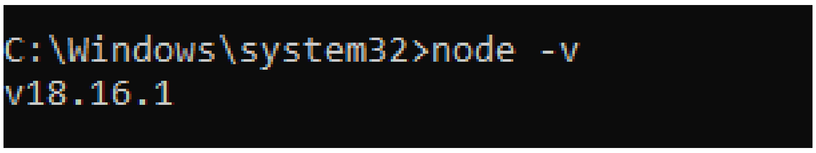
University of Colombo School of Computing

SCS 2208 - Rapid Application Development

Lab Sheet 06 - React.js

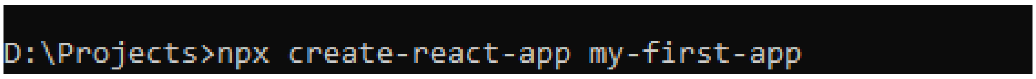
Starting with React.js

1. Set Up the Development Environment.
   - Ensure you have Node.js installed on your computer. You can download it from the official Node.js website. (https://nodejs.org/)
   - Verify that Node.js is installed correctly by opening a terminal or command prompt and running the following command.



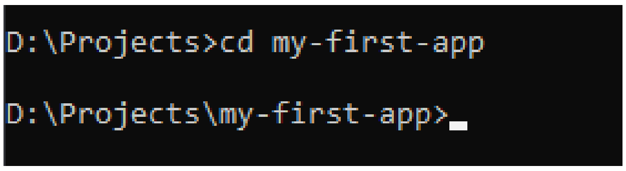
```
C:\Windows\system32>node -v
v18.16.1
```

2. Create a New React Application.
   - Open a terminal or command prompt and navigate to the directory where you want to create your React application.
   - Run the following command to create a new React application using Create React App.



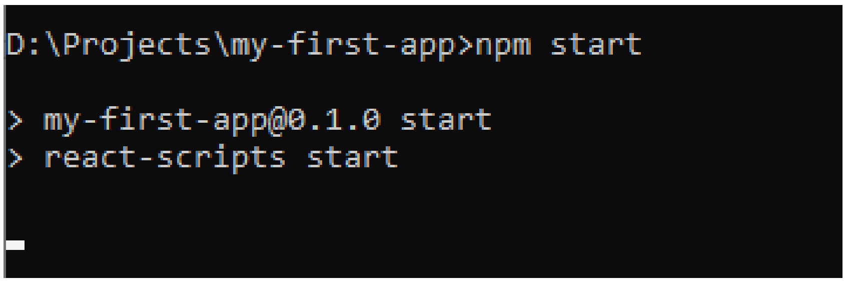
```
D:\Projects>npx create-react-app my-first-app
```

   - Once the command finishes executing, navigate into the project directory.



```
D:\Projects>cd my-first-app

D:\Projects\my-first-app>_
```

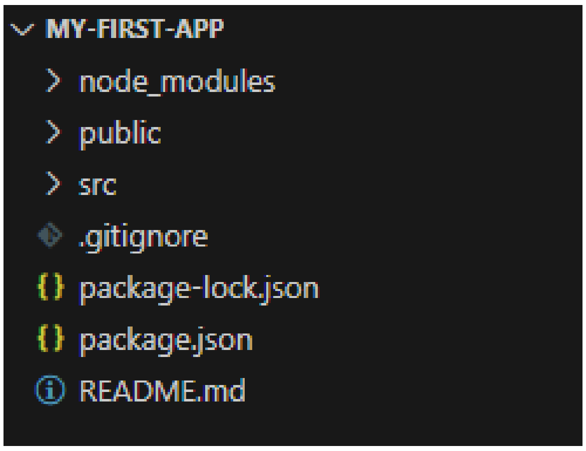
3. Start the Development Server.
   - Run the following command to start the development server.



```
D:\Projects\my-first-app>npm start

> my-first-app@0.1.0 start
> react-scripts start

```

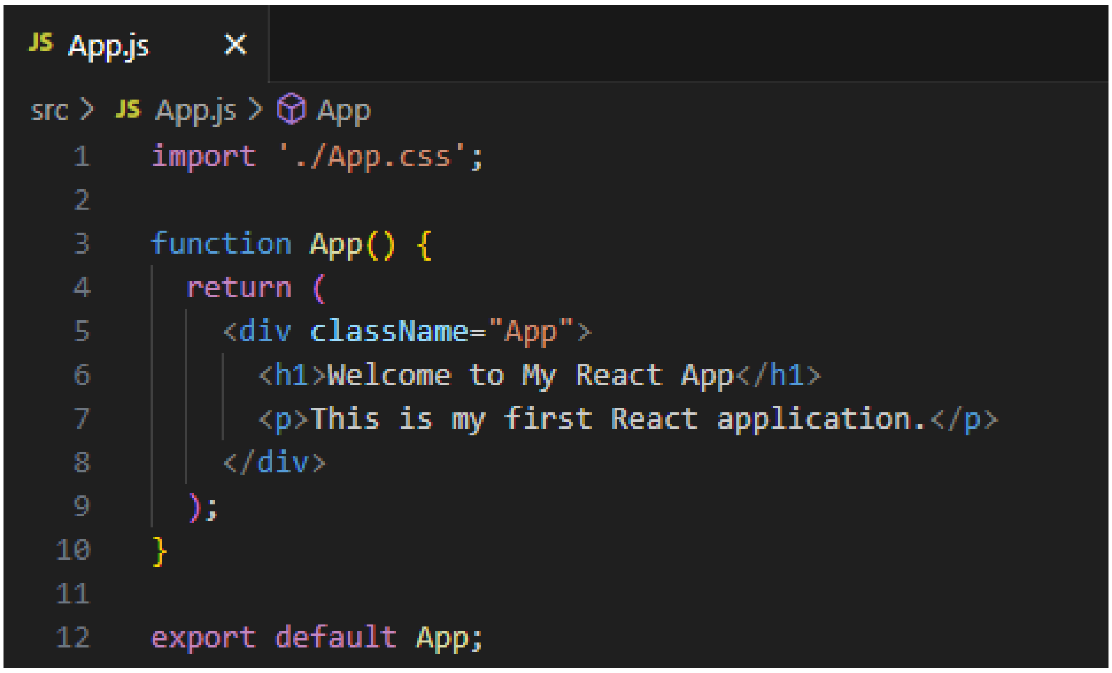
- After a few moments, your default web browser should open, displaying your React application. You can make changes to your code, and the browser will automatically update with the latest changes.

4. Explore the Project Structure.
   - Open your code editor and navigate to the project directory (my-first-app).
   - You'll find the main source code inside the **'src'** directory. The entry point for your application is the **'index.js'** file.
   - Explore the other files and directories in the project to understand their purpose.



```
MY-FIRST-APP
  > node_modules
  > public
  > src
  .gitignore
  package-lock.json
  package.json
  README.md
```

5. Make Your First Change.
   - Open the **'src/App.js'** file in your code editor.
   - Modify the JSX code inside the **'App'** component to make a simple change. For example, replace the existing content.

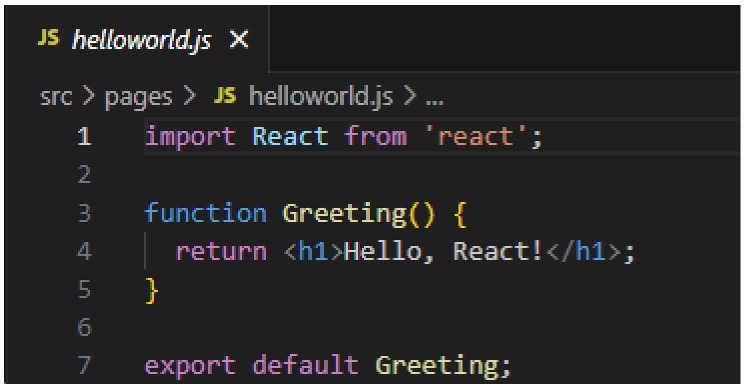


```jsx
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Welcome to My React App</h1>
      <p>This is my first React application.</p>
    </div>
  );
}

export default App;
```

- Save the file, and the development server will automatically reload the browser with the updated content.

Activities

Ex. Create a React component called "Greeting" that displays a simple greeting message, such as "Hello, React!"

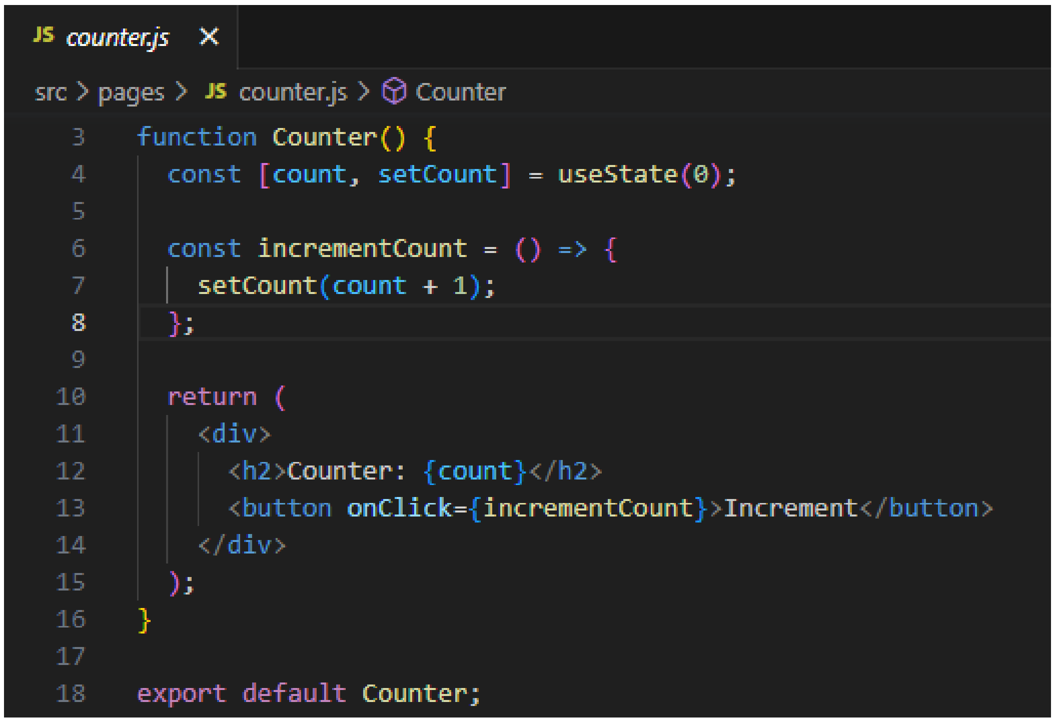


helloworld.js ×

src > pages > helloworld.js > ...

```
import React from 'react';

function Greeting() {
  return <h1>Hello, React!</h1>;
}

export default Greeting;
```

Ex. Create a React component called "Counter" that displays a button and a counter value. Clicking the button should increment the counter.



counter.js ×

src > pages > counter.js > Counter

```
function Counter() {
  const [count, setCount] = useState(0);

  const incrementCount = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <h2>Counter: {count}</h2>
      <button onClick={incrementCount}>Increment</button>
    </div>
  );
}

export default Counter;
```

1. Create a React component called "Toggle" that displays a button. Clicking the button should toggle the display of a message between "ON" and "OFF".

2. Create a React component called "Timer" that displays a timer that starts at 0 and increments every second. The timer should stop incrementing when a "Stop" button is clicked.

3. Create a React component called "TodoList" that displays a list of todo items. The todo items should be passed as an array of strings to the component.
   Sample Array: ['Learn React', 'Build a project', 'Go for a walk','Do some exercises','Join a music class','Read a novel']

4. Create a React component called "CardList" that receives an array of objects representing cards. Each card object should have a title and content property. Render each card as a separate card component.
   Sample Array: [{title: 'Card 1',content: 'This is the content of Card 1'},{title: 'Card 2',content: 'This is the content of Card 2'},{title: 'Card 3',content: 'This is the content of Card 3'}]

5. Create a React component called "Form" that includes a text input and a button. Clicking the button should display an alert with the value entered in the input.