

Lesson 02 Demo 01

Create a React Application Using React Components

Objective: To create a React application with a class component to display a desired text

Tools Required: Node terminal, React app, and Visual Studio Code

Prerequisites: Knowledge of creating a React app and understanding of the folder structure

Steps to be followed:

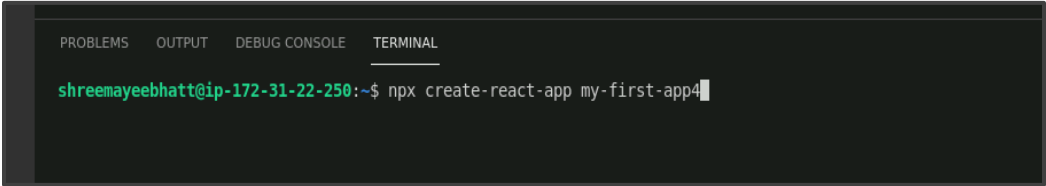
1. Create a new **React** app
2. Create a class component
3. Render the component
4. Run the app

Step 1: Create a new React app

1.1 Open your terminal

1.2 Run the command given below to create a new React app:

`npx create-react-app my-first-app4`



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
shreemayeebhatt@ip-172-31-22-250:~$ npx create-react-app my-first-app4
```

- 1.3 Move to the newly created directory by running the following command: **cd my-first-app4** and open your project in Visual Studio Code

```
shreemayeebhatt@ip-172-31-22-250:~$ cd my-first-app4/
```

Step 2: Create a Class component

- 2.1 In your code editor, open the **src/App.js** file
- 2.2 Replace the existing code in **App.js** with the following code:

```
import React from 'react';
class FirstApp extends React.Component {
  render() {
    return <h2>This is my first Assignment</h2>;
  }
}

export default FirstApp;
```

- 2.3 Define a class component named **FirstApp** with the help of the code, which extends the **React.Component** class. It has a **render** method that returns a JSX element **<h2>This is my first Assignment</h2>**

```
JS App.js  X
src > JS App.js > ...
1  import React from 'react';
2
3  class FirstApp extends React.Component {
4    render() {
5      return <h2>This is my first Assignment</h2>;
6    }
7  }
8
9  export default FirstApp;
10
```

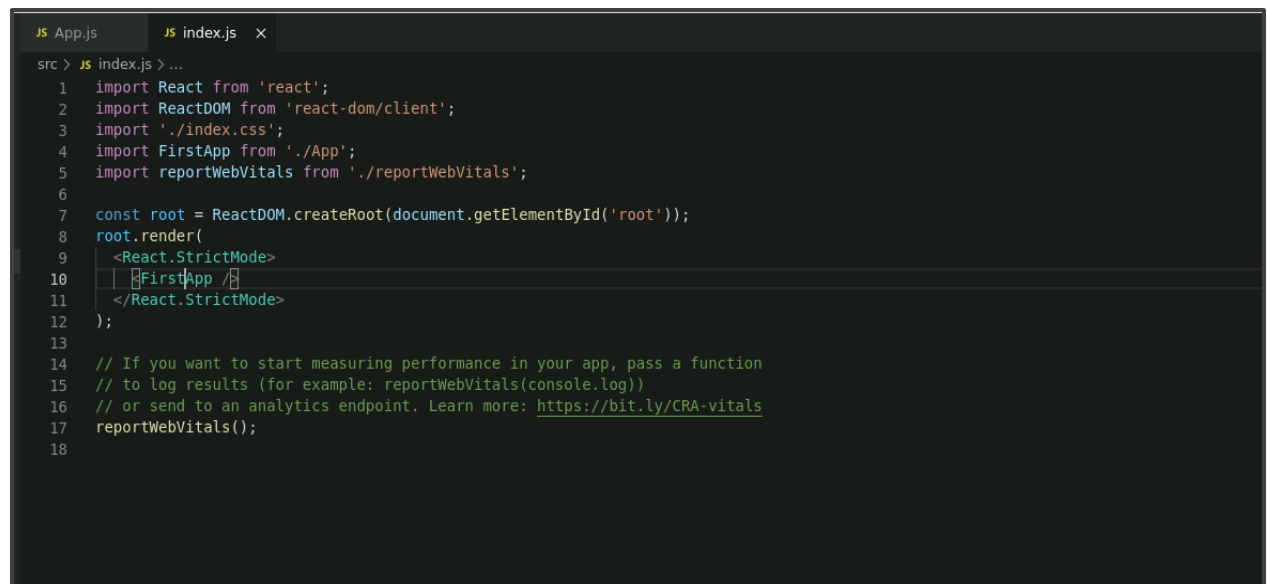
Step 3: Render the component

3.1 In `src/index.js`, import the **FirstApp** component:

```
import FirstApp from './App';
```

3.2 Inside the **ReactDOM.render** method, replace the second argument with **<FirstApp/>**:

```
ReactDOM.render(  
  <React.StrictMode>  
    <FirstApp />  
  </React.StrictMode>,  
  document.getElementById('root')  
);
```



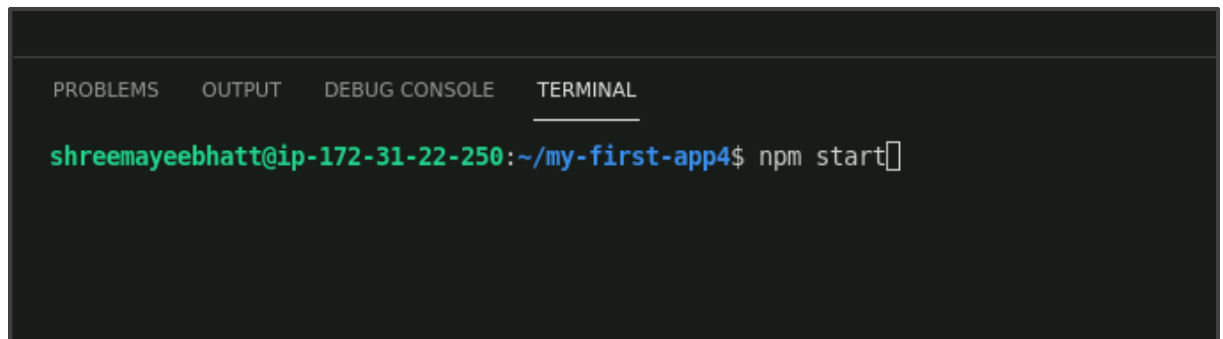
```
JS App.js JS index.js X  
src > JS index.js > ...  
1  import React from 'react';  
2  import ReactDOM from 'react-dom/client';  
3  import './index.css';  
4  import FirstApp from './App';  
5  import reportWebVitals from './reportWebVitals';  
6  
7  const root = ReactDOM.createRoot(document.getElementById('root'));  
8  root.render(  
9    <React.StrictMode>  
10     <FirstApp />  
11   </React.StrictMode>  
12 );  
13  
14 // If you want to start measuring performance in your app, pass a function  
15 // to log results (for example: reportWebVitals(console.log))  
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  
17 reportWebVitals();  
18
```

Step 4: Run the app

4.1 In the terminal, navigate to the project directory (**my-first-app4**)

4.2 Execute the following command to start the development server and run the app:

npm start

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'TERMINAL' tab is active. The prompt shows the user 'shreemayeebhatt' at IP 'ip-172-31-22-250' in the directory '~/my-first-app4'. The command 'npm start' has been entered, followed by a cursor.

4.3 Once the server starts successfully, open **http://localhost:3000** in your browser to view the app

