

# to do application in react step by step

Creating a To-Do application in React is a great way to learn the basics of React, including managing state and handling user interactions. Here's a step-by-step guide to create a simple To-Do application in React:

## Step 1: Set Up Your React Project

Before you start, make sure you have Node.js and npm installed on your computer. If not, you can download them from the [Node.js website](https://nodejs.org/).

1. Create a new React app:

```
npx create-react-app todo-app
cd todo-app
```

1. Open your code editor in the `todo-app` directory.

## Step 2: Create a To-Do Component

In this step, you'll create a To-Do component that will manage the state of your To-Do list.

1. Inside the `src` folder, create a new file named `Todo.js`.
2. In `Todo.js`, define your To-Do component:

```
2. import React, { useState } from 'react';
3.
4. function Todo() {
5.   const [todos, setTodos] = useState([]);
6.   const [task, setTask] = useState('');
7.
8.   const handleTaskChange = (event) => {
9.     setTask(event.target.value);
10.  };
11.
12.   const handleAddTodo = () => {
13.     if (task.trim() !== '') {
14.       setTodos([...todos, task]);
15.       setTask('');
16.     }
17.  };
18.
19.   const handleDeleteTodo = (index) => {
20.     const newTodos = [...todos];
21.     newTodos.splice(index, 1);
22.     setTodos(newTodos);
23.  };
24.
25.   return (
26.     <div>
27.       <h1>To-Do List</h1>
28.       <input
29.         type="text"
30.         placeholder="Add a new task"
```

```

31.         value={task}
32.         onChange={handleTaskChange}
33.     />
34.     <button onClick={handleAddTodo}>Add</button>
35.     <ul>
36.         {todos.map((todo, index) => (
37.             <li key={index}>
38.                 {todo}
39.                 <button onClick={() =>
handleDeleteTodo(index)}>Delete</button>
40.             </li>
41.         ))}
42.     </ul>
43. </div>
44. );
45. }
46.
47. export default Todo;
48.

```

### Step 3: Render the To-Do Component

In the `src/App.js` file, import and render the `Todo` component.

```

import React from 'react';
import './App.css';
import Todo from './Todo';

function App() {
  return (
    <div className="App">
      <Todo />
    </div>
  );
}

export default App;

```

### Step 4: Style Your To-Do App (Optional)

You can add CSS styles to make your To-Do app look more appealing. Create a `Todo.css` file in the `src` folder and add your styles. Then, import the CSS file into your `Todo.js` component.

### Step 5: Start the Development Server

Run the following command to start your React app:

```
npm start
```

Your To-Do application should now be running at `http://localhost:3000`, and you can start adding and deleting tasks.

Congratulations! You've created a simple To-Do application in React. You can further enhance it by adding features like editing tasks, marking tasks as completed, or saving tasks to local storage for persistence.

