MODULE:9 ReactJs Intro

- **What is React Js?**

Answer: React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta and a community of individual developers and companies. React can be used to develop single-page, mobile, or server-rendered applications with frameworks.

- **What is NPM in React Js?**

Answer: NPM stands for Node Package Manager. It is a package manager for JavaScript. NPM is used to manage and share packages (libraries, tools, and frameworks) for Node.js, which includes packages that are commonly used in front-end development, including React.js.

- **What is the Role of Node Js in react Js?**

Answer: Node.js plays a significant role in React.js development, primarily in the context of building and managing the development environment, as well as facilitating server-side rendering and back-end integration. Node.js is a crucial part of the React.js development ecosystem, serving as the foundation for development tools, build processes, server-side rendering, and server-side logic. The combination of React.js on the front end and Node.js on the back end provides a full-stack JavaScript solution for building modern web applications. Using Node. js and

React together empowers web applications to handle heavy server loads and requests. Therefore, it allows you to work easily throughout the development process.

- **What is the CLI command in React Js?**

Answer: CLI stands for Command Line Interface. They are command-line tools that help developers automate repetitive tasks and streamline their development workflow. They are typically used to create, build, test, and deploy React applications from the command line. The Command Line Interface (CLI) is often used to perform various tasks such as creating a new React application, running a development server, building the application for production, and more. The most common CLI tool used for React.js development is create-react-app.

- **What are Components in React Js?**

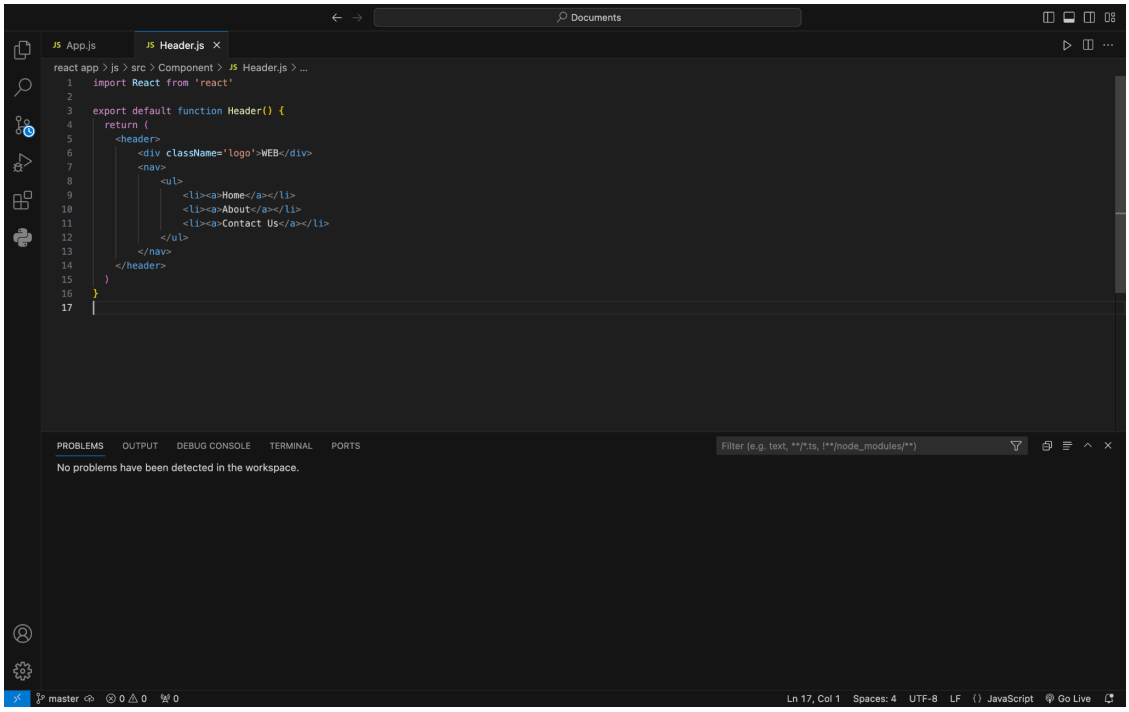
Answer: Components are the fundamental building blocks of React applications. Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components.

- **What are Header and Content Components in React Js?**

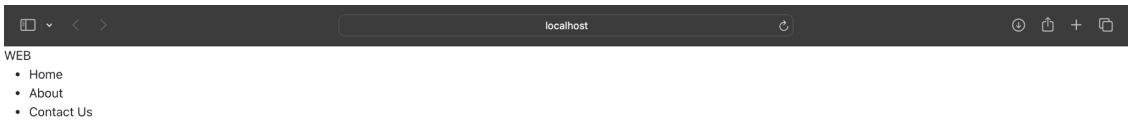
Answer: The “Header” component typically represents the top section of a web page or application. It often contains elements like the site logo, navigation menu, user authentication status, and other header-related information. The header is usually static and

remains consistent across different pages or views of the application.

For example:

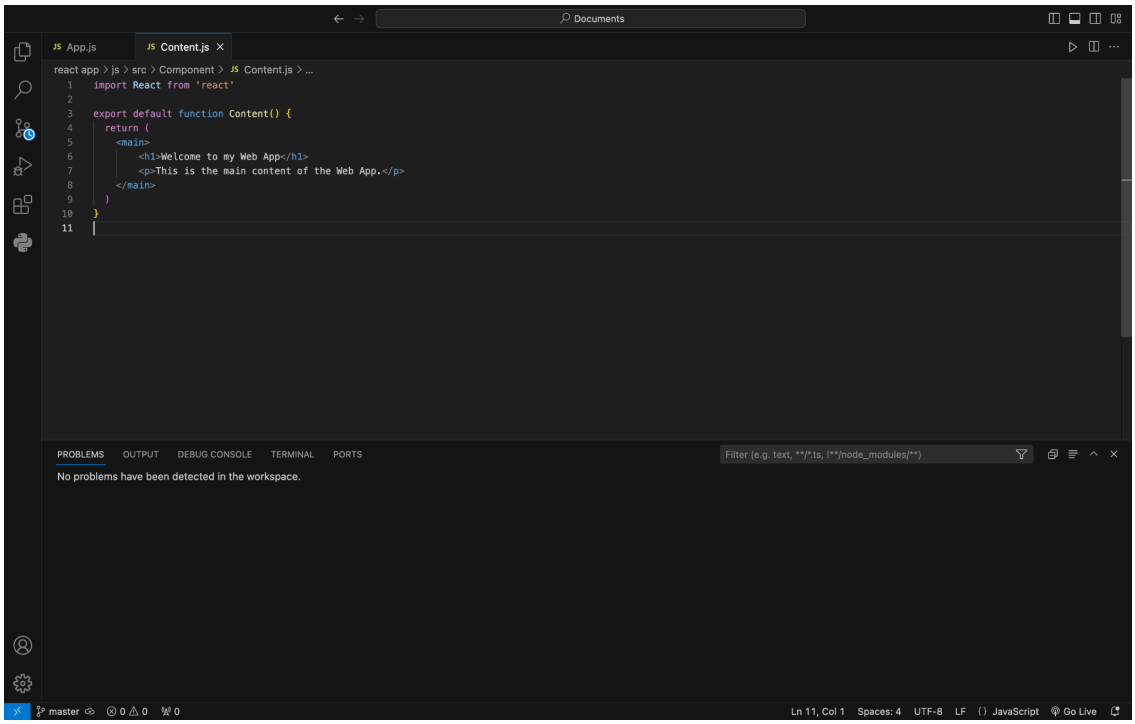


Output:

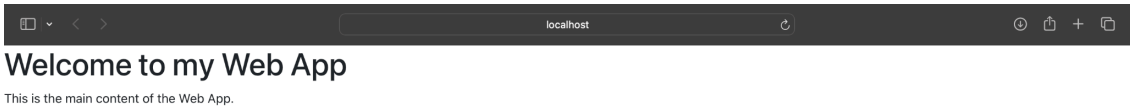


The “Content” component typically represents the main section of a web page or application where the primary content is displayed. It can vary significantly depending on the specific page or view within the application. The content section is dynamic and changes based on user interactions or navigation.

For example:



Output:



- **How to install React Js on Windows, Linux Operating System? How to install NPM and How to check the version of NPM?**

Answer: To install React Js and NPM on Windows you have to follow the below steps:
Step-1: Visit the Node.js download page at: https://nodejs.org/en/download/ .
Step-2: Download the installer for your Windows system.
Step-3: To install Node.js and npm, please run the installer and carefully follow the provided prompts.
Step-4: Install Create React App. Create React App is a command-line tool that simplifies the process of setting up a new React project with a recommended project structure and configuration. To install Create React App globally, open a command prompt and run the following command:

```
npm install -g create-react-app
```

This command installs Create React App on your system, making it available to use in any directory.
After the installation is complete, you can verify that Node.js and npm are installed by opening a command prompt and running the following commands:

```
node -v
npm -v
```

These commands should display the version numbers for Node.js and npm, respectively.

To install React Js and NPM on Linux Operating System you have to follow the below steps:
Step-1: Install npm. Login to your server as sudo user and run the following command:

```
sudo apt install npm
```

Once the installation is complete, verify the version of npm installed using the command:

```
npm --version
```

The installation of npm also installs Node.js. Confirm the version of Node installed using the command:

```
node --version
```

- **How to check the version of React Js?**

Answer: There are several ways to check the version of React Js are as follows:

1. Using Command Line:
   Step-1: Open a terminal or command prompt.

Step-2: Navigate to the root directory of your project.
Step-3: Run the following command to display the installed version of React:

```
npm list react
```

This will show the version of React installed in your project.

2. Checking package.json:
   Step-1: Open the 'package.json' file in the root directory of your React project.
   Step-2: Look for the ' "dependencies" ' section, where you'll find the version of React installed. It will be listed as ' "react": "x.x.x" '. The numbers 'x.x.x' represent the version.
   For example:

```
"dependencies": {
   "react": "^18.2.0",
 },
```

- **How to change components of React Js?**

Answer: There are several ways to change components in React JS, depending on the nature of the change and your desired functionality:

1. Updating State:

- If the change involves updating the component's internal data, use the **useState** hook to manage the state and trigger a re-render when the state changes.
- Update the state using the **setState** function within event handlers or other logic that modifies the data.
- The component will re-render with the updated state, reflecting the changes in the UI.

2. Passing Props:

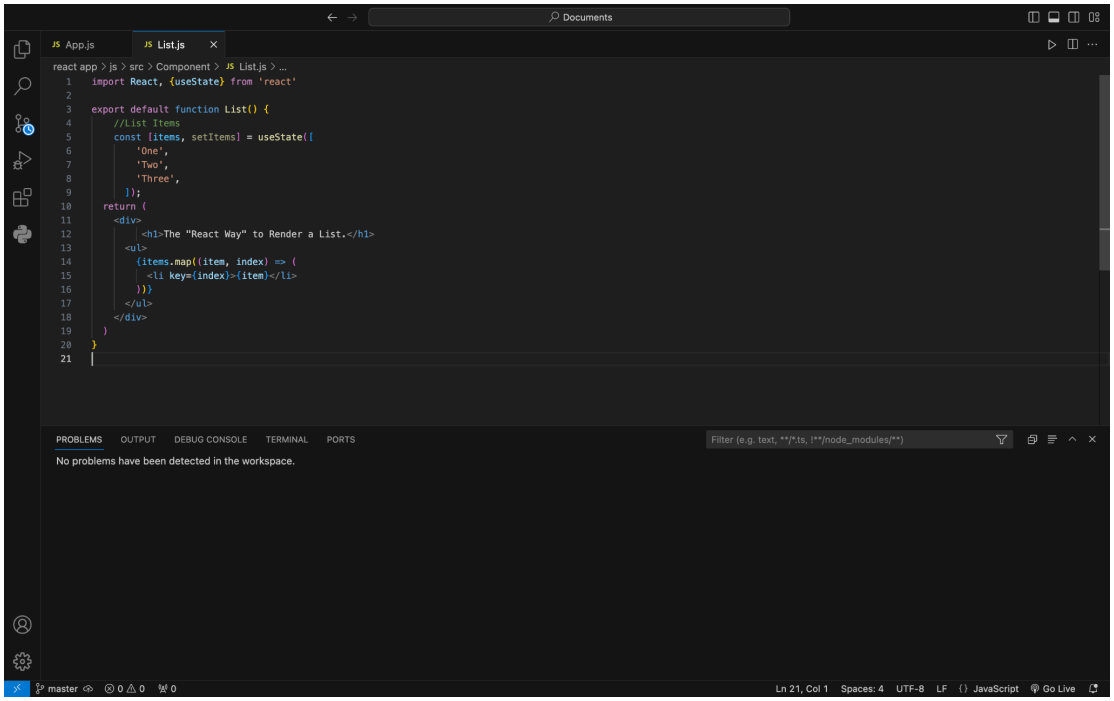
- If the change involves modifying the component based on external data, pass the data as props from the parent component.
- Update the props in the parent component when the data changes.
- The child component will receive the updated props and re-render with the new information, reflecting the change in the UI.

3. Conditional Rendering:

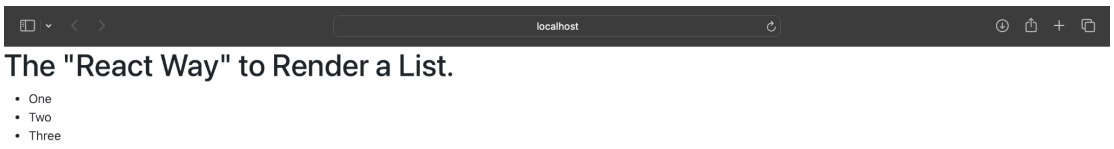
- Use conditional statements like **if** or ternary expressions (**?** **:**) to selectively render different content based on props or state.
- This allows you to dynamically show or hide parts of the component based on specific conditions.

- **How to Create a List View in React Js?**

Answer:

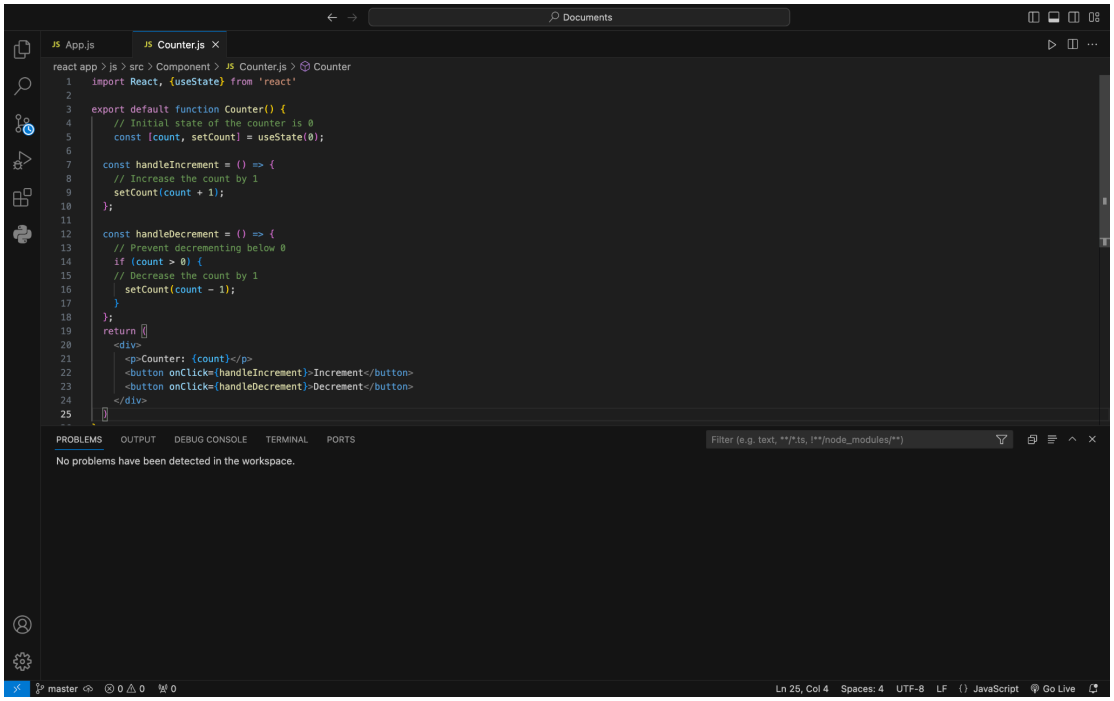


Output:



- **Create Increment decrement state change by button click?**

Answer:



Output: