

CREATE PROJECT OF REACT USING CDN (CONTENT DELIVERY NETWORK)

You can run a React project using a CDN without the need for a local development environment or npm. Keep in mind that using a CDN might limit your ability to utilize certain features or libraries that require a more extensive build process.

To run a React project using a CDN (Content Delivery Network), you can follow these steps:

1. **Set up your HTML file:** Create an HTML file where you will include the necessary scripts and stylesheets. You can use a code editor of your choice to create a new file with a .html extension.
2. **Add React and ReactDOM CDN links:** Include the following `<script>` tags in the `<head>` section of your HTML file to include React and ReactDOM from a CDN.

```
<head>
<script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
</head>
```

3. **Add Babel CDN link:** *If you are using JSX syntax, you'll need Babel to transpile it to regular JavaScript. Include the following `<script>` tag in the `<head>` section to include Babel from a CDN.*

```
<head>
<script src="https://unpkg.com/@babel/standalone@7.15.4/babel.min.js"></script>
</head>
```

4. **Create a container element:** In the `<body>` section of your HTML file, create a container element where your React component will be rendered. Give it an appropriate id.

```
<body>
<div id="root"></div>
</body>
```

5. **Write your React component:** Right below the closing `</div>` tag of the container element, write your React component using JSX or regular JavaScript.

```
<body>
<div id="root"></div>

<script type="text/babel">
  // Your React component code here
</script>
</body>
```

6. **Mount your React component:** Inside the `<script>` tag, use `ReactDOM.render()` to mount your React component to the container element.

TechCarrel

```
<body>
<div id="root"></div>
<script type="text/babel">
  ReactDOM.render(
    // Your React component here,
    document.getElementById('root')
  );
</script>
</body>
```

7. **Open the HTML file in a browser:** Save the HTML file and open it in a web browser. You should see your React component rendered in the browser.

e.g.-1

```
<html>
<head>
  <title>React App</title>
  <script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
  <script src="https://unpkg.com/@babel/standalone@7.15.4/babel.min.js"></script> </head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    class Hello extends React.Component {
      render() {
        return <div>Hello React code </div>;
      }
    }
    ReactDOM.render(<Hello />, document.getElementById("root"));
  </script>
</body>
</html>
```

e.g.-2

```
<html>
<head>
  <title>React App</title>
  <script src="https://unpkg.com/react@17.0.2/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.development.js"></script>
  <script src="https://unpkg.com/@babel/standalone@7.15.4/babel.min.js"></script>
</head>
<body>
  <div id="root"></div>
  <script type="text/babel">
    function Test()
    {
      const func={()=>{alert("Hello React")}}
      return <button onClick={func}>Click Me</button>
    }
  </script>
</body>
</html>
```

[Type text]

TechCarrel

```
class Hello extends React.Component{
  render()
  {
    return <div>Hello React code <Test /></div>
  }
}
ReactDOM.render(<Hello />,document.getElementById('root'))
</script>
</body>
</html>
```

Note: if we do not use virtual DOM code become lengthy

e.g.

```
<html>
<body>
  <div id="div1" onclick="add()" style="border: 1px solid red">
    <p id="p1">one</p>
    <p id="p2">two</p>
  </div>
  <script>
    function add() {
      const p1 = document.createElement("p");
      const node = document.createTextNode("Three");
      p1.appendChild(node);
      const element = document.getElementById("div1");
      element.appendChild(p1);
    }
  </script>
</body>
</html>
```