# Task Manager with React

## Folder Structure (client)

|  |
| --- |
| client/  │── src/  │ ├── components/  │ │ ├── PageHeader.jsx  │ ├── pages/  │ │ ├── **TaskList**.jsx  │ │ ├── TaskCreate.jsx  │ │ ├── TaskView.jsx  │ ├── App.jsx  │ ├── main.jsx  │── package.json |

## 1. Install Dependencies & Run

|  |
| --- |
| npm install  npm run dev |

# 2. React Components

### PageHeader.jsx (Navigation Menu)

|  |
| --- |
| import { Link } from "react-router-dom";  const PageHeader = () => (  <nav className="navbar navbar-expand-lg bg-body-tertiary">  <div className="container-fluid">  <Link className="navbar-brand" to="/">Task Management System</Link>  <div className="collapse navbar-collapse">  <ul className="navbar-nav me-auto mb-2 mb-lg-0">  <li className="nav-item">  <Link className="nav-link" to="/">Tasks</Link>  </li>  <li className="nav-item">  <Link className="nav-link" to="/create">Add Task</Link>  </li>  </ul>  </div>  </div>  </nav>  );  export default PageHeader; |

### TaskList.jsx (List of Tasks)

|  |
| --- |
| import PageHeader from "../components/PageHeader";  import { Link } from "react-router-dom";  const TaskList = () => {  const tasks = [  { id: 1, name: "Complete UI Design", status: "Pending" },  { id: 2, name: "Setup Backend API", status: "Completed" }  ];  return (  <div>  <PageHeader />  <div className="container">  <h3>List of Tasks</h3>  <table className="table table-success table-striped">  <thead className="table-dark">  <tr>  <th>ID</th>  <th>Task Name</th>  <th>Status</th>  <th></th>  </tr>  </thead>  <tbody>  {tasks.map(task => (  <tr key={task.id}>  <td>{task.id}</td>  <td>{task.name}</td>  <td>{task.status}</td>  <td><Link to={`/view/${task.id}`} className="btn btn-success">View</Link></td>  </tr>  ))}  </tbody>  </table>  </div>  </div>  );  };  export default TaskList; |

### TaskCreate.jsx (Add Task)

|  |
| --- |
| import PageHeader from "../components/PageHeader";  import { useState } from "react";  const TaskCreate = () => {  const [task, setTask] = useState({ name: "", status: "" });  const handleSubmit = (e) => {  e.preventDefault();  alert(`Task Created: ${task.name}, Status: ${task.status}`);  };  return (  <div>  <PageHeader />  <div className="container">  <h3>Add Task</h3>  <form onSubmit={handleSubmit}>  <div className="mb-3">  <label className="form-label">Task Name:</label>  <input type="text" className="form-control"  onChange={(e) => setTask({ ...task, name: e.target.value })}  placeholder="Enter task name" />  </div>  <div className="mb-3">  <label className="form-label">Task Status:</label>  <select className="form-control" onChange={(e) => setTask({ ...task, status: e.target.value })}>  <option value="">Select Status</option>  <option value="Pending">Pending</option>  <option value="Completed">Completed</option>  </select>  </div>  <button type="submit" className="btn btn-primary">Create Task</button>  </form>  </div>  </div>  );  };  export default TaskCreate; |

### TaskView.jsx (View Task)

|  |
| --- |
| import PageHeader from "../components/PageHeader";  import { useParams } from "react-router-dom";  const tasks = [  { id: 1, name: "Complete UI Design", status: "Pending" },  { id: 2, name: "Setup Backend API", status: "Completed" }  ];  const TaskView = () => {  const { id } = useParams();  const task = tasks.find(t => t.id === parseInt(id));  return (  <div>  <PageHeader />  <div className="container">  <h3>View Task</h3>  <div className="form-group mb-3">  <label className="form-label">Task Name:</label>  <div className="form-control">{task?.name || "Not Found"}</div>  </div>  <div className="form-group mb-3">  <label className="form-label">Task Status:</label>  <div className="form-control">{task?.status || "Not Found"}</div>  </div>  </div>  </div>  );  };  export default TaskView; |

### 3. Routing - App.jsx

|  |
| --- |
| import { BrowserRouter as Router, Routes, Route } from "react-router-dom";  import TaskList from "./pages/TaskList";  import TaskCreate from "./pages/TaskCreate";  import TaskView from "./pages/TaskView";  const App = () => (  <Router>  <Routes>  <Route path="/" element={<TaskList />} />  <Route path="/create" element={<TaskCreate />} />  <Route path="/view/:id" element={<TaskView />} />  </Routes>  </Router>  );  export default App; |

### 4. Entry Point - main.jsx

|  |
| --- |
| import React from "react";  import ReactDOM from "react-dom/client";  import App from "./App";  import "bootstrap/dist/css/bootstrap.min.css";  ReactDOM.createRoot(document.getElementById("root")).render(  <React.StrictMode>  <App />  </React.StrictMode>  ); |

## How to Run?

1. **Navigate to client/ folder**

|  |
| --- |
| cd client |

1. **Install dependencies**

|  |
| --- |
| npm install |

1. **Run the app**

|  |
| --- |
| npm run dev |

# Line-by-Line Explanation of Code

Now, let’s break down each file and explain every line of code in detail.

## 1. PageHeader.jsx - Navigation Menu

import { Link } from "react-router-dom";

import { Link } from "react-router-dom"; → Imports the Link component from React Router, which is used for navigation between pages **without refreshing the page**.

const PageHeader = () => (

Defines a functional component called PageHeader. Functional components return JSX.

<nav className="navbar navbar-expand-lg bg-body-tertiary">

* <nav> → Creates a navigation bar.
* className="navbar navbar-expand-lg bg-body-tertiary" → Uses Bootstrap classes to style the navbar.

<div className="container-fluid">

* container-fluid ensures the navbar spans the full width of the page.

<Link className="navbar-brand" to="/">Task Management System</Link>

* <Link> is used instead of <a> to enable **client-side navigation** in React.
* to="/" → Clicking this navigates to the **home (task list) page**.

<div className="collapse navbar-collapse">

<ul className="navbar-nav me-auto mb-2 mb-lg-0">

* navbar-collapse makes the navbar responsive.
* navbar-nav creates a Bootstrap-styled menu.
* me-auto mb-2 mb-lg-0 adds margin and spacing.

<li className="nav-item">

<Link className="nav-link" to="/">Tasks</Link>

</li>

* Creates a navigation link to the Task List page.

<li className="nav-item">

<Link className="nav-link" to="/create">Add Task</Link>

</li>

* Creates a navigation link to the **Task Create page**.

export default PageHeader;

* Exports the PageHeader component so it can be imported into other files.

## 2. TaskList.jsx - List of Tasks

import PageHeader from "../components/PageHeader";

import { Link } from "react-router-dom";

* Imports the PageHeader component and Link from react-router-dom for navigation.

const TaskList = () => {

const tasks = [

{ id: 1, name: "Complete UI Design", status: "Pending" },

{ id: 2, name: "Setup Backend API", status: "Completed" }

];

* Defines a **hardcoded array** of tasks.
* id is used to uniquely identify each task.

return (

<div>

<PageHeader />

<div className="container">

* Renders the PageHeader component and creates a Bootstrap-styled container.

<h3>List of Tasks</h3>

<table className="table table-success table-striped">

* Displays a **Bootstrap-styled table** with a green success color.

<thead className="table-dark">

<tr>

<th>ID</th>

<th>Task Name</th>

<th>Status</th>

<th></th>

</tr>

</thead>

* Defines table headers for Task ID, Task Name, and Status.

<tbody>

{tasks.map(task => (

<tr key={task.id}>

* Maps over the tasks array to dynamically create rows.

<td>{task.id}</td>

<td>{task.name}</td>

<td>{task.status}</td>

* Displays the task details inside the table.

<td><Link to={`/view/${task.id}`} className="btn btn-success">View</Link></td>

* Creates a **"View" button** that links to TaskView.jsx, passing the task ID.

## 3. TaskCreate.jsx - Add Task Form

import { useState } from "react";

* Imports useState to manage form inputs.

const TaskCreate = () => {

const [task, setTask] = useState({ name: "", status: "" });

* Initializes task state with empty values.

const handleSubmit = (e) => {

e.preventDefault();

alert(`Task Created: ${task.name}, Status: ${task.status}`);

};

* Prevents the default form submission and shows an alert with entered data.

<form onSubmit={handleSubmit}>

* Handles form submission.

<input type="text" className="form-control"

onChange={(e) => setTask({ ...task, name: e.target.value })}

* Updates name in the task state when typing.

<select className="form-control" onChange={(e) => setTask({ ...task, status: e.target.value })}>

<option value="">Select Status</option>

<option value="Pending">Pending</option>

<option value="Completed">Completed</option>

</select>

* Updates the status in the task state.

## 4. TaskView.jsx - Viewing a Task

import { useParams } from "react-router-dom";

* useParams() extracts the id from the URL.

const task = tasks.find(t => t.id === parseInt(id));

* Finds the task that matches the ID from the URL.

<div className="form-control">{task?.name || "Not Found"}</div>

* Displays the task name or "Not Found" if the task doesn't exist.

## 5. App.jsx - Routing

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

* Imports React Router for navigation.

<Router>

<Routes>

<Route path="/" element={<TaskList />} />

<Route path="/create" element={<TaskCreate />} />

<Route path="/view/:id" element={<TaskView />} />

</Routes>

</Router>

* Defines different routes for navigating between pages.