users who have difficulties in that area either through limitations or disabilities.

The application of screen readers and screen magnifiers have been a commonality amongst operating systems as they have been added to improve the experience of users for operating systems and websites. By understanding the many different factors that contribute to the use of screen reads and magnifiers such as blurred vision due to cataracts or a user having some form of disability and how they are implemented by users, this allowed me to re-evaluate my understanding of elements and what to include within the user interface and how to style it to ensure that users are able to easily understand what I have included into the project, making way for a great experience overall. As a result, I was able to self-reflect on the significance of design and styles for front-end projects as by having a consistent styling design that can be understood quickly and is easy to read, I would ensure that my project is succinct and is easy to follow.



React

From learning React as a part of this course, apart from being reintroduced to concepts I was already aware of, there were some areas that still allowed me to self-reflect on my knowledge and skills within the framework of React. This area of realisation came from the introduction to the functionalities of `useEffect` and `useReducer` of React and how they contribute to developers being able to create applications with advanced functionalities. As a result, I was able to learn more about how to better optimise my solutions and be able to perform multiple actions without needing too much effort.

The implementation of `useEffect` and `useReducer` were vital in allowing me to learn more about their significance in the operations of React. This can be seen in *Figure 9.1* and *Figure 9.2*, where in these exercises, the instructor was able to perform two actions through the nesting of another action within `useEffect` and be able to fetch data using that same function. From this action, I was able to identify why `useEffect` is a useful tool within React due to its ability to directly update the DOM (Document Object Model) and in some cases being able to collect data. In addition, the `useReducer`'s ability to reduce complication of the state of a particular element, I was able to learn about a tool that keeps track of a state of an element as a project gets more complex. By learning how these functionalities are implemented, I was able to not only extend and evolve my knowledge on React, but also recognise the importance of these functionalities and what scenarios they can be applied by developers to make their front-end project more optimal.

```
5 import ReactDOM from "react-dom";
6 import "./index.css";
7
8 function Checkbox() {
9   const [checked, setChecked] = useState(false);
10   useEffect(() => {
11     alert(`checked: ${checked.toString()}`);
12   });
13 }
```

Figure 9.1: `useEffect()` implementation to nest an action

```
8 function GitHubUser({ login }) {
9   const [data, setData] = useState(null);
10   useEffect(() => {
11     fetch(`https://api.github.com/users/${login}`)
12       .then(res => res.json())
13       .then(setData)
14       .catch(console.error);
15   }, []);
16
17   if (data) {
18     return <div>{JSON.stringify(data)}</div>;
19   }
20   return null;
21 }
22
23 function App() {
24   return <GitHubUser login="moonhighway" />;
25 }
```

Figure 9.2: `useEffect()` implementation to fetch data

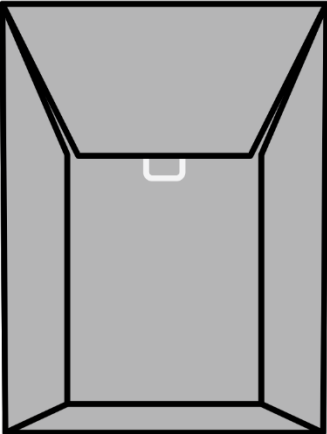


Keeping Track of Progress

Exercises

One of the many different ways that helped me keep track of my progress throughout the duration of my CPD project was through the use of exercises on both main aspects of the project. Exercises were a commonality within the practical aspect of my project as they were used as the main means to move forward and complete each topic section within the practical section of my project. One of the most notable ones that ensured that my progress to achieving my goal of understanding front-end development included using event listeners and rendering elements from many different means. Exercises like this allowed me to not only self-reflect on their implementation and significance but also be able to check my capabilities and evaluate if I was meeting my goals for this CPD project.

The implementation of event listeners to update the value of a particular element that is presented to the user from their input as shown in *Figure 8.2* was one of the exercises that allowed me to not only learn new advanced functionalities involved in front-end development. As a result, the understanding of this topic and its practical application allowed me to conceptualise its logic onto my CPD project deliverable. As a result, this allowed the implementation of data storage and display much easily in the deliverable of my CPD project. From learning more about event listener's advanced functionalities and implementing them, I was able to self-reflect on my approach towards how to collect, store and output input data in my future front-end projects, further enhancing my knowledge on base concepts that are a part of font-end projects.



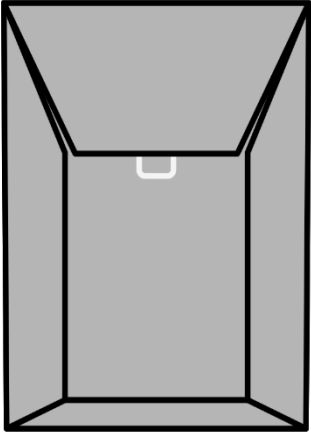
EVERYDAY BACKPACK

Volume: 30l
Color: grey
Age: 1276 days old
Number of pockets: 15
Left strap length: 26 inches

Right strap length: 26 inches

Lid status: closed

Open lid



EVERYDAY BACKPACK

Volume: 30l
Color: grey
Age: 1276 days old
Number of pockets: 15
Left strap length: 12 inches

Right strap length: 26 inches

Lid status: closed

Open lid

Figure 8.2 : Element change before and after using event listener



Small-Scale Projects(CSS Portfolio Webpage)

Throughout the duration of this project, I participated in small-size projects that applied the knowledge I had learned. These projects allowed me to identify areas that I was strong at and what I needed to improve on, further increasing my knowledge on these areas. One prominent project was the CSS Portfolio Webpage which I conducted as a part of learning CSS with LinkedIn. By undertaking this project, I was able to apply skills like the use of the grid display to be able to showcase my new learnings and find their advantages in better styling front-end projects.

The use of the grid display can be further seen from the work experience section of the CSS portfolio as seen in *Figure 7.2*. With the inclusion of the grid display, I was able to recognise what design styles suit certain elements in a webpage or application as seen from the use of overflow for the project section of this webpage in *Figure 10.1* and its corresponding code in *Figure 10.2*. This in turn, would further increase my knowledge on different design styles that are involved in CSS implementation and how to apply those skills for the deliverable of my CPD project and ensure it is visually efficient in display.

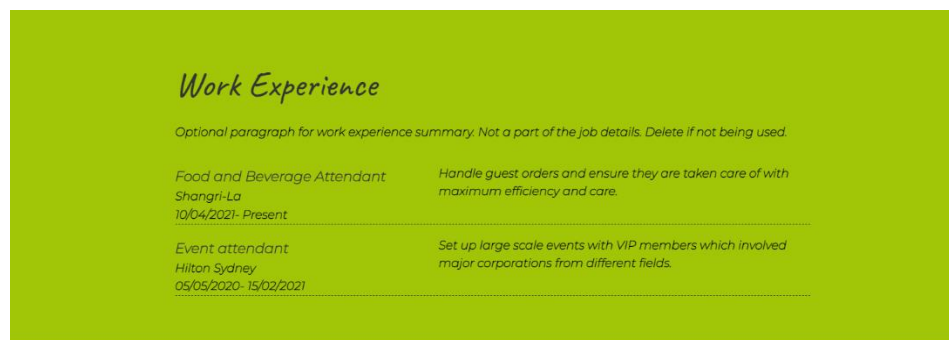


Figure 7.2



Figure 10.1 : Overflow output

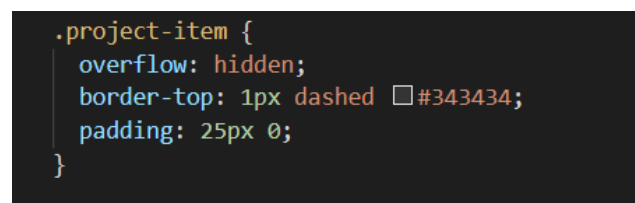


Figure 10.2 : Overflow implementation



Quizzes

Quizzes were an integral part of my learning within the theoretical aspect of my learning for this CPD project. Quizzes came as a part of majority of the courses I had learned and students of said courses were required to achieve a full mark to proceed to the next part of the course, otherwise it would be incomplete. From participating in these quizzes, I was able to reflect on the knowledge I had learnt and more importantly explore areas where I had weaknesses, allowing me to further refine my knowledge and improve my skills.

JSX Calculator(Deliverable)

The main deliverable of this project, the JSX calculator was another main indicator that was used to track my progress in achieving the goals of my CPD project. This was mainly done from the implementation of different functions for my CPD project like the calculator's end-result along with the welcome page and the history section. By being able to add these functionalities, I was able to not only show my new changes I have made to achieve the goal I had set out for the deliverable and the CPD project itself but also incorporate new areas of learning onto my work, resulting in additional functionalities within the deliverable itself.

The implementation of the interface of the calculator and generating its output, for instance as seen in *Figure 11.1* was through the use of grid elements that I had learned through Bootstrap. By using a grid based system instead of an input text for users to input numbers, I was able to add more interactivity where users were able to make calculations by simply clicking on items rather than typing the entire expression they would like to output. From the establishment of this interface, I was able to show how I used a concept I had learned previously and applied it to my deliverable to provide more interactivity and freedom to users.

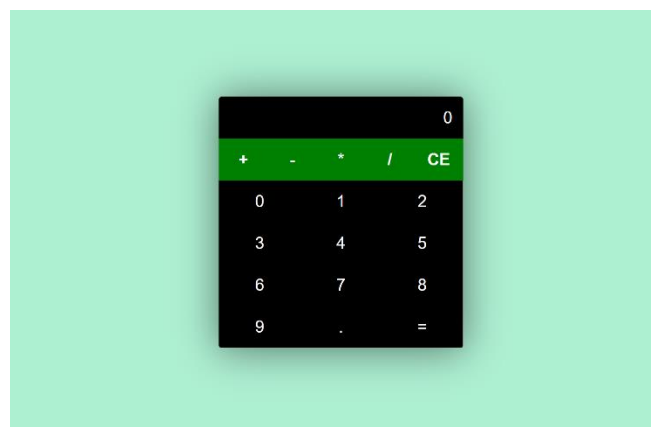


Figure 11.1 : JSX Calculator initial interface



The application of the welcome page and the history section in this deliverable were generated through the use of event listeners. By being able to use this concept with `localStorage`, I was able to successfully store and display information that users have inputted within the history section and the welcome page, allowing more interactivity for users while also being successful in storing information that can be useful for the user. By being able to implement event listeners as an integral part of data storage and output, I was able to reflect how far my learning has come by being able to use concepts I have learned and being able to implement them alongside other libraries I have not been introduced to, showing my evolving improvement as I keep learning front-end development in general.

Deliverables

JSX Calculator

The JSX calculator, a front-end application, is the main deliverable of this project and contains two main sections, the welcome page and the calculator page. The main purpose of this calculator is to calculate mathematical expressions from users and be able to render them through the calculator interface. In addition, the calculator is also responsible for being able to store all the calculations made by the user and be able to clear it if the user wants to start new.

Welcome Page

The Welcome page, as seen in *Figure 12.1* is used to retrieve the name of the user to be taken in and the user is redirected to the main user interface area of the application, the calculator page. The main functionality of this page is checking if a user has inputted their name onto the input box. If the user has not done so, they are asked to write their name down. Once they have written their name down, they are redirected to the second area of the JSX calculator application which is the calculator page as seen in *Figure 12.2*.

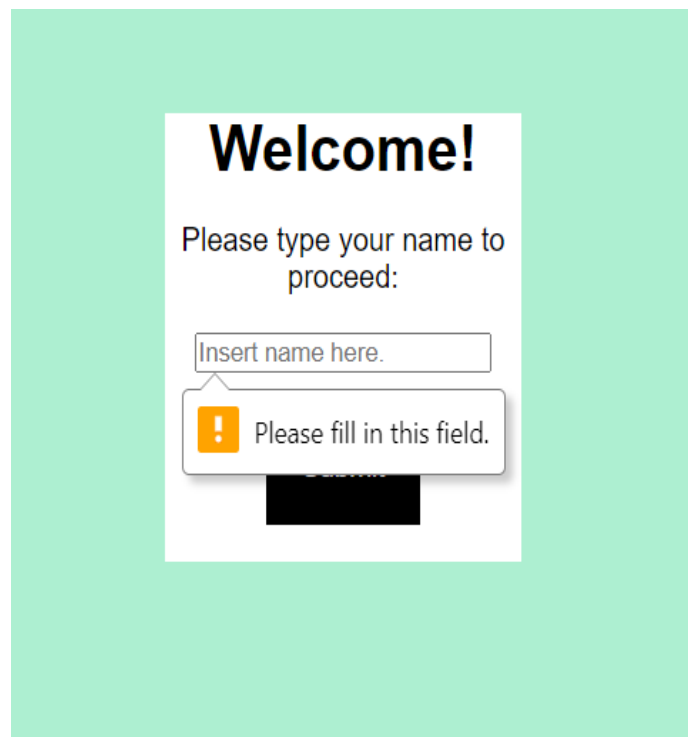


Figure 12.1: JSX Calculator Welcome Page



Calculator Page

The calculator page is the main user interface of the JSX calculator and performs calculations and stores and/or deletes the calculations made by the user. This page contains a heading at the top containing a welcome message to the user, a calculator in the middle of the page with buttons representing each number and arithmetic operations along with a history section on the right hand side of the page with a clear button to remove the history data for the as seen in *Figure 12.2*. The calculator's first and primary operation is to take in the expression the users generate. This is done by taking in the expressions that the user generates through the buttons of the calculator. After the user has calculated the result, the application uses `localStorage` to store the expression that the user has returned along with the output as a bullet point under the history section that is on the right hand side of the application as shown in *Figure 12.3*. Along with storage, one of the other important functionalities that the JSX calculator has on the application is being able to clear the history list of the user upon clicking the clear button.

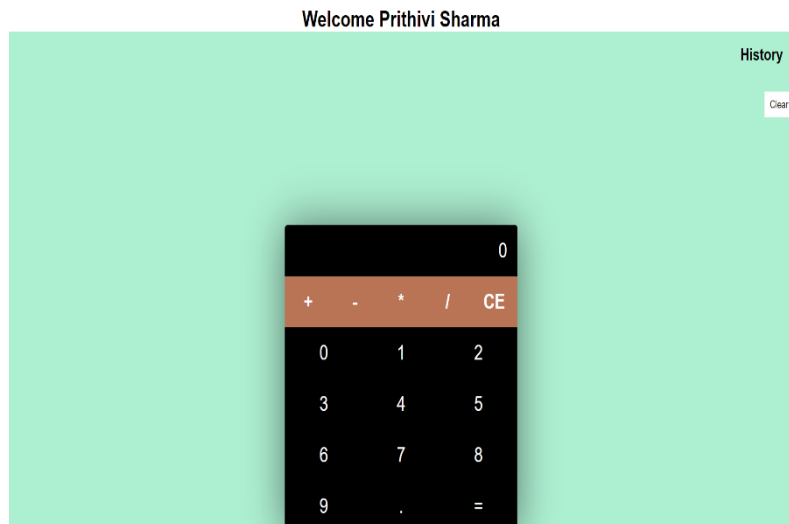


Figure 12.2 : JSX Calculator with no entries by user

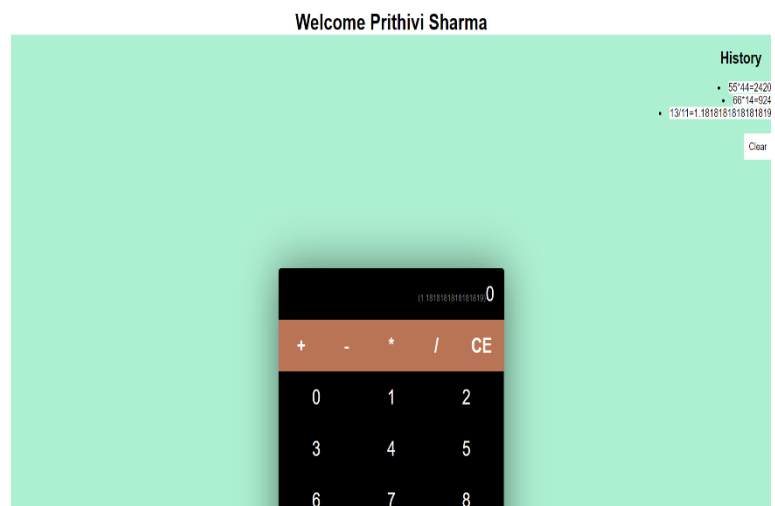


Figure 12.3 : JSX Calculator App with history of user



Reflection

What I learned

From the beginning of this project, I decided to pursue the continuous professional development of software engineering skills that I haven't applied during my tenure at university. As a result, I was exposed to a new area of learning and development to diversify my technical skill-set through front-end development, and I was able to reflect on my knowledge on software development skills that are universal as well helped me strengthen my understanding of JBSE guidelines such as software project management and mobile applications.

On a technical scale, I had learned many new skills that centred around front-end development. One of the main skills I had learned to was to make front-end development projects like websites and applications that have data storage as evident from my deliverable for this CPD project through *Figure 12.3*. This allowed me to learn more about how certain elements of JSX such as event listeners interact and how they are responsible for other actions like data collection and storage. I was also able to learn a lot more about how important technical areas like HTML, CSS and JavaScript and non-technical areas like accessibility are in creating front-end development projects which are inclusive to everyone. In addition, I learned how to integrate all 3 of the base languages involved in front-end development and generate a project with their functionalities combined. From looking at these areas of what I learned, I was able to see I achieved my goal as I was able to get familiarised with fundamental aspects of web programming and expand my professional portfolio while also further refining my knowledge on guidelines involving being a well-rounded software engineer.

Areas to Improve On

After completing this CPD project, I recognised many areas of improvement that were not just limited to my approach but also expanded in my approach of implementation for future front-end development projects. By doing so, I recognised areas which I can revisit later on and be able to refine on project planning and front-end development, further developing my knowledge on both areas in the future.

One significant area where I can improve on in implementation is interactivity as although I was able to find a more interactive way for users to input expressions to be calculated, due to time constraints, I still lacked in other areas such as allowing users to select particular areas from their history section to delete rather than deleting their entire history directly. This in turn, reduced the increased levels of interactivity they could have.



In the context of project planning, I found many skills that if improved, can help me extensively to create projects that are more efficient in execution and on schedule. One of the major areas I recognised to improve was to start small as by being able to learn base concepts of a topic, I would be able to devote more of my time to getting better in one topic and my learning focus be more condensed rather than stretched out. This in turn, would allow me to learn a lot about a particular topic in a short period of time and not worry about overload of work to catch up on. Another part of project planning I believe I could have improved was to plan ahead as I had difficulties being able to complete the first segment of the CPD project. This was primarily because of my approach to complete practical exercises first before refreshing my knowledge on the base theory concepts and their practical application which would have been more advantageous in return. From this experience, I was able to reflect on different approaches I can use in the future before beginning a project which would allow me to have a better learning process when looking at concepts I have not learned before.

Where To from Here

After learning the concepts of front-end development and getting a starting point on web-development, I am planning on moving forward to other concepts that are just as important as front-end development towards web development in general like back-end and full stack development as this would give me an opportunity to learn more about other concepts and also allow me to further expand my skillset and portfolio.