

Build a simple React App

Step 1: Setup the Project

1. Install Node.js and npm:

- Download and install Node.js from [here](#).
- npm (Node Package Manager) comes bundled with Node.js.

2. Create a New React Project:

- Open your terminal/command prompt.
- Run the following command to create a new React app (replace todo-list with your desired project name):

```
npx create-react-app todo-list
```

3. Navigate to the Project Directory:

```
cd todo-list
```

4. Start the Development Server:

```
npm start
```

- This will start the development server and open your new React app in the browser.

Step 2: Create the To-do List Component

1. Open the Project in Your Code Editor:

- Use your preferred code editor (e.g., VSCode) to open the todo-list directory.

2. Create a To-do List Component:

- Inside the src folder, create a new file called TodoList.js.

3. Edit TodoList.js to Add the Component:

```
import React, { useState } from 'react';

function TodoList() {
  const [todos, setTodos] = useState([]);
  const [inputValue, setInputValue] = useState('');

  const handleAddTodo = () => {
    if (inputValue.trim() !== '') {
      setTodos([...todos, inputValue]);
      setInputValue('');
    }
  };

  const handleInputChange = (e) => {
    setInputValue(e.target.value);
  };

  const handleDeleteTodo = (index) => {
    const newTodos = todos.filter((todo, todoIndex) => todoIndex !== index);
    setTodos(newTodos);
  };
}
```

```

    return (
      <div>
        <h1>To-do List</h1>
        <input type="text" value={inputValue}
onChange={handleInputChange} />
        <button onClick={handleAddTodo}>Add</button>
        <ul>
          {todos.map((todo, index) => (
            <li key={index}>
              {todo} <button onClick={() =>
handleDeleteTodo(index)}>Delete</button>
            </li>
          )));
        </ul>
      </div>
    );
}

export default TodoList;

```

Step 3: Integrate the Component into Your App

1. Edit `App.js` to Include the To-do List Component:

```

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

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <TodoList />
      </header>
    </div>
  );
}

export default App;

```

2. Add Basic Styling (Optional):

- o Open `src/App.css` and add some basic styling:

```

.App {
  text-align: center;
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

```

```

input {
  padding: 10px;
  margin: 10px;
  font-size: 16px;
}

button {
  padding: 10px 20px;
  margin: 10px;
  font-size: 16px;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  margin: 10px 0;
  font-size: 18px;
}

li button {
  margin-left: 20px;
}

```

Step 4: Run Your App

1. Start the Development Server Again (if it's not running):

```
npm start
```

2. Open Your Browser:

- o Navigate to <http://localhost:3000/> to see your To-do List app in action.

Step 5: Additional Enhancements (Optional)

1. Add Persistence:

- o You can use `localStorage` to save and load the to-dos.

2. Add More Features:

- o Add editing functionality for the to-dos.
- o Add filtering (e.g., show only completed tasks).

Example of Adding Persistence with `localStorage`:

1. Update `TodoList.js` to Use `localStorage`:

```

import React, { useState, useEffect } from 'react';

function TodoList() {
  const [todos, setTodos] = useState(() => {
    const savedTodos = localStorage.getItem('todos');
    return savedTodos ? JSON.parse(savedTodos) : [];
  });
  const [inputValue, setInputValue] = useState('');

  useEffect(() => {

```

```
localStorage.setItem('todos', JSON.stringify(todos));
}, [todos]);

const handleAddTodo = () => {
  if (inputValue.trim() !== '') {
    setTodos([...todos, inputValue]);
    setInputValue('');
  }
};

const handleInputChange = (e) => {
  setInputValue(e.target.value);
};

const handleDeleteTodo = (index) => {
  const newTodos = todos.filter((todo, todoIndex) => todoIndex !== index);
  setTodos(newTodos);
};

return (
  <div>
    <h1>To-do List</h1>
    <input type="text" value={inputValue}
      onChange={handleInputChange} />
    <button onClick={handleAddTodo}>Add</button>
    <ul>
      {todos.map((todo, index) => (
        <li key={index}>
          {todo} <button onClick={() =>
            handleDeleteTodo(index)}>Delete</button>
        </li>
      ))}
    </ul>
  </div>
);
}

export default TodoList;
```

By following these steps, you should have a fully functional React To-do List app.