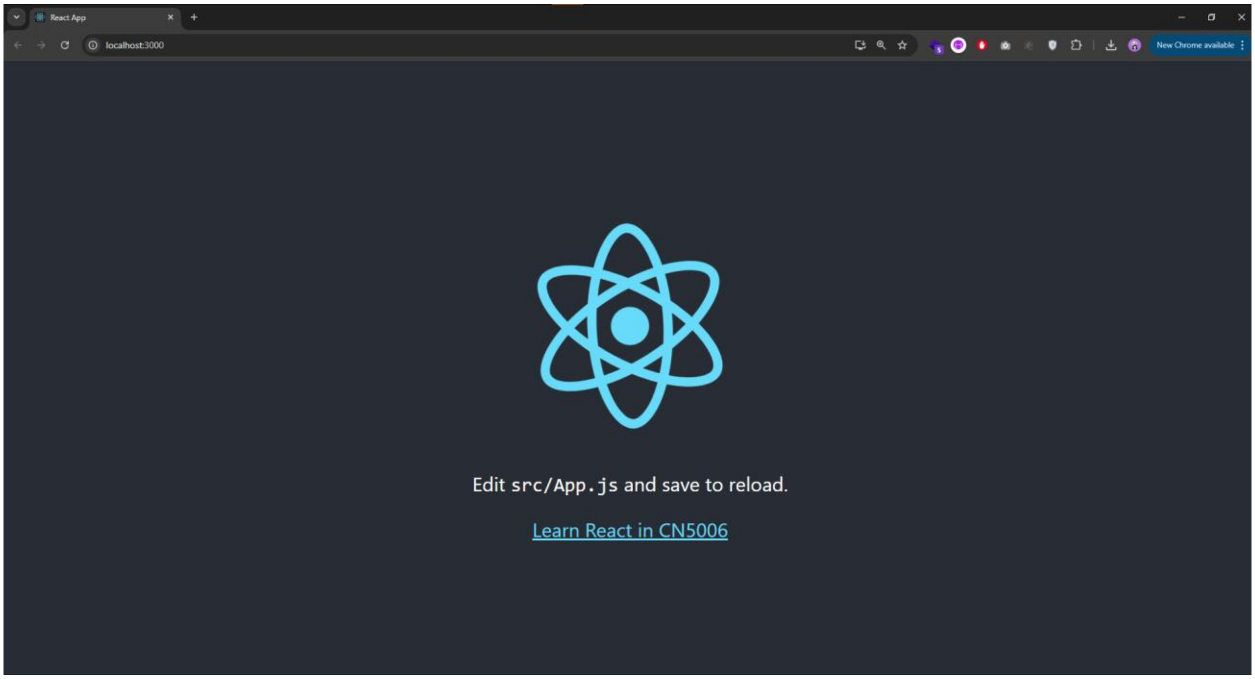
Week 5

Task 1

In this task I have change the text in react which is defined in src\App.js

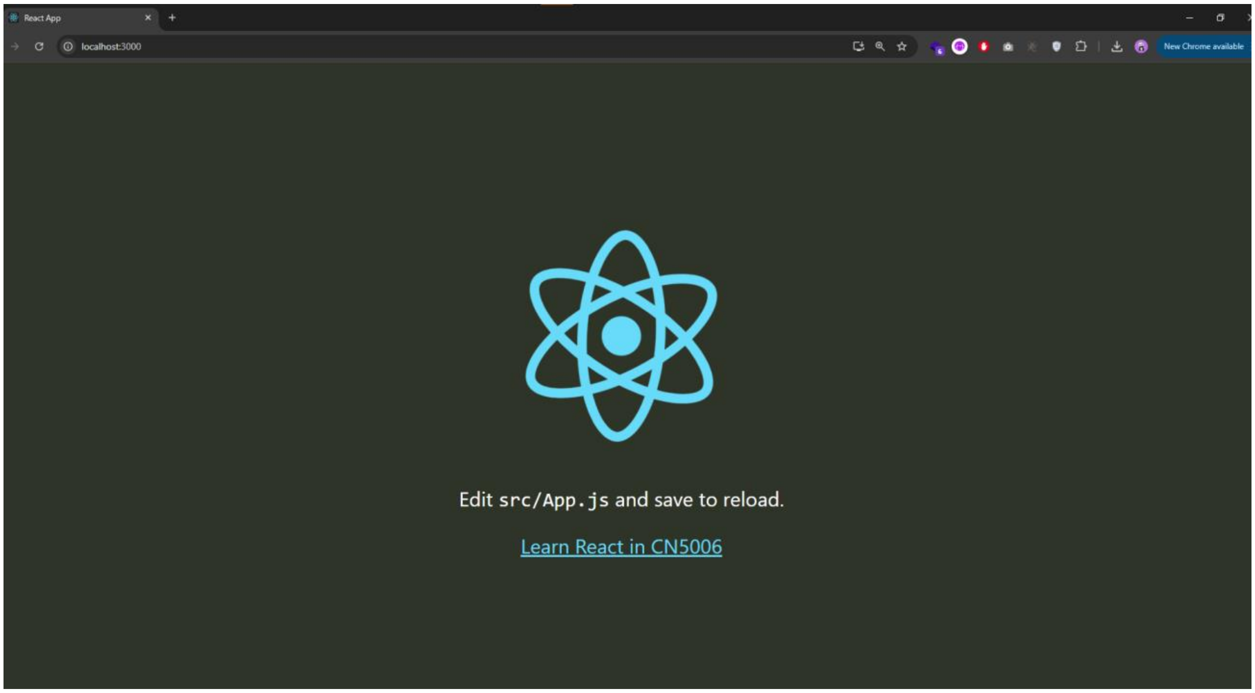


Edit `src/App.js` and save to reload.

Learn React in CN5006

Task 2

In this task I have changed the color of background using css

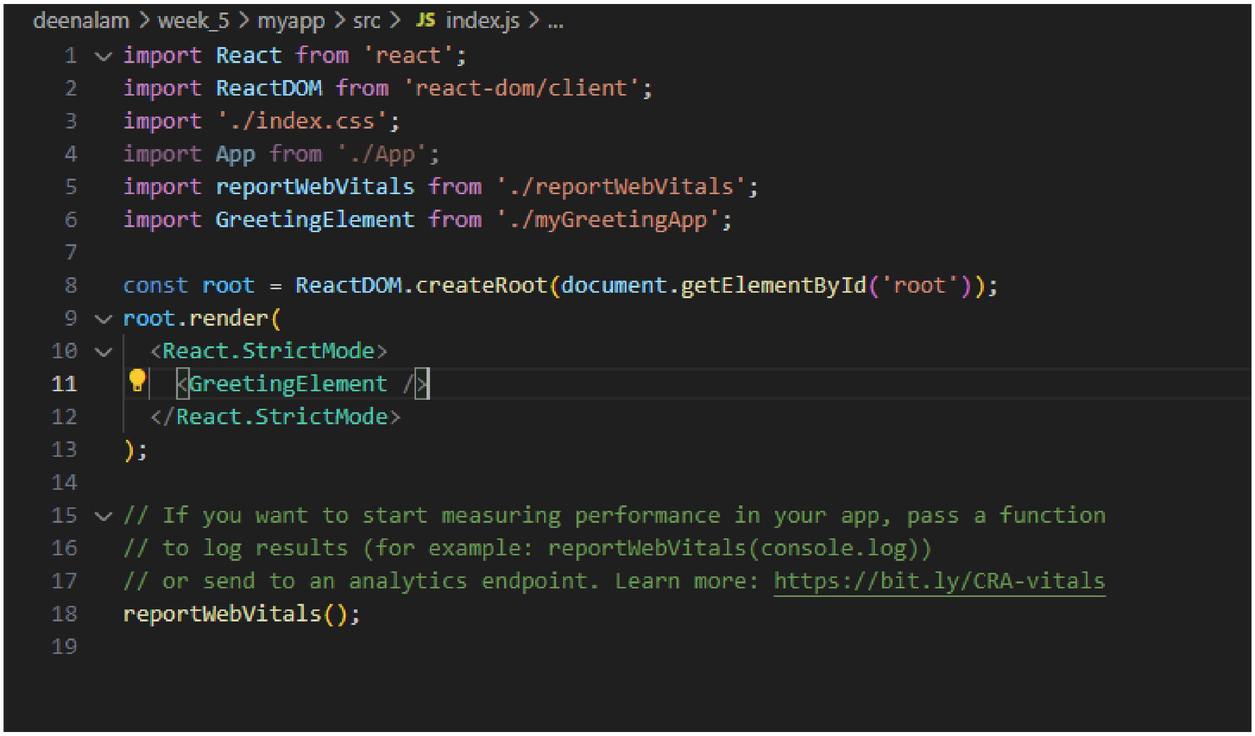


Edit `src/App.js` and save to reload.

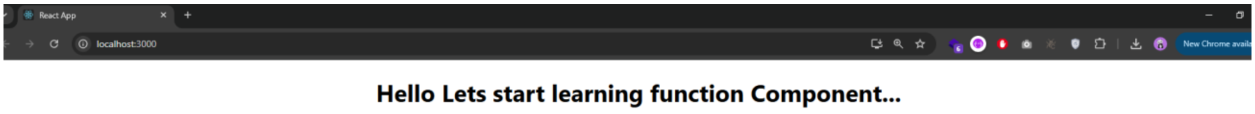
Learn React in CN5006

Task 3

In this task I have created a new component with a message in it and import it in app.js to render. Now instead of render <App/> it renders my custom components on run



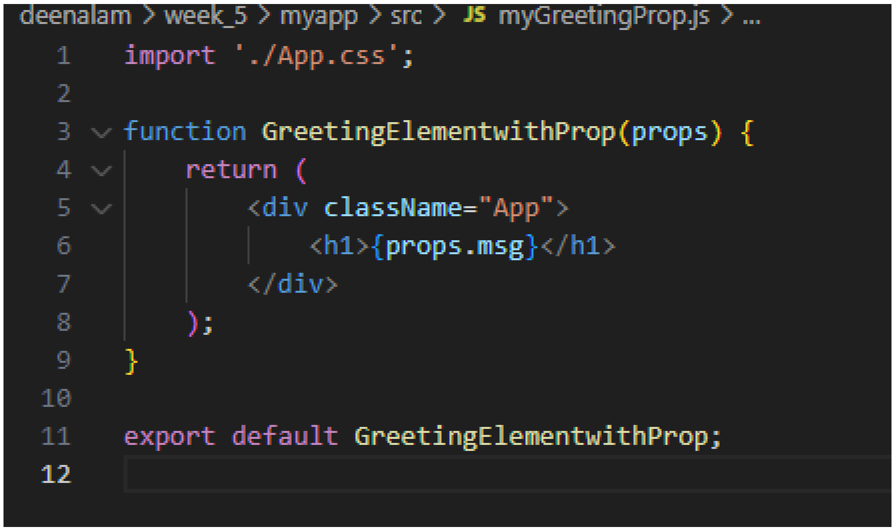
```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import GreetingElement from './myGreetingApp';

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

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```



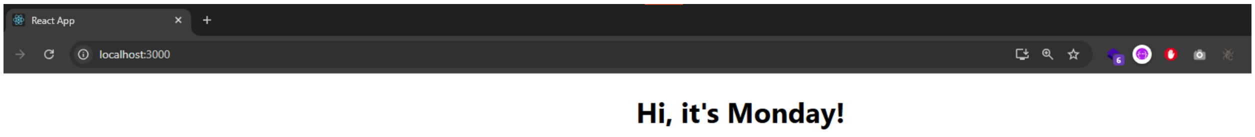
Hello Lets start learning function Component...

Task 4

In this task I have created a functional component that use message as prop to display the message. This component is being used in index.js to render


deenalam > week_5 > myapp > src > JS myGreetingProp.js > ...

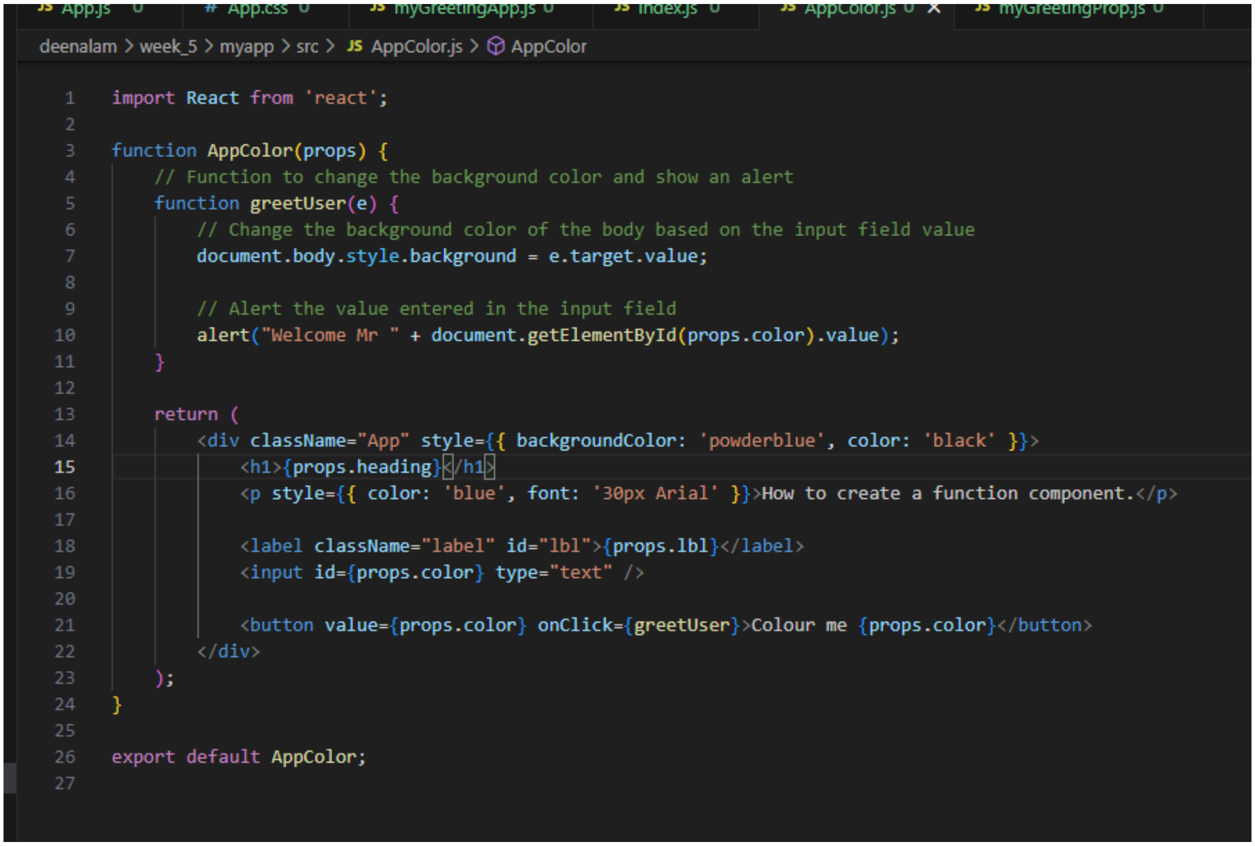
```
import './App.css';

function GreetingElementwithProp(props) {
    return (
        <div className="App">
            <h1>{props.msg}</h1>
        </div>
    );
}

export default GreetingElementwithProp;

```



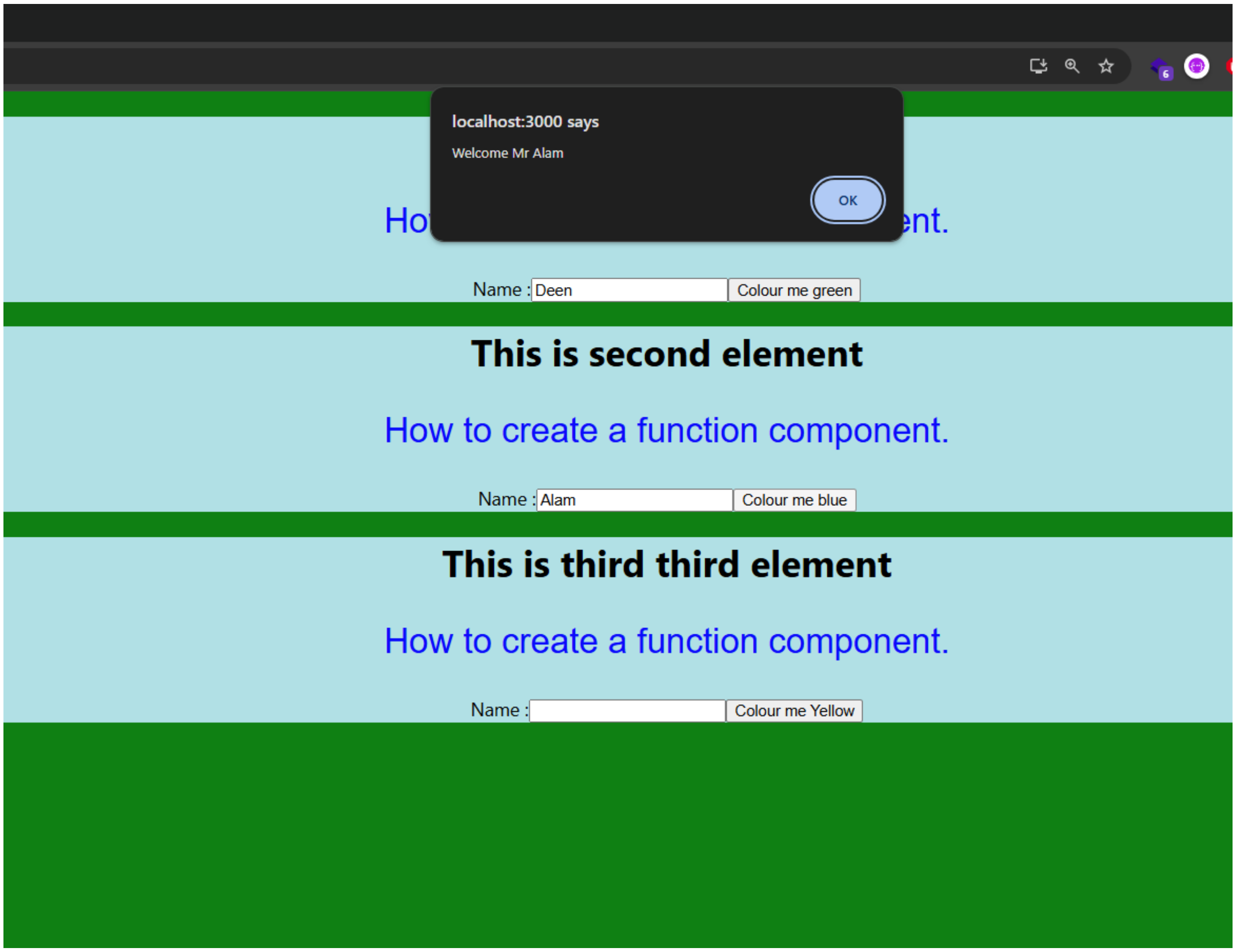
Hi, it's Monday!

Task 5

In this task I have created a component to change the background color random



```
import React from 'react';

function AppColor(props) {
    // Function to change the background color and show an alert
    function greetUser(e) {
        // Change the background color of the body based on the input field value
        document.body.style.background = e.target.value;

        // Alert the value entered in the input field
        alert("Welcome Mr " + document.getElementById(props.color).value);
    }

    return (
        <div className="App" style={{ backgroundColor: 'powderblue', color: 'black' }}>
            <h1>{props.heading}</h1>
            <p style={{ color: 'blue', font: '30px Arial' }}>How to create a function component.</p>

            <label className="label" id="lbl">{props.lbl}</label>
            <input id={props.color} type="text" />

            <button value={props.color} onClick={greetUser}>Colour me {props.color}</button>
        </div>
    );
}

export default AppColor;
```



localhost:3000 says

Welcome Mr Alam

Ho ent.

Name : Deen

This is second element

How to create a function component.

Name : Alam

This is third third element

How to create a function component.

Name :

1. What is React?

React is an open-source JavaScript library used for building user interfaces, primarily for single-page applications (SPAs). It was developed by Facebook and allows developers to create reusable UI components, manage state efficiently, and render dynamic content. React uses a virtual DOM (Document Object Model) to optimize performance by minimizing the number of actual DOM manipulations. This results in faster updates and better user experience.

2. What do you understand by React component and what command do you use to create a React component with or without a property?

A React component is a fundamental building block in React applications. It can either be a function or a class that returns JSX (JavaScript XML) to represent the UI elements. Components can accept input called "props" (properties) which allow them to be customized.

Without properties: You can create a simple functional component like this:

jsx

Copy code

```
function MyComponent() {
  return <div>Hello, World!</div>;
}
```

With properties: You can create a functional component that accepts props:

jsx

Copy code

```
function MyComponent(props) {
  return <div>{props.message}</div>;
}
```

To create a React component, the following command is typically used if you're using Create React App:

bash

Copy code

```
npx create-react-app my-app
```

You can then create the component manually in a new file (e.g., MyComponent.js).

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

To render the MyReact component in a React app, you will use the following code in your main App.js or a separate render file:

jsx

Copy code

```
import React from 'react';
import ReactDOM from 'react-dom';
import MyReact from './MyReact';  // Assuming the component is in MyReact.js

ReactDOM.render(
  <MyReact />,
  document.getElementById('root')
);
```

Alternatively, if using JSX within App.js, the component would be rendered as follows:

jsx

Copy code

```
import MyReact from './MyReact';

function App() {
  return (
    <div>
      <MyReact />
    </div>
  );
}
```

```
export default App;
```

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”.

To pass the heading property to the MyReact component, you would render it as shown below:

jsx

Copy code

```
import React from 'react';


function MyReact(props) {
  return <h1>{props.heading}</h1>;
}


export default MyReact;


// In the main App.js file:
import MyReact from './MyReact';


function App() {
  return (
    <div>
      <MyReact heading="This is my first element" />
    </div>
  );
}
```

```
export default App;
```

5. Observe this code and answer the questions below:

jsx

Copy code

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

a. What is the name of the React Component?

The name of the React component is AppColor.

b. How many properties does this component use?

The component uses three properties: heading, lbl, and color.

6. Look at the following Code:

jsx

Copy code

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

To export the function correctly, replace ?????? with GreetingElementwithProp:

jsx

Copy code

```
export default GreetingElementwithProp;
```

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

Here's the code for the component that adds two numbers and displays the sum when a button is clicked:

jsx

Copy code

```
import React, { useState } from 'react';

function AddNumbers(props) {
  const [sum, setSum] = useState(0);

  const handleAdd = () => {
    const total = props.num1 + props.num2;
    setSum(total);
  };

  return (
    <div>
      <button onClick={handleAdd}>Add Numbers</button>
      <h2>Sum: {sum}</h2>
    </div>
  );
}

export default AddNumbers;

// In your main component (App.js)
```

```
import AddNumbers from './AddNumbers';

function App() {
  return (
    <div>
      <AddNumbers num1={5} num2={10} />
    </div>
  );
}

export default App;
```