Week 5

Exercise 1:

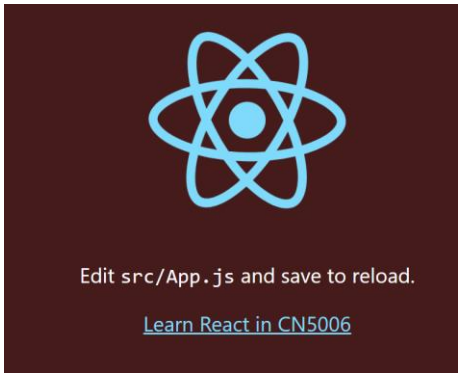
Output:

Explanation:

I modified the anchor HTML element in the App.js file to display a new message.

Exercise 2:

Output:

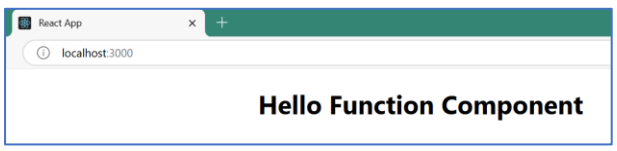


Explanation:

I changed the background colour within the App.css file.

Exercise 3:

Output:

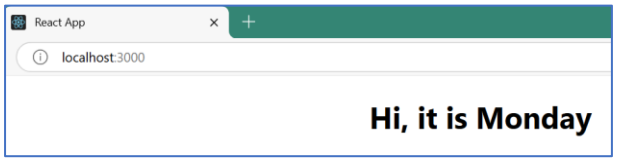


Explanation:

I created my own React component and imported it into index.js file. The component's contents were rendered in the .render() method.

Exercise 4:

Output:

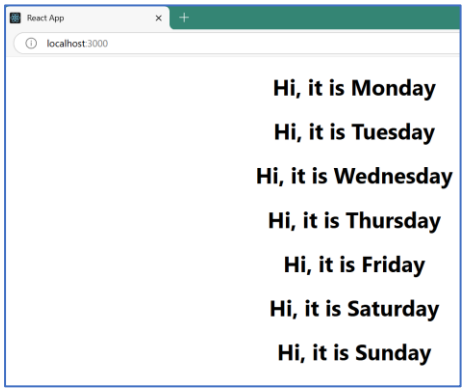


Explanation:

I created another React component, however, this one included a prop. When the component is called in the .render() method, the prop is passed as an argument and a String message is displayed.

Exercise 5:

Output:

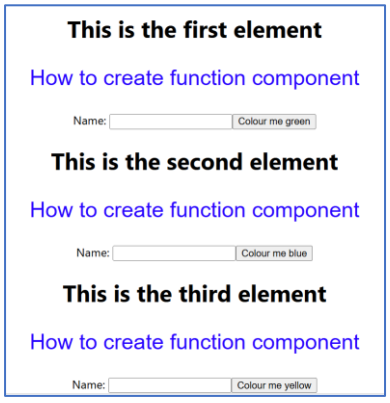


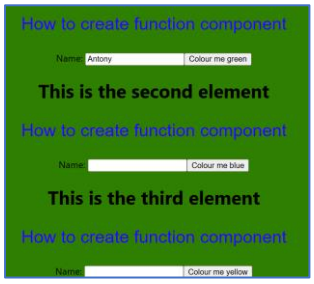
Explanation:

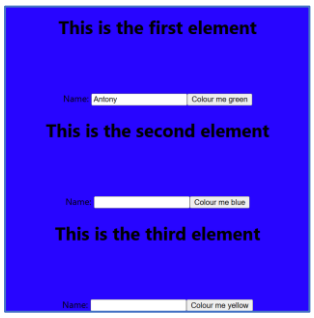
I simply added an additional six lines of code and, as a result, called the functional component six additional times in the .render() method to display each day of the week.

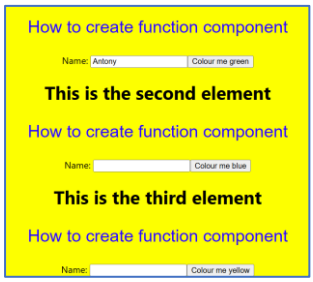
Exercise 6:

Output:









Explanation:

I created a new component which displays HTML text and input elements. The properties heading, name, and color were passed as various arguments within the .render() method to add functionality to the interface. Clicking the button calls a function that both alerts the user with a concatenation of “Mr “ and the user’s name and changes the background colour of the page.

Questions:

1. What is React?

   Ans: React is a JavaScript library used to develop responsive user interfaces for single-page applications. It uses a virtual DOM in memory to enable quick and responsive rendering of elements.

2. What do you understand about React components and what command must you use to create a React component with or without properties?

   Ans: React components are functions that return HTML elements. There are two ways to create a component: class and function.

Class:

```
class ClassComponent extends React.Component
{
  render()
  {
    return null;
  };
};
```

Function:

```
function FunctionComponent()
{
  return null;
};
```

Properties are defined in the parentheses.

3. What command do you use to render the newly created component named MyReact?

   Ans:

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <MyReact/>
  </React.StrictMode>
);
```

4. Suppose the MyReact component has a property heading, write down the code that could be used to render the MyReact component, and pass the message to the property heading as “This is my first element”.

   Ans:

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <MyReact heading="This is my first element"/>
  </React.StrictMode>
);
```

5. Observe the code below and answer the questions:

```
<AppColor heading="This is first element" lbl
="Name :"  color="green"/>
```

   a. What is the name of the React component?

      Ans: AppColor.

   b. How many properties does this component use?

      Ans: Three.

6. Look at the following code:

```
function GreetingElementwithProp(props) {

 return (
  <div className="App">
   <h1>Wellcome , {props.studentname}</h1>;
  </div>
 );
}
 export default ??????
```

What can you write to make the exporting of this function correct?

Ans: GreetingElementWithProp.

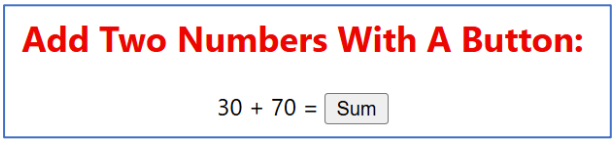
7. Add a function that takes two properties as numbers, adds these numbers on the click event of a button, and then displays the sum.

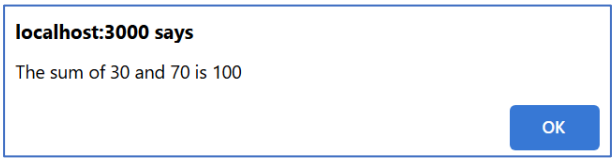
Ans:

```
function SumOfNumbers({num1, num2})
{
    function AddNumbers()
    {
        const sum = Number(num1) + Number(num2);
        alert('The sum of ' + num1 + ' and ' + num2 + ' is ' + sum);
    };

    return (
        <div className='App'>
            <h2 style={{color:'red'}}>Add Two Numbers With A Button:</h2>
            <p>{num1} + {num2} = <button onClick={AddNumbers}>Sum</button></p>
            <br/><br/>
        </div>
    );
};

export default SumOfNumbers;
```





Reflection:

This week I learned the basics of the React library and JSX. React uses a virtual DOM in memory and compares it with the actual DOM to determine which elements must be re-rendered; This is the key to React's responsiveness. I learned how to develop a simple React interface using components, props, JSX, HTML, CSS, and the .render() method. Lastly, I was able to familiarise myself with the structure of React, including the rendered index.html file in the public folder and the various files within the src folder.