SURAJ S KADUVETTI
FOIS
4077224OO15

Camlin Page

ASSIGNMENT 6

1 Next.js

Next.js is a modern, open source web development framework built on top of React, matined by vercel. By 2025 it is reassigned as one of the most powerful, flexible and future-rich solution for building web applications and websites

Next.js extends React basic capabilities prominantly client-side rendering by natively supporting robust features for sever side rendering (SSR), static side generation (SSG) and hybrid sites giving developers the tools for high performance scalable and SEO friendly web application

Core principal and philosophy

Performance first
Flexibility
Developer Experience
Full Stack Capabilities

Key Architectural Features

App Directory & file based routing
Rendering options
Turbo pack Compiler
Image and Asset Optimization
Advanced SEO tools

Camlin Page

Why Adapt Next.js

Improved scalability, Performance and SEO out of the box

Optionally manage frontend, backend in one to codebase

Intuitive, file based setup reduces boilerpoint

Stays compatible with cutting edge tooling, including react severs components, streaming and edge computing.

Future proof - Built in with trends such as WASM, headers CMS, module first strategies and AI integration in mind.

2 Next.js Features

Server Side Rendering (SSR) - Render pages on the server per request improsoning both load speed and SEO for dynamic content

Static Site Generation (SSG) - Prerenders pages at built time, but for pages that dont change after, resulting in fast load times

Instrumental Static Regeneration (ISR) - Combines strengths of SSG and SSR allowing static pages to be updated after displayment as data changes without rebuilds

API Routes -

Camlin Page

API Routes - Enables to create backend endpoints within app eliminating the need for seperate server

File based routing - Automatic creation of routes based on file structure in the pages on app directories

Automating Code Splitting - Loads only necessary JS for each page speeding up loading and navigation

Hot Module Replacement (HMR) - Updates modules in real time during development for faster workflows

Result in CSS/SaaS support straightforward integration of CSS and SCSS files into components

Image Optimization - Responsive image delivery and conversation for improved performance

Streaming and edge rendering - Supports modern feature like React server components and edge function for low-latency location aware apps

SEO and Assessibility Enhancements - Integrate meta tag control and impound Schema markup

Advanced performance tool - Turbopack bundles for rapid builds and improved resource management

Camlin Page

3 ReactJS vs NextJS

ReactJS	NextJS
Javascript library for UI	Framework built on top of React
Client side only	Server Side Rendering, Static support (SSR/SSG)
Use third party libraries	Built in file based routing
Poor out of box for dynamic content	Excellent due to SSG, SSR
Easy, basic webapp development	Require React knowledge advance concepts
SPAs, UIs, Module apps	Full stack apps, JAM stack, hybrid

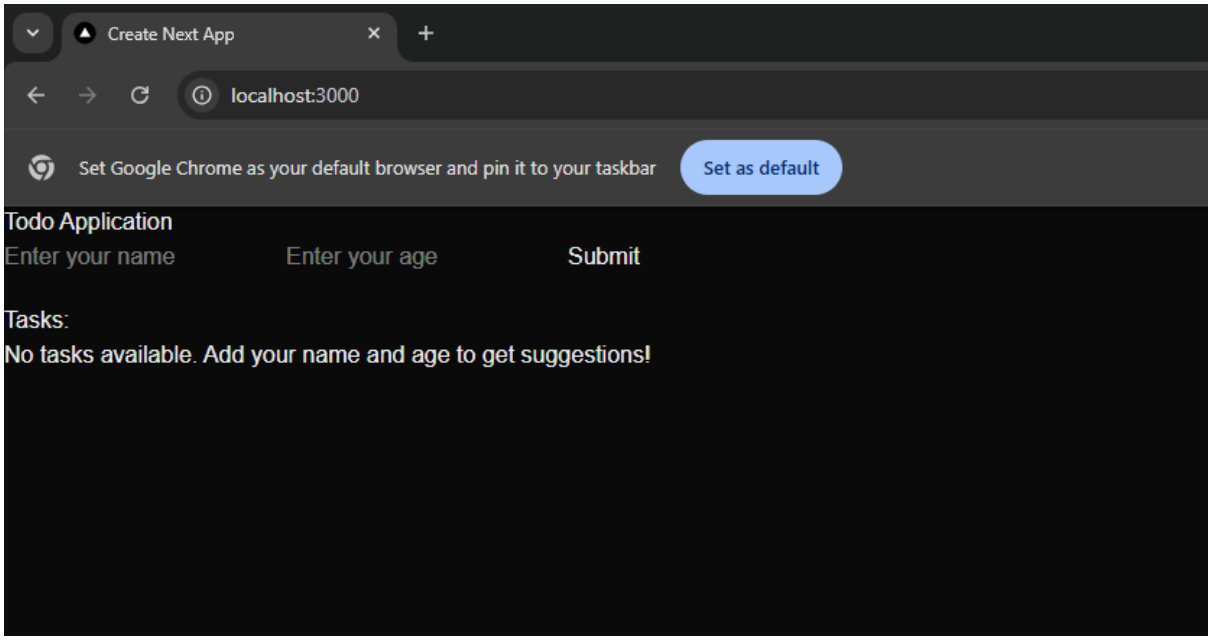
4 ESlint

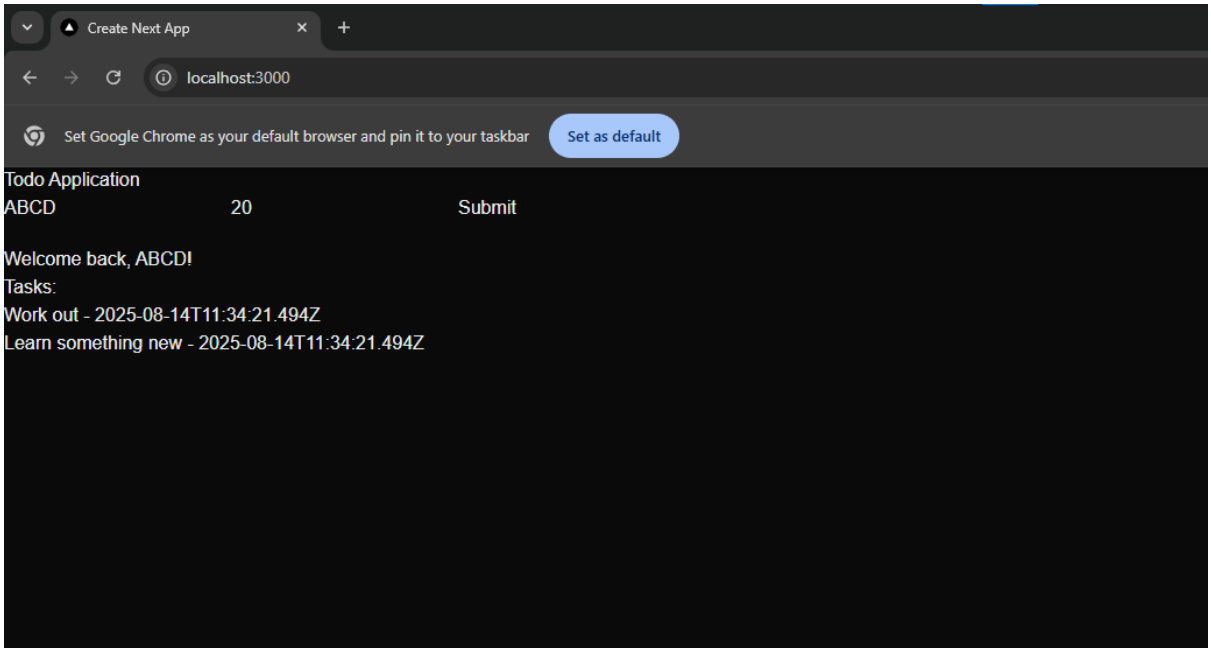
Eslint is popular Javascript and type script JSX, etc static code analysis tool can lintin. It helps developers fixed and fix problems, enforce coding standards and maintain code quality. Eslints rules are completely configurable and extensible via plugins, enabling both style and code quality checks. Common use case include identifying syntax errors, enforcing best practices and catching bugs before runtime

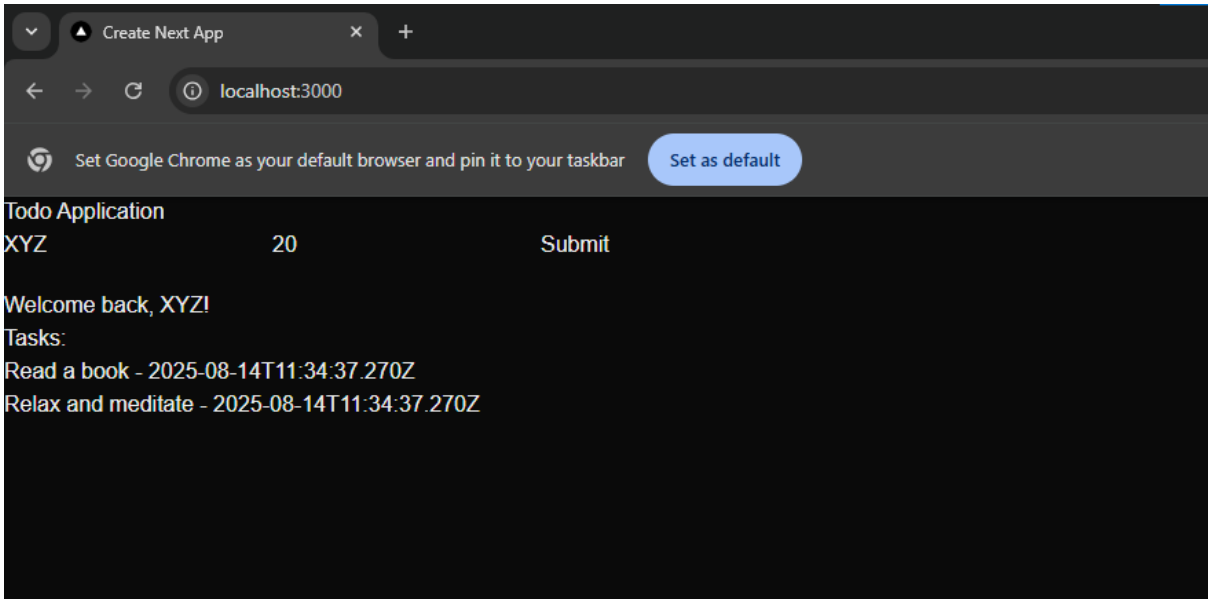
Camlin Page

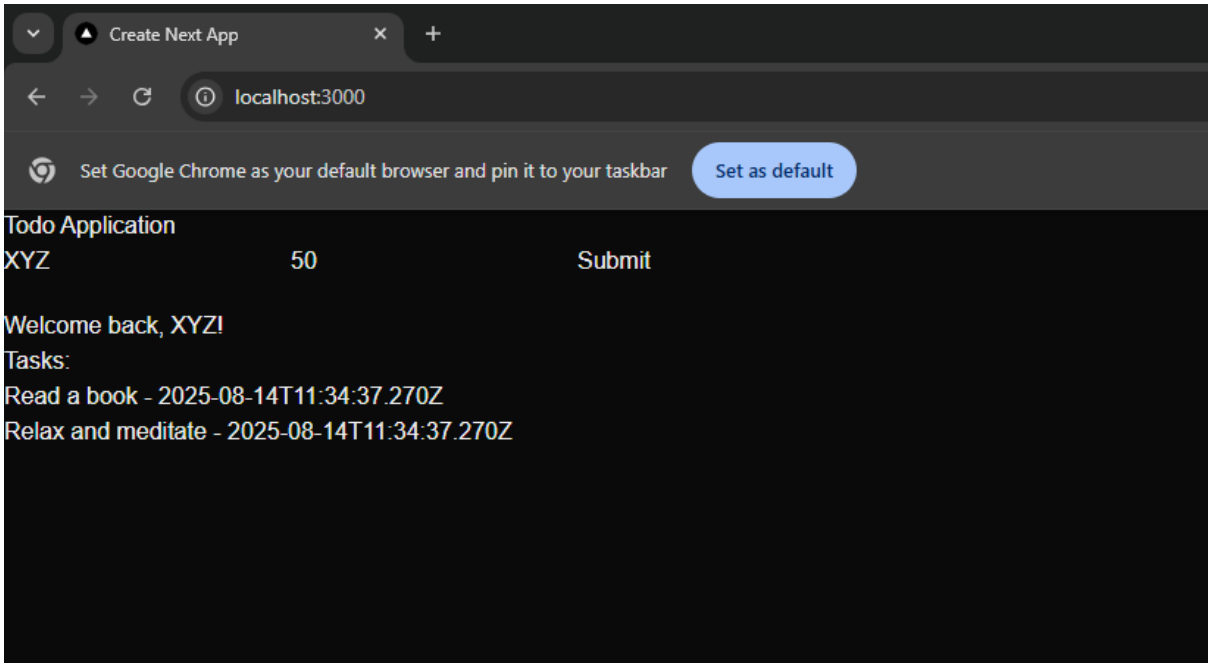
5 Prettier

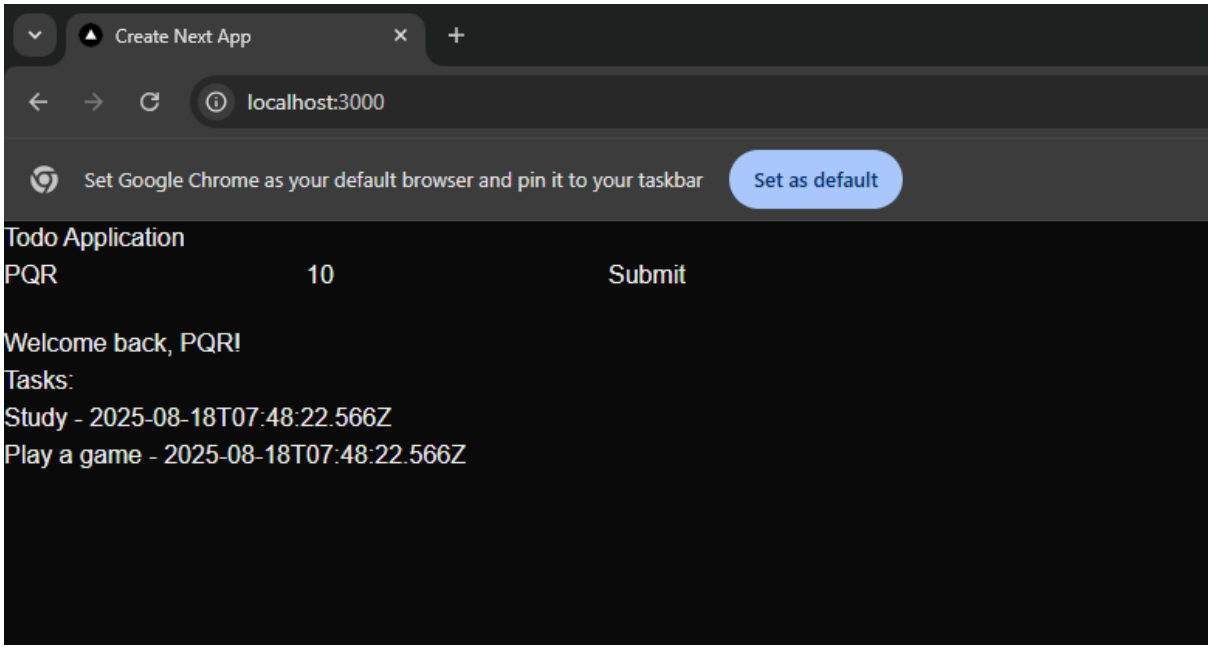
Prettier is an opinionated code formatter. Its main job is to consistently enforce a consistent coding style across entire codebase handling whitespace, semicolons, quotes and more. Unlike Eslint, prettier does not perform error checking, it simply format code to look uniform, depending upon workflows and reducing code review own kind

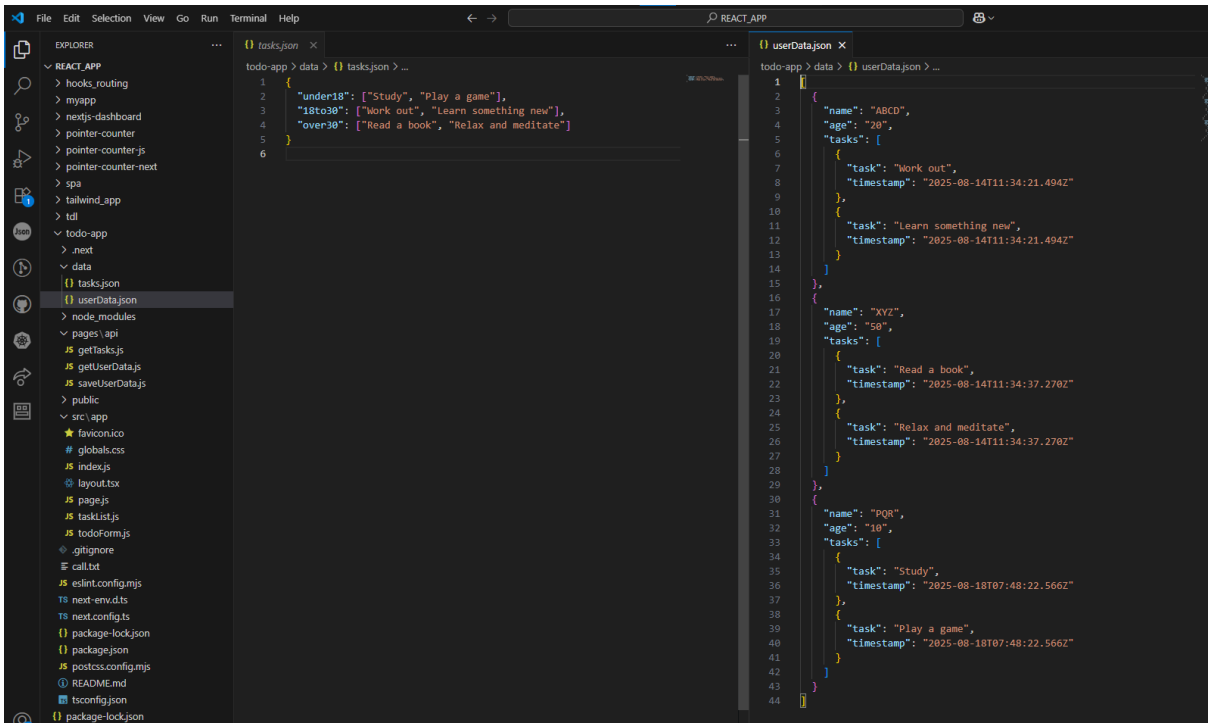

Create Next App
localhost:3000
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Todo Application
Enter your name
Enter your age
Submit
Tasks:
No tasks available. Add your name and age to get suggestions!

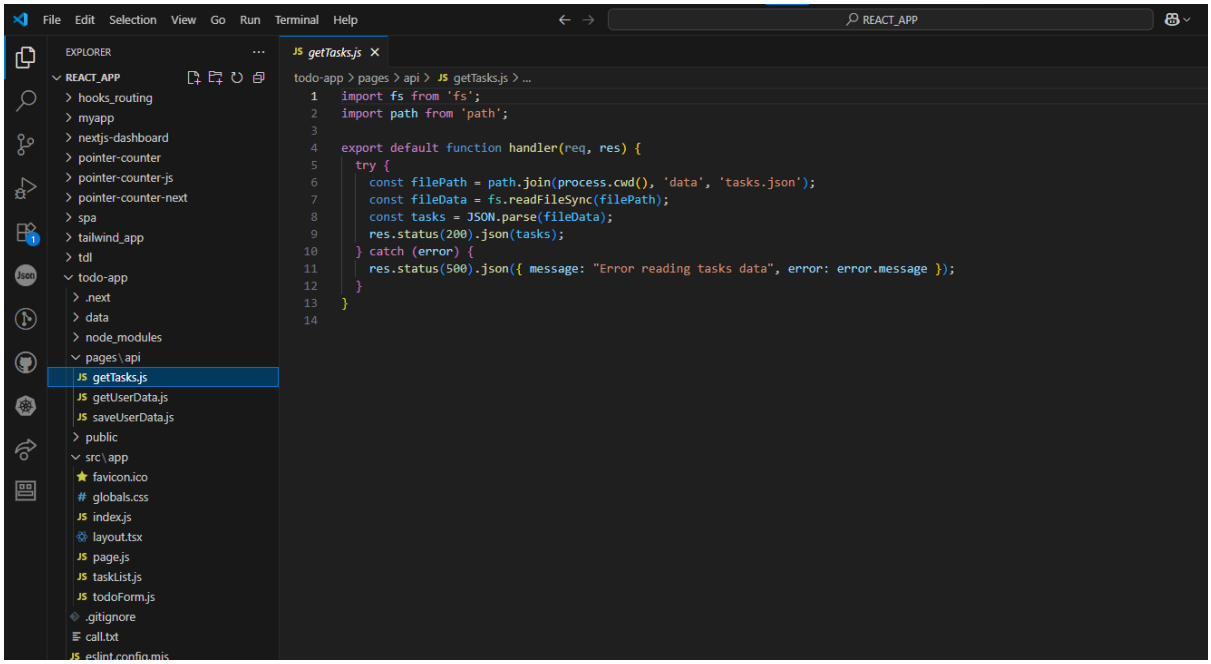

Create Next App
localhost:3000
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Todo Application
ABCD
20
Submit
Welcome back, ABCD!
Tasks:
Work out - 2025-08-14T11:34:21.494Z
Learn something new - 2025-08-14T11:34:21.494Z

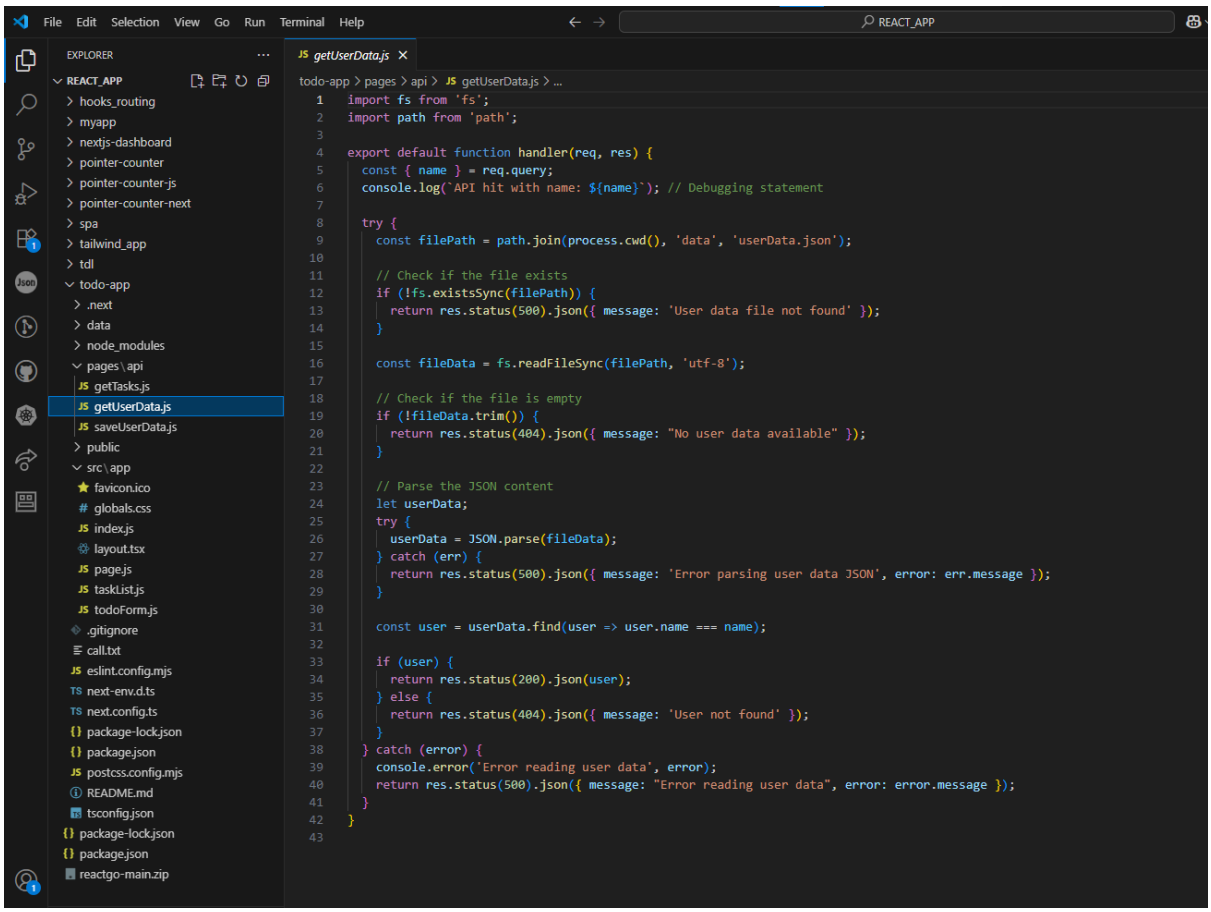

Create Next App
localhost:3000
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Todo Application
XYZ
20
Submit
Welcome back, XYZ!
Tasks:
Read a book - 2025-08-14T11:34:37.270Z
Relax and meditate - 2025-08-14T11:34:37.270Z

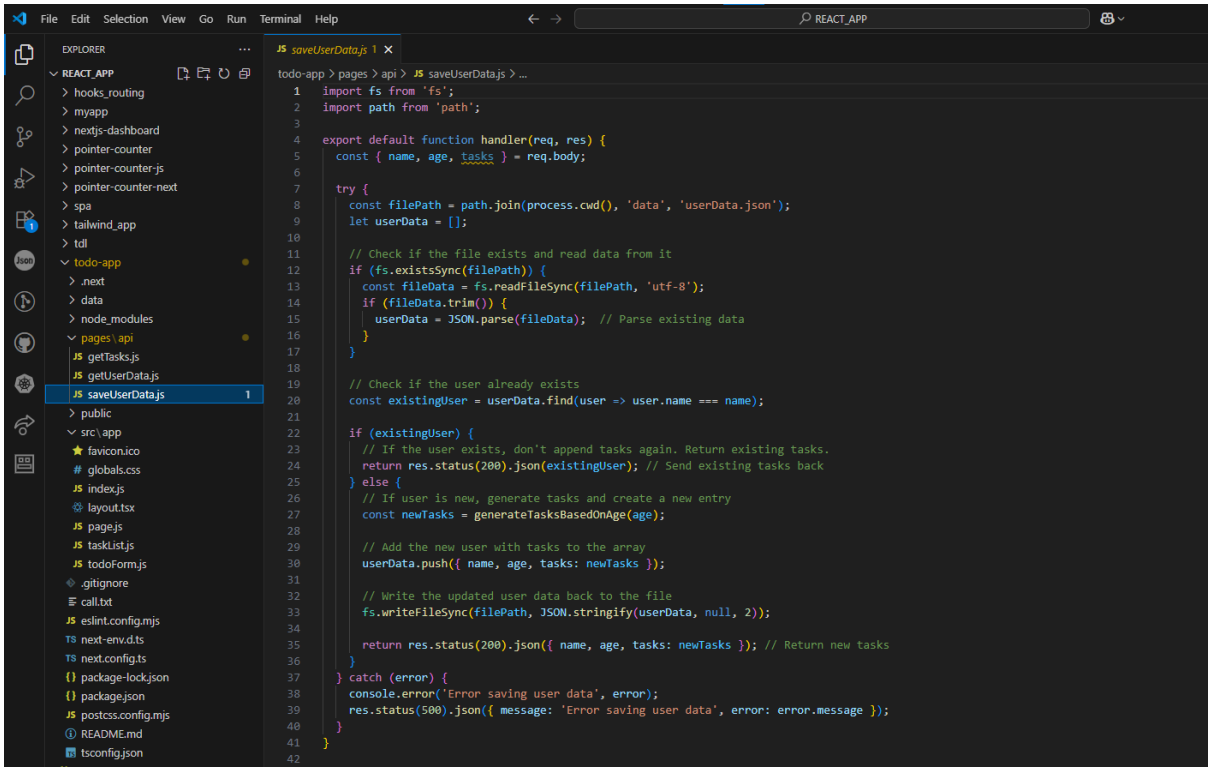

Create Next App
localhost:3000
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Todo Application
XYZ
50
Submit
Welcome back, XYZ!
Tasks:
Read a book - 2025-08-14T11:34:37.270Z
Relax and meditate - 2025-08-14T11:34:37.270Z

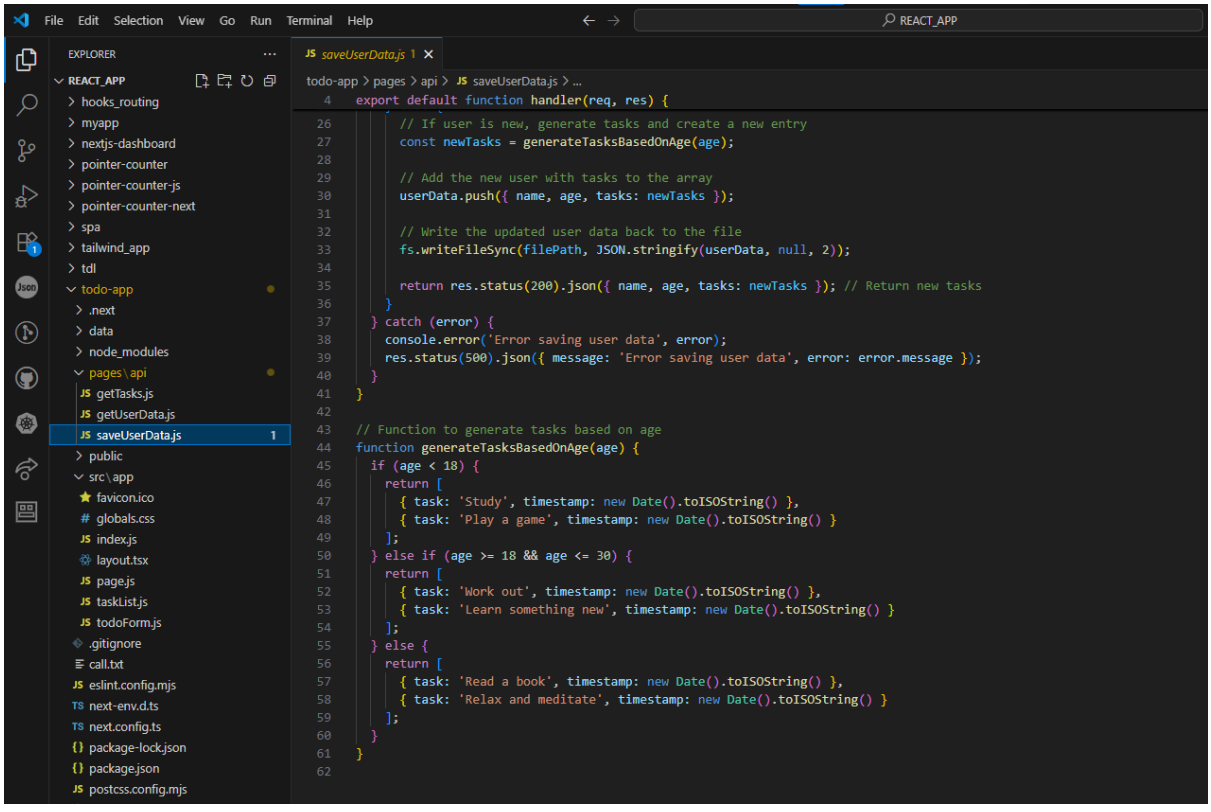

Create Next App
localhost:3000
Set Google Chrome as your default browser and pin it to your taskbar
Set as default
Todo Application
PQR
10
Submit
Welcome back, PQR!
Tasks:
Study - 2025-08-18T07:48:22.566Z
Play a game - 2025-08-18T07:48:22.566Z

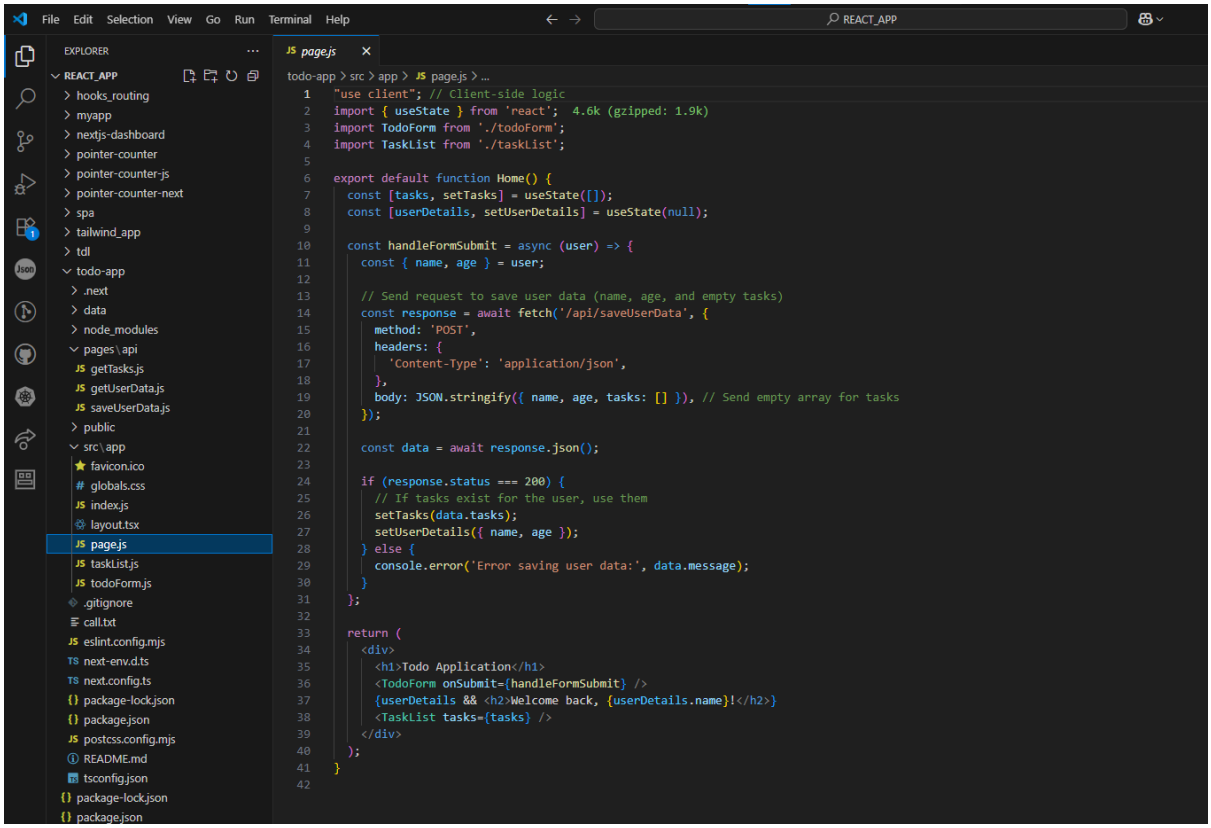

File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
tasks.json
userData.json
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
package.json
postcss.config.mjs
README.md
tsconfig.json
package-lock.json
tasks.json
todo-app > data > tasks.json > ...
{
"under18": ["Study", "Play a game"],
"18to30": ["Work out", "Learn something new"],
"over30": ["Read a book", "Relax and meditate"]
}
userData.json
todo-app > data > userData.json > ...
[
{
"name": "ABCD",
"age": "20",
"tasks": [
{
"task": "Work out",
"timestamp": "2025-08-14T11:34:21.494Z"
},
{
"task": "Learn something new",
"timestamp": "2025-08-14T11:34:21.494Z"
}
]
},
{
"name": "XYZ",
"age": "50",
"tasks": [
{
"task": "Read a book",
"timestamp": "2025-08-14T11:34:37.270Z"
},
{
"task": "Relax and meditate",
"timestamp": "2025-08-14T11:34:37.270Z"
}
]
},
{
"name": "PQR",
"age": "10",
"tasks": [
{
"task": "Study",
"timestamp": "2025-08-18T07:48:22.566Z"
},
{
"task": "Play a game",
"timestamp": "2025-08-18T07:48:22.566Z"
}
]
}
]


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
getTasks.js
todo-app > pages > api > getTasks.js > ...
import fs from 'fs';
import path from 'path';
export default function handler(req, res) {
try {
const filePath = path.join(process.cwd(), 'data', 'tasks.json');
const fileData = fs.readFileSync(filePath);
const tasks = JSON.parse(fileData);
res.status(200).json(tasks);
} catch (error) {
res.status(500).json({ message: "Error reading tasks data", error: error.message });
}
}
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
getUserData.js
todo-app > pages > api > getUserData.js > ...
import fs from 'fs';
import path from 'path';
export default function handler(req, res) {
const { name } = req.query;
console.log(`API hit with name: ${name}`); // Debugging statement
try {
const filePath = path.join(process.cwd(), 'data', 'userData.json');
// Check if the file exists
if (!fs.existsSync(filePath)) {
return res.status(500).json({ message: 'User data file not found' });
}
const fileData = fs.readFileSync(filePath, 'utf-8');
// Check if the file is empty
if (!fileData.trim()) {
return res.status(404).json({ message: "No user data available" });
}
// Parse the JSON content
let userData;
try {
userData = JSON.parse(fileData);
} catch (err) {
return res.status(500).json({ message: 'Error parsing user data JSON', error: err.message });
}
const user = userData.find(user => user.name === name);
if (user) {
return res.status(200).json(user);
} else {
return res.status(404).json({ message: 'User not found' });
}
} catch (error) {
console.error('Error reading user data', error);
return res.status(500).json({ message: "Error reading user data", error: error.message });
}
}
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
package.json
postcss.config.mjs
README.md
tsconfig.json
package-lock.json
package.json
reactgo-main.zip

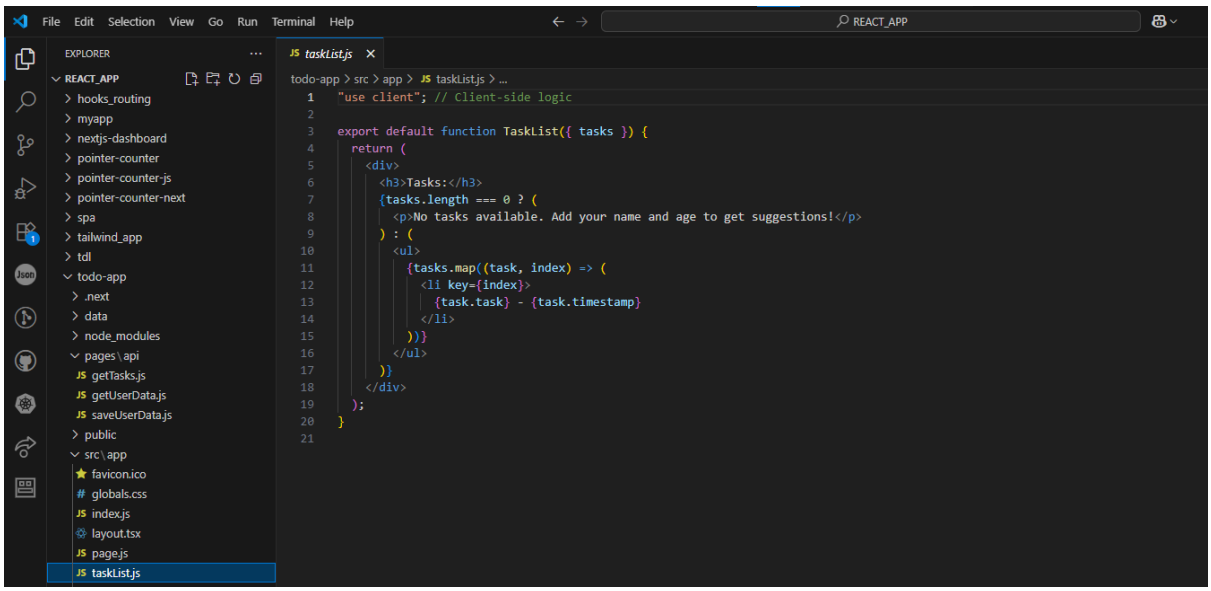

File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js 1
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
package.json
postcss.config.mjs
README.md
tsconfig.json
saveUserData.js 1
todo-app > pages > api > saveUserData.js > ...
import fs from 'fs';
import path from 'path';
export default function handler(req, res) {
const { name, age, tasks } = req.body;
try {
const filePath = path.join(process.cwd(), 'data', 'userData.json');
let userData = [];
// Check if the file exists and read data from it
if (fs.existsSync(filePath)) {
const fileData = fs.readFileSync(filePath, 'utf-8');
if (fileData.trim()) {
userData = JSON.parse(fileData); // Parse existing data
}
}
// Check if the user already exists
const existingUser = userData.find(user => user.name === name);
if (existingUser) {
// If the user exists, don't append tasks again. Return existing tasks.
return res.status(200).json(existingUser); // Send existing tasks back
} else {
// If user is new, generate tasks and create a new entry
const newTasks = generateTasksBasedOnAge(age);
// Add the new user with tasks to the array
userData.push({ name, age, tasks: newTasks });
// Write the updated user data back to the file
fs.writeFileSync(filePath, JSON.stringify(userData, null, 2));
return res.status(200).json({ name, age, tasks: newTasks }); // Return new tasks
}
} catch (error) {
console.error('Error saving user data', error);
res.status(500).json({ message: 'Error saving user data', error: error.message });
}
}


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js 1
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
package.json
postcss.config.mjs
saveUserData.js 1
todo-app > pages > api > saveUserData.js > ...
export default function handler(req, res) {
// If user is new, generate tasks and create a new entry
const newTasks = generateTasksBasedOnAge(age);
// Add the new user with tasks to the array
userData.push({ name, age, tasks: newTasks });
// Write the updated user data back to the file
fs.writeFileSync(filePath, JSON.stringify(userData, null, 2));
return res.status(200).json({ name, age, tasks: newTasks }); // Return new tasks
}
} catch (error) {
console.error('Error saving user data', error);
res.status(500).json({ message: 'Error saving user data', error: error.message });
}
}
// Function to generate tasks based on age
function generateTasksBasedOnAge(age) {
if (age < 18) {
return [
{ task: 'Study', timestamp: new Date().toISOString() },
{ task: 'Play a game', timestamp: new Date().toISOString() }
];
} else if (age >= 18 && age <= 30) {
return [
{ task: 'Work out', timestamp: new Date().toISOString() },
{ task: 'Learn something new', timestamp: new Date().toISOString() }
];
} else {
return [
{ task: 'Read a book', timestamp: new Date().toISOString() },
{ task: 'Relax and meditate', timestamp: new Date().toISOString() }
];
}
}


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
page.js
todo-app > src > app > page.js > ...
"use client"; // Client-side logic
import { useState } from 'react'; 4.6k (gzipped: 1.9k)
import TodoForm from './todoForm';
import TaskList from './taskList';
export default function Home() {
const [tasks, setTasks] = useState([]);
const [userDetails, setUserDetails] = useState(null);
const handleFormSubmit = async (user) => {
const { name, age } = user;
// Send request to save user data (name, age, and empty tasks)
const response = await fetch('/api/saveUserData', {
method: 'POST',
headers: {
'Content-Type': 'application/json',
},
body: JSON.stringify({ name, age, tasks: [] }), // Send empty array for tasks
});
const data = await response.json();
if (response.status === 200) {
// If tasks exist for the user, use them
setTasks(data.tasks);
setUserDetails({ name, age });
} else {
console.error('Error saving user data:', data.message);
}
};
return (

<h1>Todo Application</h1>
<TodoForm onSubmit={handleFormSubmit} />
{userDetails && <h2>Welcome back, {userDetails.name}!</h2>}
<TaskList tasks={tasks} />

);
}
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
package.json
postcss.config.mjs
README.md
tsconfig.json
package-lock.json
package.json


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
taskList.js
todo-app > src > app > taskList.js > ...
"use client"; // Client-side logic
export default function TaskList({ tasks }) {
return (

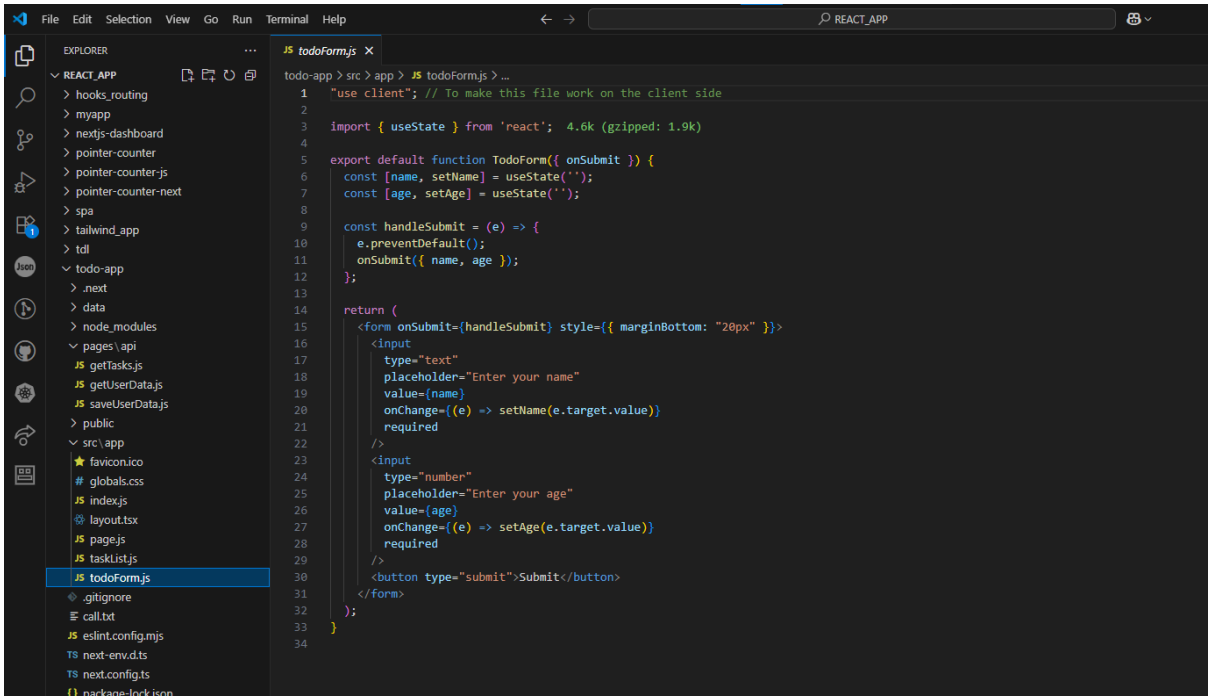
<h3>Tasks:</h3>
{tasks.length === 0 ? (
No tasks available. Add your name and age to get suggestions!
) : (

{tasks.map((task, index) => (
<li key={index}>
{task.task} - {task.timestamp}

))}

)}

);
}
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js


File Edit Selection View Go Run Terminal Help
REACT_APP
EXPLORER
REACT_APP
hooks_routing
myapp
nextjs-dashboard
pointer-counter
pointer-counter-js
pointer-counter-next
spa
tailwind_app
tdl
todo-app
.next
data
node_modules
pages\api
getTasks.js
getUserData.js
saveUserData.js
public
src\app
favicon.ico
globals.css
index.js
layout.tsx
page.js
taskList.js
todoForm.js
.gitignore
call.txt
eslint.config.mjs
next-env.d.ts
next.config.ts
package-lock.json
todoForm.js
todo-app > src > app > todoForm.js > ...
1 "use client"; // To make this file work on the client side
2
3 import { useState } from 'react'; 4.6k (gzipped: 1.9k)
4
5 export default function TodoForm({ onSubmit }) {
6 const [name, setName] = useState('');
7 const [age, setAge] = useState('');
8
9 const handleSubmit = (e) => {
10 e.preventDefault();
11 onSubmit({ name, age });
12 };
13
14 return (
15 <form onSubmit={handleSubmit} style={{ marginBottom: "20px" }}>
16 <input
17 type="text"
18 placeholder="Enter your name"
19 value={name}
20 onChange={(e) => setName(e.target.value)}
21 required
22 />
23 <input
24 type="number"
25 placeholder="Enter your age"
26 value={age}
27 onChange={(e) => setAge(e.target.value)}
28 required
29 />
30 <button type="submit">Submit</button>
31 </form>
32);
33 }
34