

Lesson 02 Demo 02

Creating a React Application Using React Props and State

Objective: To implement a React web application to store any details about a bike

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

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

Steps to be followed:

1. Create a new React app
2. Implement the Bike component
3. Render the Bike component
4. Run the application

Step 1: Create a new React app

- 1.1 Create a new React app using the create-react-app command as shown below:

npx create-react-app my-app

```
shreemayeebhatt@ip-172-31-22-250:~$ npx create-react-app my-app
```

- 1.2 Move to the newly created directory by running the following command:

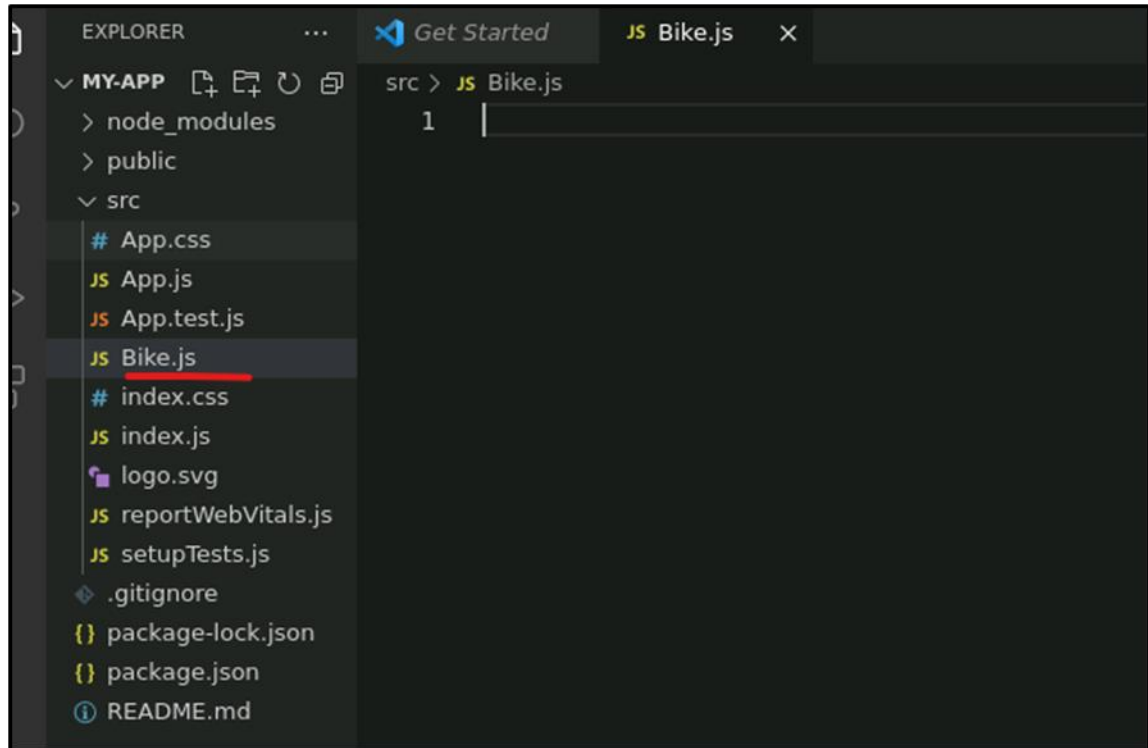
cd my-app

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

Note: Open your project in Visual Studio Code

Step 2: Implement the Bike component

2.1 In the **src** directory, create a new file named **Bike.js**



2.2 Import React and define a class component called **Bike** using the following code:
class Bike extends React.Component

```
import React from "react";  
  
class Bike extends React.Component {
```

- 2.3 Implement the constructor method in the **Bike** class using the **constructor(props)**, and inside the constructor, call **super(props)** to invoke the parent class constructor

```
constructor(props) {  
  super(props);  
}
```

- 2.4 Define the initial state of the component with the **make**, **model**, and **color** properties as shown below:

```
this.state = {  
  make: "Yamaha",  
  model: "R15",  
  color: "blue"  
};
```

```
this.state = {  
  make: "Yamaha",  
  model: "R15",  
  color: "blue"  
};  
}
```

- 2.5 Implement the **changeBikeColor** method, which uses **setState()** to update the **color** property to **black**:

```
changeBikeColor = () => {  
  this.setState({ color: "black" });  
}
```

```
changeBikeColor = () => {  
  this.setState({color: "black"});  
}
```

2.6 Implement the **render** method, which returns a **JSX** element representing the **Bike** component

2.7 In the **JSX**, access the state properties using **this.state.propertyName**

2.8 Display the make using `<h2>My {this.state.make}</h2>`

2.9 Display the color and model using the below code:

```
<p>It is a {this.state.color}  
{this.state.model}.</p>
```

2.10 Add a button with an **onClick** event handler that triggers the **changeBikeColor** method:
`<button type="button" onClick={this.changeBikeColor}>Change color</button>`

```
render() {  
  return (  
    <div>  
      <h2>My {this.state.make}</h2>  
      <p>  
        It is a {this.state.color}  
        {this.state.model}.  
      </p>  
      <button  
        type="button"  
        onClick={this.changeBikeColor}  
      >Change color</button>  
    </div>  
  );  
}
```

```
export default Bike;
```

Step 3: Render the Bike component

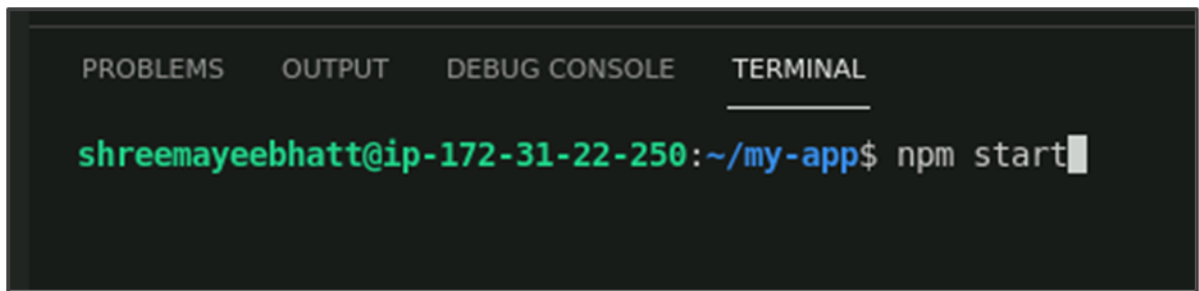
- 3.1 Open the **index.js** file in the **src** directory
- 3.2 Import **React** from React in **index.js** file
- 3.3 Import the **Bike** component from **./Bike**
- 3.4 Use the **ReactDOM.render()** method to render the **Bike** component:
ReactDOM.render(<Bike />, document.getElementById('root'));

```
src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import Bike from './Bike';
5
6
7  const root = ReactDOM.createRoot(document.getElementById('root'));
8  root.render(
9    <React.StrictMode>
10     <Bike />
11   </React.StrictMode>
12 );
```

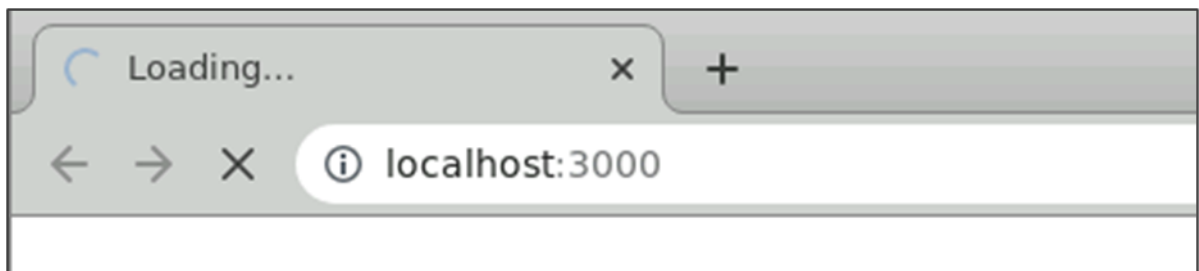
Step 4: Run the application

- 4.1 In the terminal, navigate to the project's root directory

4.2 Run the command **npm start** to start the development server

A screenshot of a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing a green prompt 'shreemayeebhatt@ip-172-31-22-250:~/my-app\$' followed by the command 'npm start' and a cursor.

4.3 Open your browser and navigate to <http://localhost:3000>



Note: You should see the app with the initial bike information displayed along with a button to change the bike color

4.4 Click the **Change color** button to update the **bike color** to **black**



By following these steps, you have successfully created a React web application using React Props and State to manage and display bike details.