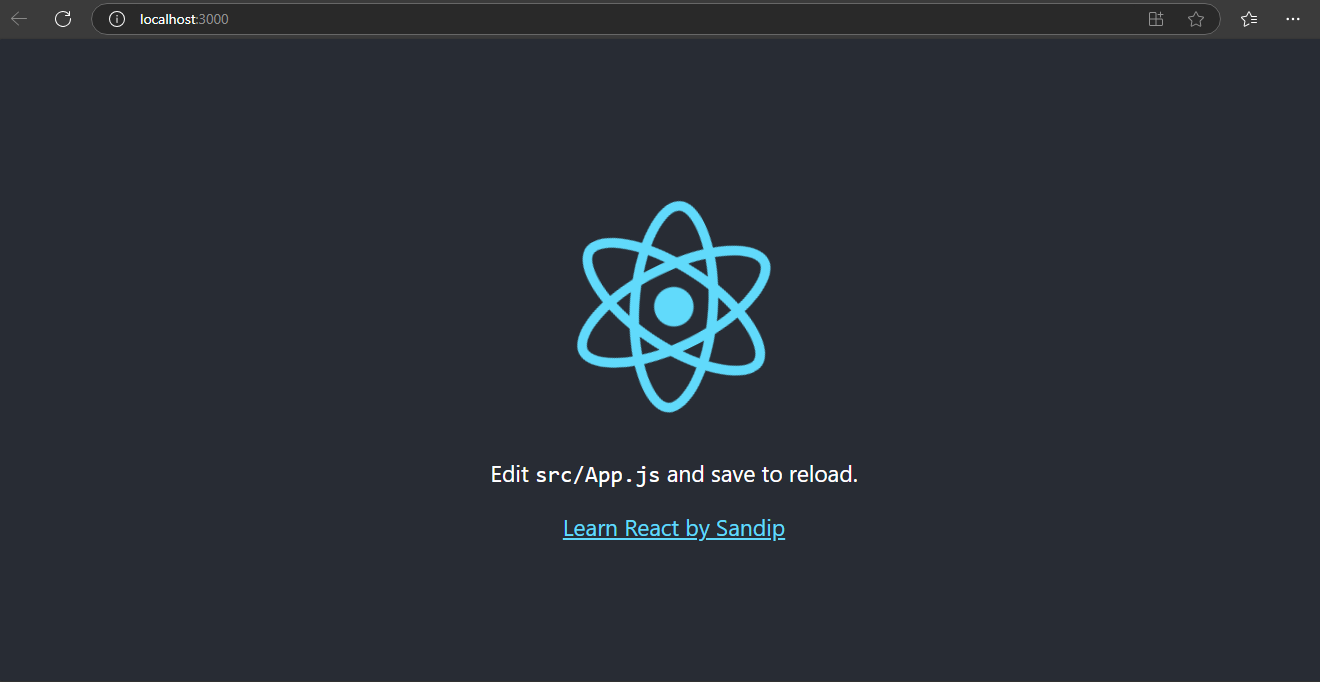
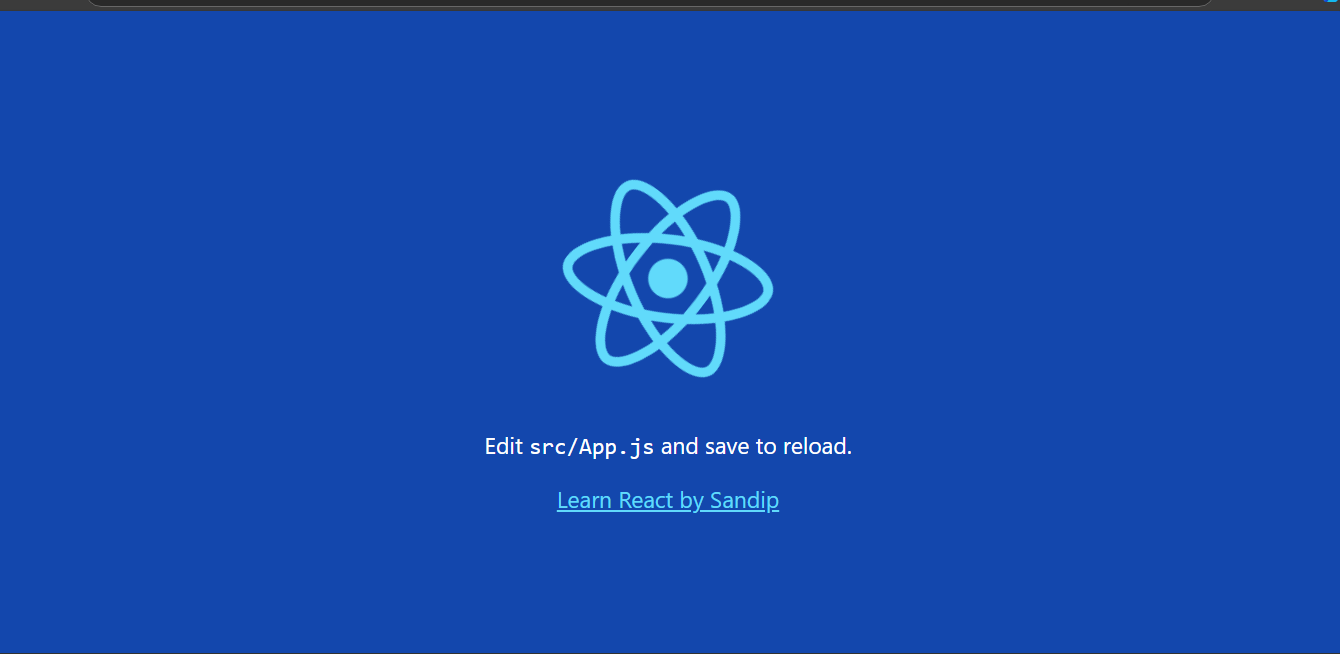
Reflection(week 5):-

In this week 5 lab, the lab tutor introduces us to basic react programming and how to create functional components to create user interfaces. We installed the react application by typing: npx create-react-app (folder name) command in the vs code terminal then something was running in my browser at localhost:3000 as all output and answer questions are given below for all tasks given in this week's lab tutorial. Also, all the code I have performed in this lab tutorial is attached with GitHub link.

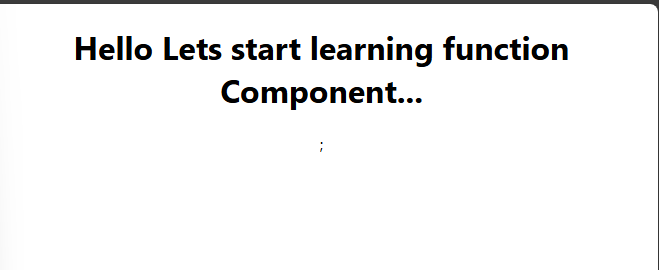
Task 1: Change the text appearing into your React Application:



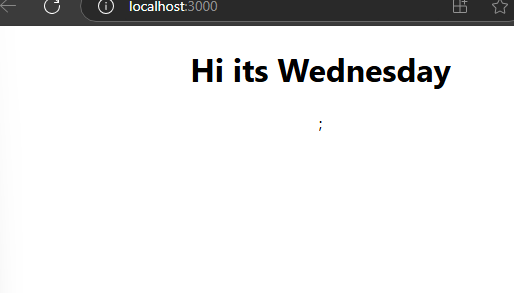
Task2: Change the background color of your application:

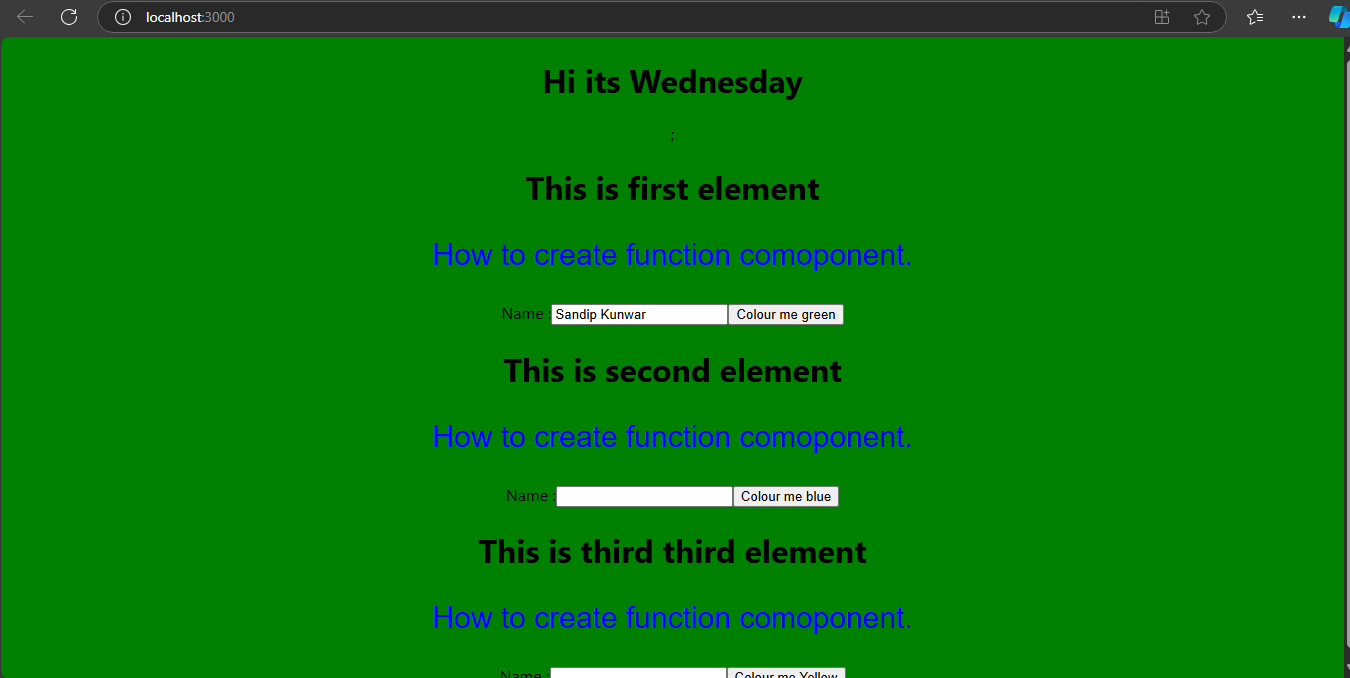


Task3 : Creating a stateless function component using React:



Task4 : Creating the Functional Component using properties:





For todays Lab submission, After you complete the lab write down a word document answering following questions for your portfolio:

1. What is React?

Ans:- React is a framework that employs Webpack to automatically compile React, JSX, and ES6 code while handling CSS file prefixes. React is a javascript-based UI development library. Although React is a library rather than a language, it is widely used in web development.

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

Ans:- A **React component** is a JavaScript function or class that optionally accepts inputs (called "props") and returns a React element that describes how the UI should appear. Components can be either **functional** or **class-based**.

Without property:

function MyComponent() {

return <h1>Hello, World!</h1>;

}

With property:

function MyComponent(props) {

return <h1>Hello, {props.name}!</h1>;

}

To create a component using the **command line**, you can use a command like:

npx create-react-app my-app

cd my-app

npm start

If you’re adding a component to the project:

# Create a new file inside the src folder, for example:

touch src/MyComponent.js

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

Ans:- To render the MyReact component, you typically add it to the App.js file (or another root component file). You use the following command in JSX syntax to render MyReact:

import MyReact from './MyReact';

function App() {

return (

<div className="App">

<MyReact />

</div>

);

}

export default App;

Make sure that MyReact is defined and exported properly, and you import it in the file where you want to render it.

1. 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”

function MyReact(props) {

return <h1>{props.heading}</h1>;

}

// In the parent component (e.g., App.js)

function App() {

return (

<div className="App">

<MyReact heading="This is my first element" />

</div>

);

}

export default App;

5. Observe this code and answer the questions below:



1. What is the name of the React Component

Ans:- AppColor

1. How many properties this component uses

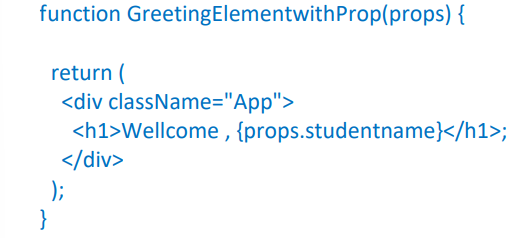
Ans:- 3

1. heading

2. lbl

3. color

7. Look at the following Code:



export default ??????

what will you write to make this export this function correctly? Hint you need to replace ?????? with the correct word.

Ans:- export default GreetingElementwithProp;

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

Hint you will be using in jsx

import React, { useState } from 'react';

function AddNumbers(props) {

const [sum, setSum] = useState(0);

// Function to add the two numbers

const handleClick = () => {

setSum(props.num1 + props.num2);

};

return (

<div>

<button onClick={handleClick}>Add Numbers</button>

<p>Sum: {sum}</p>

</div>

);

}

export default AddNumbers;

**Explanation:**

* The AddNumbers component takes two props: num1 and num2.
* The handleClick function adds these two numbers when the button is clicked.
* The sum is displayed inside a <p> tag.

To use this component and pass the numbers, you would render it like this:

<AddNumbers num1={5} num2={3} />

When you click the "Add Numbers" button, it will display Sum: 8 on the screen.