

Steps to Download Node.js and Create a React Project

step:1 → Download and Install Node.js:

- Visit [Node.js official website](https://nodejs.org/).
- Download the latest LTS (Long-Term Support) version for stability.
- Install Node.js by running the downloaded installer.
- Verify installation by opening a terminal and typing:

```
node -v
```

```
npm -v
```

step:2 → Install Vite and Create a React Project

step:1 -> run command => npm create vite@latest

step:2 ->press enter to say yes that i want to create project

step:3 ->Give your project name=>demo

step:4 ->Select the framework -> React

step:5 ->Select the language (variant) ->JavaScript

step:6 ->select the directory ->

```
cd projectName=>[demo] ///## or you can just cut the cmd...and go inside the  
project directory(folder)
```

```
npm install
```

```
npm run dev
```

step:3 → Where to Write Code in React

Main Project Structure:

Inside the srcfolder, you will find the main files where you write code. Common files include:

- index.js - Entry point of the React app.
- App.js - Main component file where most of the code goes.

You can add or modify components in the App.js file or create new component files in the src folder.

step:4 → How to use Inline CSS in React

To apply inline CSS in React, use the style attribute with a JavaScript object for the styles.

Example:

```
function App() {
  return (
    <div style={{ backgroundColor: 'lightblue', padding: '20px' }}>
      <h1 style={{ color: 'red' }}>Hello, React!</h1>
    </div>
  );
}

export default App;
```

step:5 → Using Bootstrap in React

- **Install Bootstrap:** Install Bootstrap using npm:

npm install bootstrap

- **Import Bootstrap in index.js:** In the src/index.js file, add the following import statement to include Bootstrap CSS:

import 'bootstrap/dist/css/bootstrap.min.css';

- **Use Bootstrap Classes:** Now you can use Bootstrap classes in your components like so:

```
function App() {
  return (
    <div className="container">
      <h1 className="text-primary">Hello, Bootstrap with React!</h1>
      <button className="btn btn-success">Click Me</button>
    </div>
  );
}

export default App;
```

step: 6 → Using Routing in React (with react-router-dom)

Install react-router-dom: To manage routing in React, install react-router-dom:

npm install react-router-dom

- **Create Separate Files for Home.js and About.js**

Create Home.js in the src Folder: Inside the src folder, create a new file called Home.js and add the following code:

```
// src/Home.js
import React from 'react';

function Home() {
  return <h2>Home Page</h2>;
}

export default Home;
```

- **Create About.js in the src Folder:** Similarly, create a file called About.js in the src folder:

```
// src/About.js
import React from 'react';

function About() {
  return <h2>About Page</h2>;
}

export default About;
```

step:7 → **Setup Routes:** In App.js, import the necessary components and define the routes:

```
// src/App.jsx
import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';
import Home from './Home';
import About from './About';

function App() {
  return (
    <Router>
      <div>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/about" element={<About />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

step:8 → Using the Link Component for Navigation

Instead of using regular anchor (<a>) tags, React Router provides the Link component for navigation.

1. **Import Link:** Add the import for Link from react-router-dom:

```
import { Link } from 'react-router-dom';
```

2. **Use Link for Navigation:** Replace <a> tags with the Link component:

```
<Router>
  <div>
    <nav>
      <Link to="/">Home</Link>
      <Link to="/about">About</Link>
    </nav>
    <Routes>
      <Route path="/" element={ <Home /> } />
      <Route path="/about" element={ <About /> } />
    </Routes>
  </div>
</Router>
);
}

export default App;
```