**REACT - ASSIGNMENT**

**1. Explain React Components**

React **components** are the building blocks of any React application. They let you split the UI into **independent, reusable pieces**, and think about each piece in isolation.

* Components can be **functions or classes**.
* Each component returns **JSX** that describes what the UI should look like.
* Components can accept **props** (input data) and manage **state** (data that changes over time).

**2. Identify the Differences Between Components and JavaScript Functions**

|  | **React Component** | **JavaScript Function** |
| --- | --- | --- |
|  | Used to render UI in React | Used for general-purpose logic |
|  | Returns JSX (UI code) | Returns a value or performs a task |
|  | Has lifecycle methods (in class components) | No lifecycle |
|  | Can hold and manage state (with hooks or classes) | Cannot hold state |
|  | Returns JSX syntax | Returns primitive values (numbers, strings, etc.) |

**3. Identify the Types of Components**

There are **two main types** of React components:

1. **Class Components**
2. **Function Components**

**4. Explain Class Component**

A **class component** is a JavaScript class that extends React.Component and has a render() method which returns JSX.

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Hello, {this.props.name}</h1>;

}

}

**Key Features:**

* Uses ES6 class syntax
* Can have a **constructor**
* Can use **state** and **lifecycle methods**

**5. Explain Function Component**

A **function component** is a simpler way to write components using a JavaScript function. It returns JSX and can use **React Hooks** to manage state and lifecycle.

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

Or using hooks:

import { useState } from 'react';

function Counter() {

const [count, setCount] = useState(0);

return <button onClick={() => setCount(count + 1)}>Click {count}</button>;

}

**6. Define Component Constructor**

The **constructor** is a special method used in class components to:

* Initialize the state
* Bind event handler methods

It runs before the component is mounted.

class Welcome extends React.Component {

constructor(props) {

super(props);

this.state = { message: "Hello" };

}

}

**7. Define render() Function**

The **render()** method is required in class components. It tells React **what to display** on the screen. It must return **JSX**.

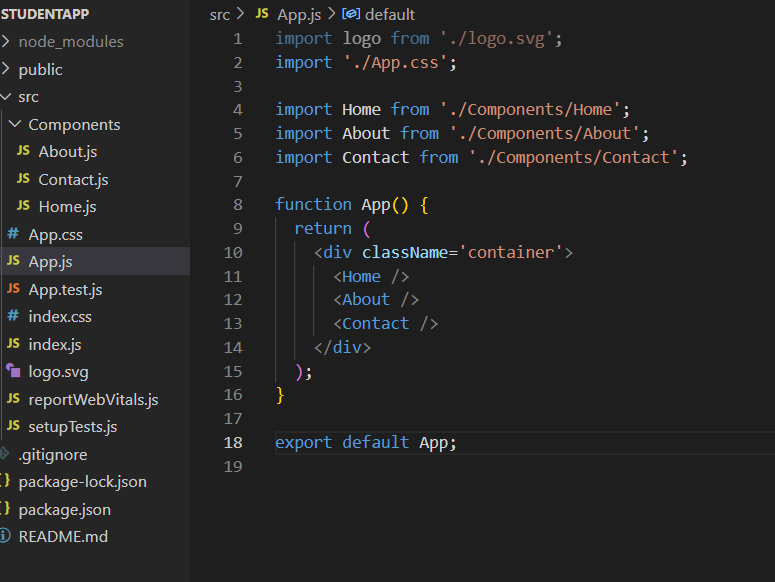
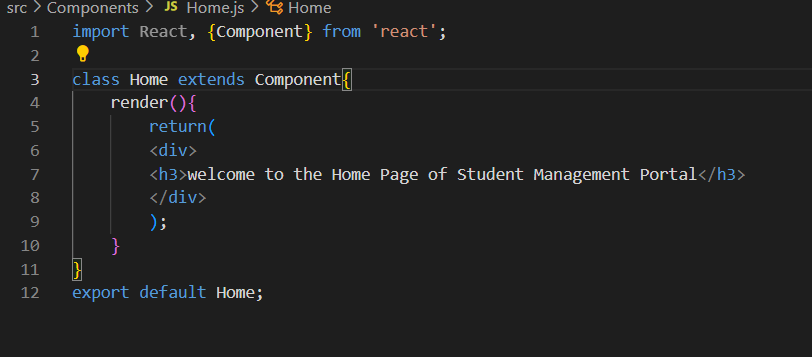
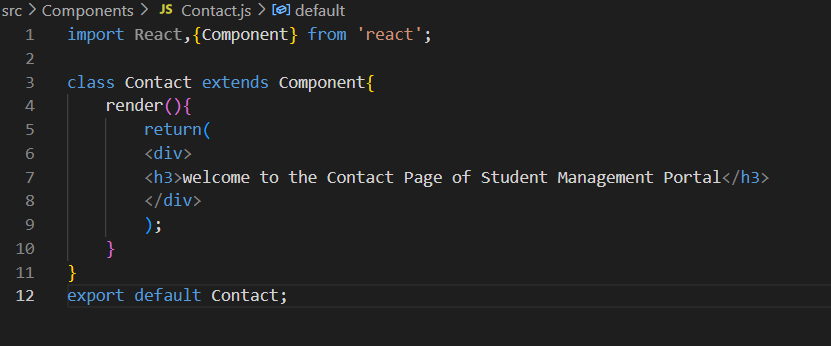
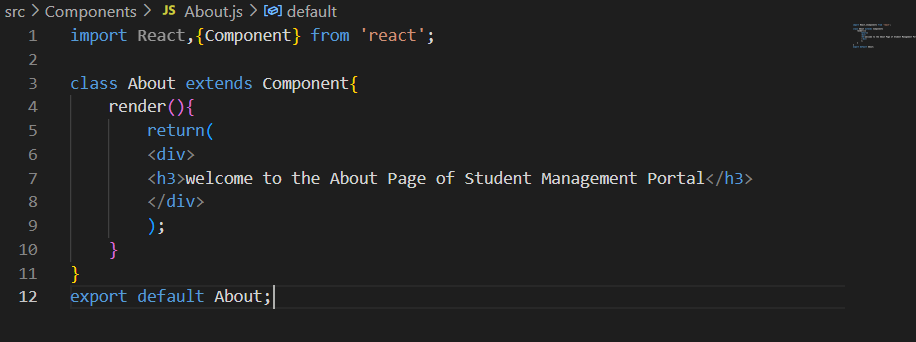
render() {

return <h1>Welcome to React!</h1>;

}

* It gets called whenever the component’s **state or props** change.

HANDS-ON:

OUTPUT:

