**Portfolio (Week-5)**

This week’s lab session focused on creating a **React application**. React, an open-source JavaScript library, simplifies the process of building user interfaces and is especially useful for creating single-page applications. Below is a detailed explanation of the tasks I performed and the lessons I learned:

1. **Checking Node.js Installation**

Before starting, I verified the installation of Node.js on my computer by running the following command in the terminal:

This command confirmed the installed version of Node.js, ensuring the environment was set up correctly.

A screenshot of a computer

Description automatically generated

 **Setting Up the React Application**

I created a directory named **React Work** in my Week 5 folder and initialized a new React application using the npx command:

After the application was created, I navigated into the myapp directory using the cd command:

Finally, I started the React development server with the command:

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Output**

The default React application loaded successfully in the browser, displaying a sample interface with the React logo.

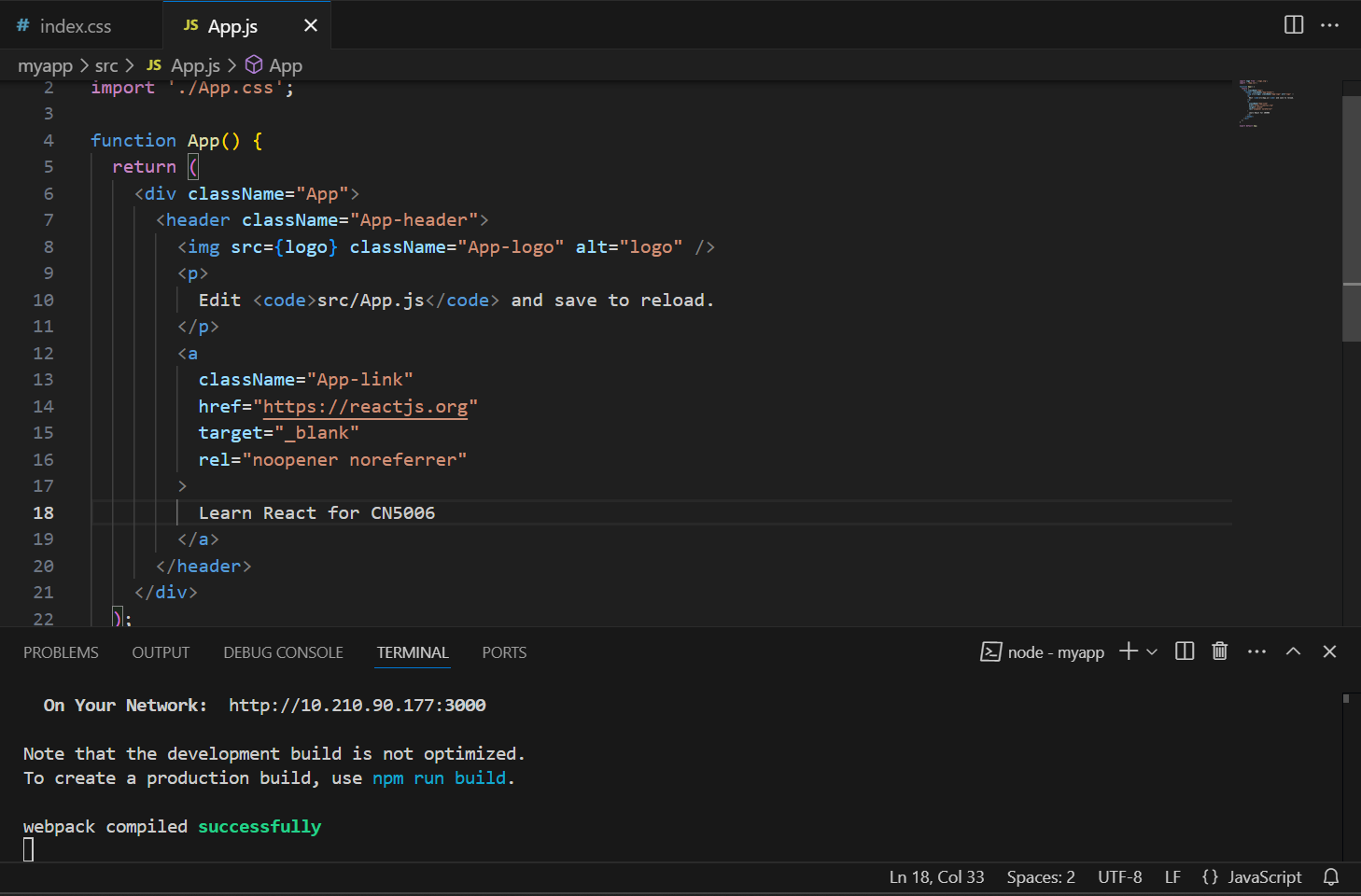
A screen shot of a computer

Description automatically generated

**Task 1: Modifying the Application Text**

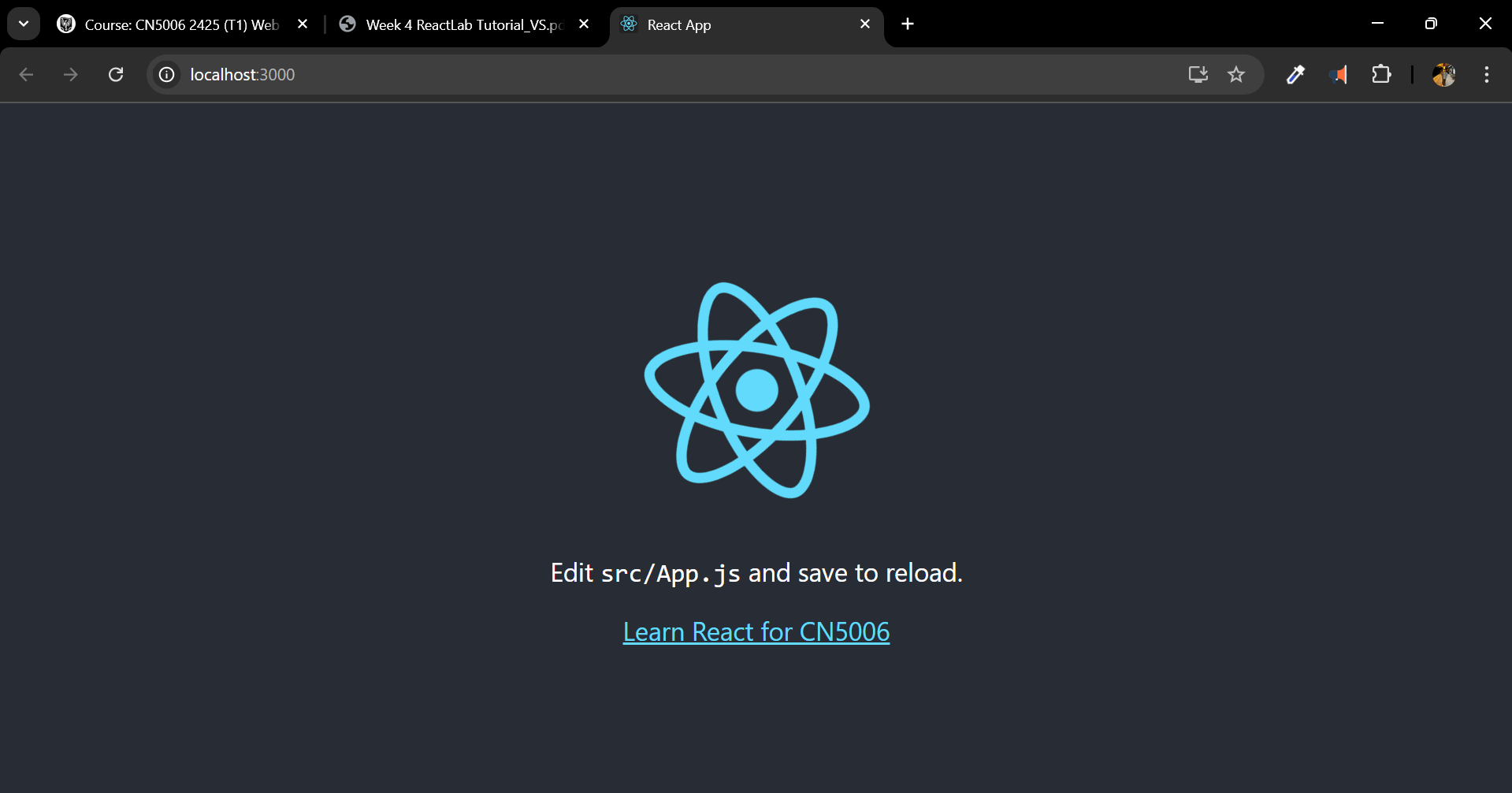
I made changes to the default React application by updating the text displayed on the webpage. Here’s what I did:

* Edited the App.js file to modify the JSX code.
* Changed the heading text to something personalized, such as "Welcome to My React App!".



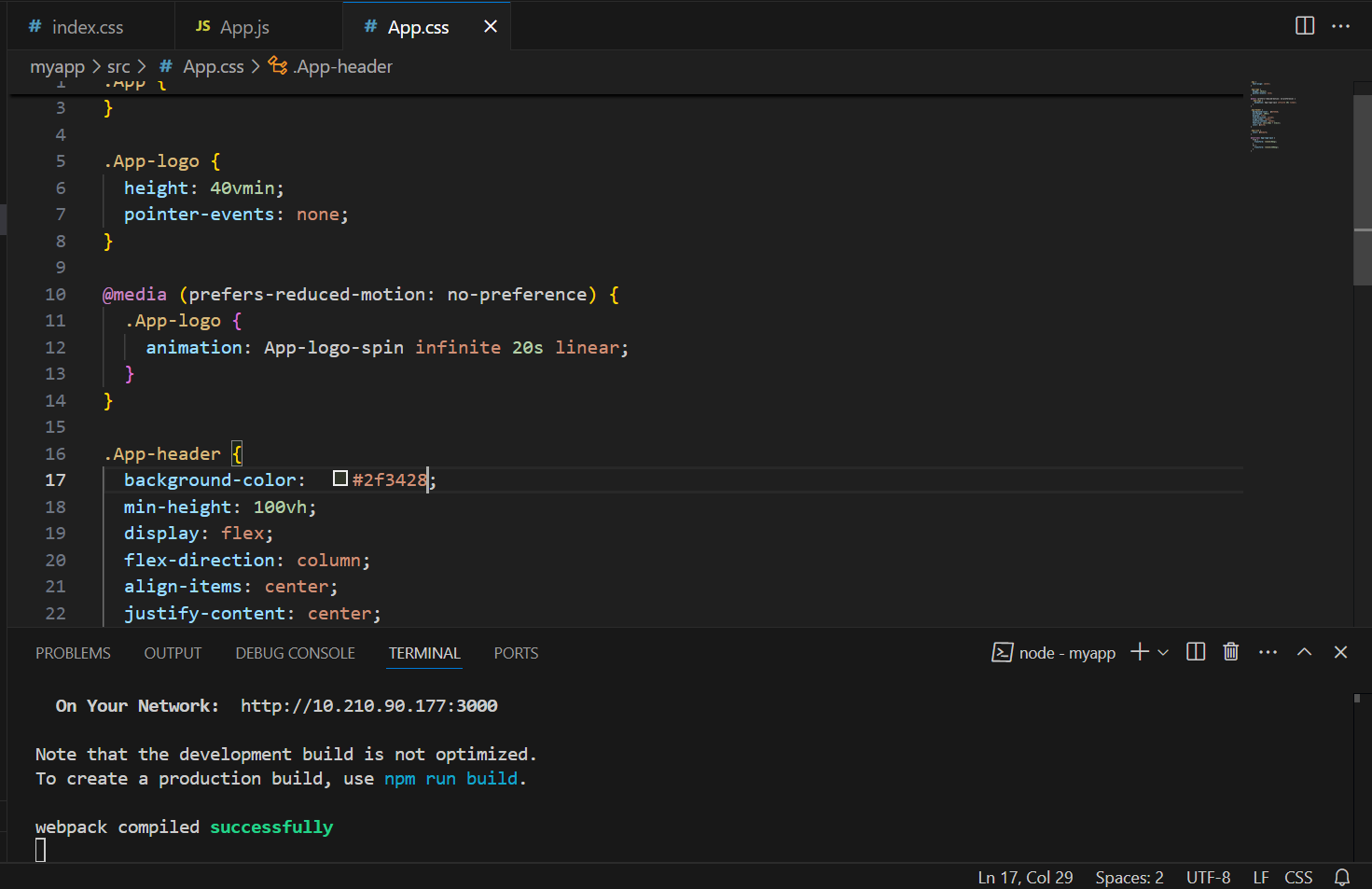
**Output**

The updated text appeared on the webpage as expected.



**Task 2: Changing Background Color**

To customize the application further, I changed the background color by editing the App.css file. I replaced the existing color code with a new one to achieve the desired appearance.



**Output**

The background color of the application changed as specified.

A screen shot of a computer

Description automatically generated

**Task 3: Creating a New React Component**

For this task, I created a new React component called MyGreetingApp. Here’s how I approached it:

1. Created a file named MyGreetingApp.js in the src folder.
2. Wrote a functional component that displays a personalized greeting message.
3. Imported the component into the index.js file.
4. Replaced the <App /> tag in index.js with <MyGreetingApp />.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**SS**

**Output**

The browser displayed the greeting message from the MyGreetingApp component.

A screenshot of a computer

Description automatically generated

**Task 4: Working with Props**

In this task, I created another component named MyGreetingProp.js, which demonstrates the use of props. Here are the steps:

1. Created the MyGreetingProp.js file in the src folder.
2. Defined a functional component that accepts a name prop and displays a greeting message.
3. Imported and used the component in index.js, passing a name value as a prop.

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Output**

The browser displayed a personalized greeting message based on the passed prop.

A screenshot of a computer

Description automatically generated

**Task 5: Creating Weekly Greetings**

To create greetings for each day of the week, I followed these steps:

1. Copied the greeting line for Monday and updated it for other days of the week.
2. Ensured that all days were displayed dynamically in the application.

A screen shot of a computer program

Description automatically generated **Output**

The browser displayed greetings for all seven days of the week.

A screenshot of a computer

Description automatically generated

**Task 6: Changing Background Color Dynamically**

For this task, I created a file named AppBackgroundColour.js. Here’s what I did:

1. Defined a component that dynamically changes the background color based on user interaction or predefined logic.
2. Imported this component into index.js and incorporated it into the application.

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Now let’s jump on to the question and answer part:

1. What is React

Ans: React is open-source JavaScript library that is created by Facebook. It helps in building user interfaces, and it is especially used for single-page applications.

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

Ans: React component is reusable data that defines a part of user interface. When creating React component for with property we use “**function** MyComponent() { **return** Hello, World!; }” and for without property we use “**function** MyComponent(props) { **return** <h1>Hello, {props.name}</h1>; }”.

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

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

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: ReactDOM.render( <MyReact heading="this is my first element" />, document.getElementById('root') );

5. Observe this code and answer the questions below

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

a. What is the name of the React Component

b. How many properties this component uses

Ans:

a) AppColor

b) Three: heading, lbl, and color

7. Look at the following Code: function GreetingElementwithProp(props) { return (

**Wellcome , {props.studentname}**

;

); } export default ??????

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

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

Ans: To export default: **export** **default** GreetingElementwithProp;

To a function that takes two properties as numbers:

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

**function** handleClick() { setSum(props.num1 + props.num2); }

**return** ( <div> <button onClick={handleClick}>Add Numbers</button> <p>Sum: {sum}</p> </div> ); }

To add these numbers on the click event of the button and display the sum:

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

**Learning Objectives**

* To understand the fundamentals of React and its component-based architecture.
* To practice creating and styling React components.
* To learn how to use props to make components dynamic and reusable.
* To explore the basics of state management in React.

**Key Points**

* React simplifies UI development through reusable components.
* Props allow components to accept dynamic data.
* CSS can be integrated with React to customize the application’s appearance.
* State management enables components to handle dynamic behavior and data updates.

**Problems Faced**

1. **Issue Starting React App:** I encountered an error when running npm start due to a missing dependency.
   * **Solution:** I reinstalled the dependencies using npm install and resolved the issue.
2. **CSS Styling Confusion:** I initially struggled to apply CSS styles to components.
   * **Solution:** I referred to the React documentation and learned how to import and use CSS files correctly.
3. **Prop Usage Errors:** While working with props, I faced errors due to missing or incorrectly defined properties.
   * **Solution:** I ensured that all required props were passed and that the component definitions matched the expected inputs.

By addressing these challenges, I developed a deeper understanding of React and its powerful features for building interactive user interfaces.