

Module 4

Exercise 5 (NextJs Page & Routing)

Q1.) Create a user listing page using fetch data of users using this api

<https://jsonplaceholder.typicode.com/users>

A1.) Nextjs

```
Module4 > Exercise5 > todo-app > app > users > 📄 page.jsx > 🗂️ Users
1  import Link from "next/link";
2  import './page.css'
3  export default async function Users(){
4
5      const data=await fetch("https://jsonplaceholder.typicode.com/users");
6      const users=await data.json();
7      return(
8          <div className="users">
9              <h2>Users List</h2>
10
11             <ul>
12                 {users.map(({id,name,username,email}),i)=>(
13                     <li key={i}>
14                         <h3>User Id : {id}</h3>
15                         <span>Name: {name}</span>
16                         <span>UserName: {username}</span>
17                         <span>Email: {email}</span>
18                         <Link href={`/users/${id}`}>User Details</Link>
19                     </li>
20                 ))}
21             </ul>
22         </div>
23     );
24
25 }
26 }
```

Output

The screenshot shows a web application interface with a dark theme. At the top, there is a navigation bar with three items: "Home", "Users" (which is highlighted in green), and "Todos". Below the navigation bar, the title "Users List" is centered. The main content area displays three user entries, each in its own card:

- User Id : 1**
Name: Leanne Graham
UserName: Bret
Email: Sincere@april.biz
[User Details](#)
- User Id : 2**
Name: Ervin Howell
UserName: Antonette
Email: Shanna@melissa.tv
[User Details](#)
- User Id : 3**
Name: Clementine Bauch
UserName: Samantha
Email: Nathan@yesenia.net

Q2.) Create a page which show individual user details using this api

[https://jsonplaceholder.typicode.com/users/\[id\]](https://jsonplaceholder.typicode.com/users/[id])

A2.) NextJs

```
odule4 > Exercise5 > todo-app > app > users > [userId] > 🌐 page.jsx > 📁 UserDetail
1  import Link from "next/link";
2  import "./page.css";
3
4  export default async function UserDetail({ params }) {
5
6      const { userId } = await params;
7
8      const data = await fetch(
9          `https://jsonplaceholder.typicode.com/users/${userId}`
10     );
11
12      const { id, name, username, email, address, phone, website } = await data.json();
13
14      return (
15          <div className="user-detail">
16              <div className="card">
17
18                  <div className="top">
19                      <h4>User ID: {id}</h4>
20                      <h2>{name}</h2>
21                      <p className="username">@{username}</p>
22                  </div>
23
24                  <div className="middle">
25                      <p><strong>Email:</strong> {email}</p>
26                      <p><strong>Phone:</strong> {phone}</p>
27                      <p>
28                          <strong>Address:</strong> {address.suite}, {address.street}, {ad
```

Output

User ID: 1

Leanne Graham

@Bret

Email: Sincere@april.biz

Phone: 1-770-736-8031 x56442

Address: Apt. 556, Kulas Light, Gwenborough

[Visit Website](#) [Todo of User](#)

Q3.) On user Details page show todos of the that user using this api
<https://jsonplaceholder.typicode.com/users/1/todos>

A3.) NextJs

```
Module4 > Exercise5 > todo-app > app > users > [userId] > todo > page.jsx > UserTodo
1 import Link from "next/link";
2 import './page.css'
3 export default async function UserTodo({params}){
4   const {userId} = await params
5   const data=await fetch(`https://jsonplaceholder.typicode.com/users/${userId}/todos`);
6   const todos=await data.json();
7   return(
8     <div className="todos">
9       <h2>Users todo</h2>
10      <ul>
11        {todos.map(({userId,id,title},i)=>(
12          <li key={i}>
13            <h3>User Id : {userId}</h3>
14            <span>id: {id}</span>
15            <span>title: {title}</span>
16            <Link href={`/todos/${id}`}>Todo details</Link>
17          </li>
18        )));
19      </ul>
20    </div>
21  );
22}
23
24
25 }
```

Output

The screenshot shows a web application interface with a dark blue header bar at the top. The header contains three navigation items: "Home", "Users", and "Todos". Below the header, the main content area has a title "Users todo" centered at the top. The content is presented in two separate rounded rectangular boxes, each representing a user profile.

User Id : 1

id: 1
title: delectus aut autem

[Todo details](#)

User Id : 1

id: 2
title: quis ut nam facilis et officia qui

[Todo details](#)

Q4.) Create a todos listing page using fetch data of todos using this api
<https://jsonplaceholder.typicode.com/todos>

A4.) NextJs

```
Module4 > Exercise5 > todo-app > app > todos > page.jsx > Todos
1 import Link from "next/link";
2 import "./page.css";
3
4 export default async function Todos(){
5
6     const data = await fetch("https://jsonplaceholder.typicode.com/todos");
7     const todos = await data.json();
8
9     return(
10         <div className="todos">
11             <h2>Todos</h2>
12
13             <ul>
14                 {todos.map(({id,title})=>(
15                     <li key={id}>
16                         <h3>Todo Id: {id}</h3>
17                         <p className="title">{title}</p>
18                         <Link href={`/todos/${id}}>Todo Details</Link>
19                     </li>
20                 )));
21             </ul>
22         </div>
23     );
24 }
25
```

Output

The screenshot shows a web application interface with a dark theme. At the top, there is a navigation bar with three items: "Home", "Users", and "Todos". The "Todos" item is highlighted with a green background and white text. Below the navigation bar, the word "Todos" is centered in a large, bold, white font. The main content area contains two items, each in its own rounded rectangular box:

- Todo Id: 1**
delectus aut autem
[Todo Details](#)
- Todo Id: 2**
quis ut nam facilis et officia qui
[Todo Details](#)

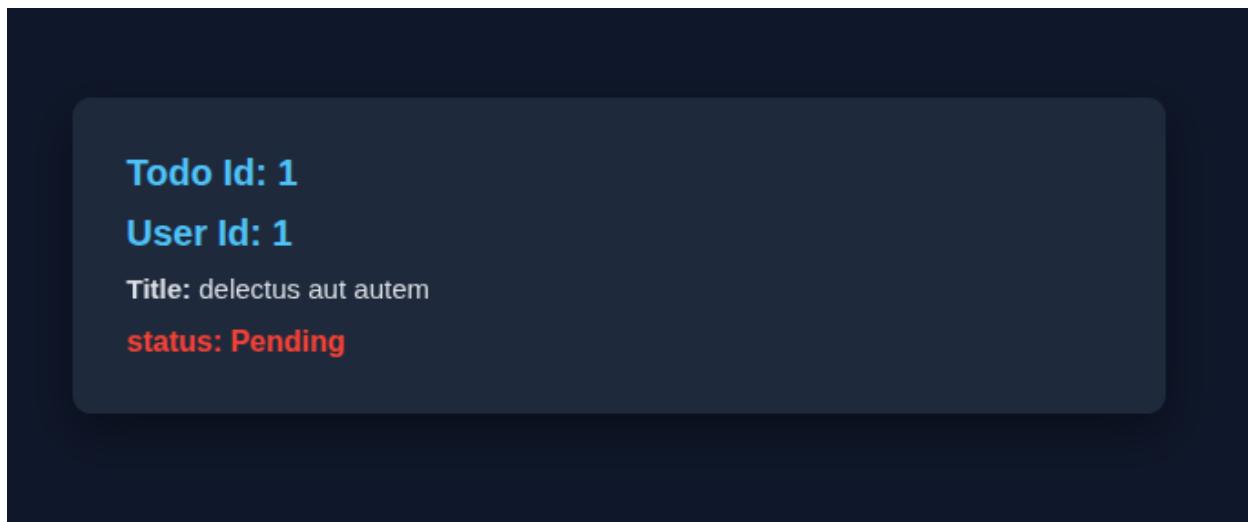
A small portion of a third item is visible at the bottom left, showing "Todo Id: 3".

Q5.) Create a page which show individual todos details using this api
<https://jsonplaceholder.typicode.com/todos>

A5.) NextJs

```
Module4 > Exercise5 > todo-app > app > todos > [todoId] > page.jsx > TodoDetails
1 import './page.css'
2
3 export default async function TodoDetails({ params }) {
4   const { todoId } = await params;
5   const data = await fetch(`https://jsonplaceholder.typicode.com/todos/${todoId}`);
6   const { id, userId, title, completed } = await data.json();
7
8   return (
9     <div className="todo-card">
10      <div className="card">
11        <h3>Todo Id: {id}</h3>
12        <h3>User Id: {userId}</h3>
13        <span><strong>Title: </strong>{title}</span>
14        <p style={{ color: completed ? "#4CAF50" : "#F44336" }}>
15          | status: {completed ? "Completed" : "Pending"}
16        </p>
17      </div>
18    </div>
19  );
20}
21
```

Output



Q6.) Create a common Header and Footer components and add these to all the pages.

A6.) NextJs

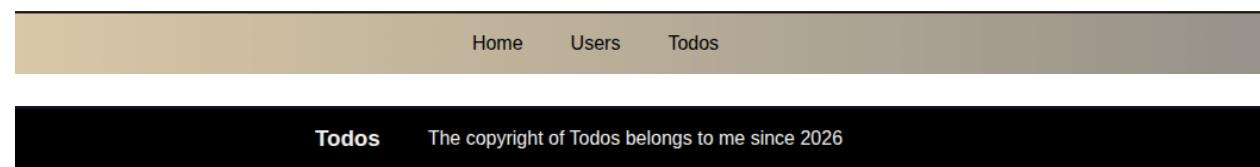
Header.JS

```
Module4 > Exercise5 > todo-app > app > components > Header.jsx > ...
1  "use client"
2  import Link from "next/link";
3  import './Header.css'
4  import { usePathname } from "next/navigation";
5  export default function Header(){
6      const pathname=usePathname();
7      return(
8          <div className="header">
9              <nav>
10                  <Link href="/" className={pathname=='/'?"link":"not-active"}>Home</Link>
11                  <Link href="/users" className={pathname=='/users'?"link":"not-active"}>Users</Link>
12                  <Link href="/todos" className={pathname=='/todos'?"link":"not-active"}>Todos</Link>
13              </nav>
14          </div>
15      );
16 }
```

Footer.JS

```
Module4 > Exercise5 > todo-app > app > components > Footer.jsx > Footer
1  import './Footer.css'
2  export default function Footer(){
3      return(
4          <div className="footer">
5              <h3>Todos</h3>
6              <span>The copyright of Todos belongs to me since 2026</span>
7          </div>
8      );
9 }
```

Output

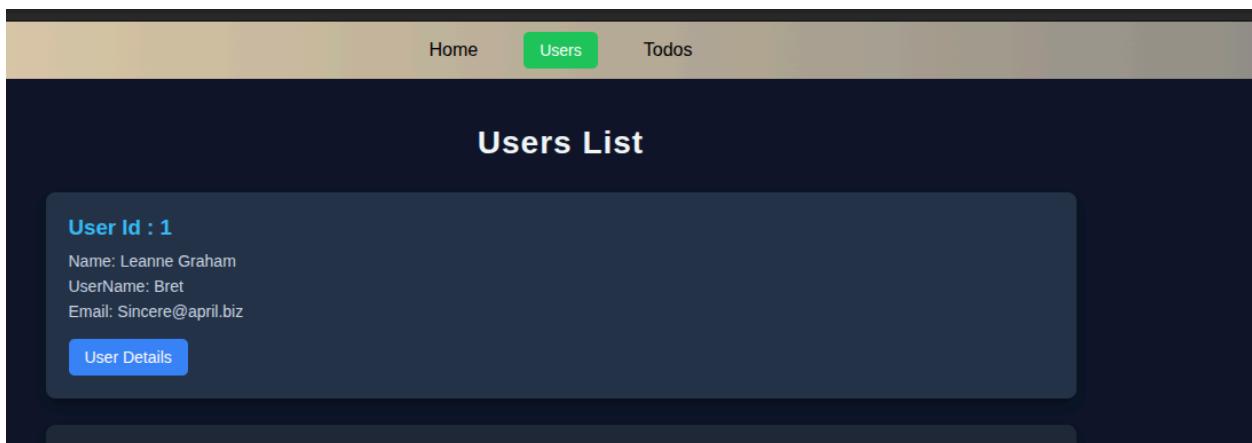


Q7.) usePathname to show active menu items.

A7.) NextJs

```
1 "use client"
2 import Link from "next/link";
3 import './Header.css'
4 import { usePathname } from "next/navigation";
5 export default function Header(){
6   const pathname=usePathname();
7   return(
8     <div className="header">
9       <nav>
10         <Link href="/" className={pathname === '/' ? "link": "not-active"}>Home</Link>
11         <Link href="/users" className={pathname === '/users' ? "link": "not-active"}>Users</Link>
12         <Link href="/todos" className={pathname === '/todos' ? "link": "not-active"}>Todos</Link>
13       </nav>
14     </div>
15   );
16 }
```

Output



Q8.) Add loading UI using [loading.js](#).

A8.) NextJs

```
export default function Loader() {
  return (
    <div style={{
      position: "fixed",
      top: 0, left: 0,
      width: "100%",
      height: "100%",
      backgroundColor: "rgba(0,0,0,0.5)",
      display: "flex",
      justifyContent: "center",
      alignItems: "center",
      color: "#fff",
      fontSize: "2rem",
      zIndex: 999
    }}>
      Loading...
    </div>
  );
}
```

Output



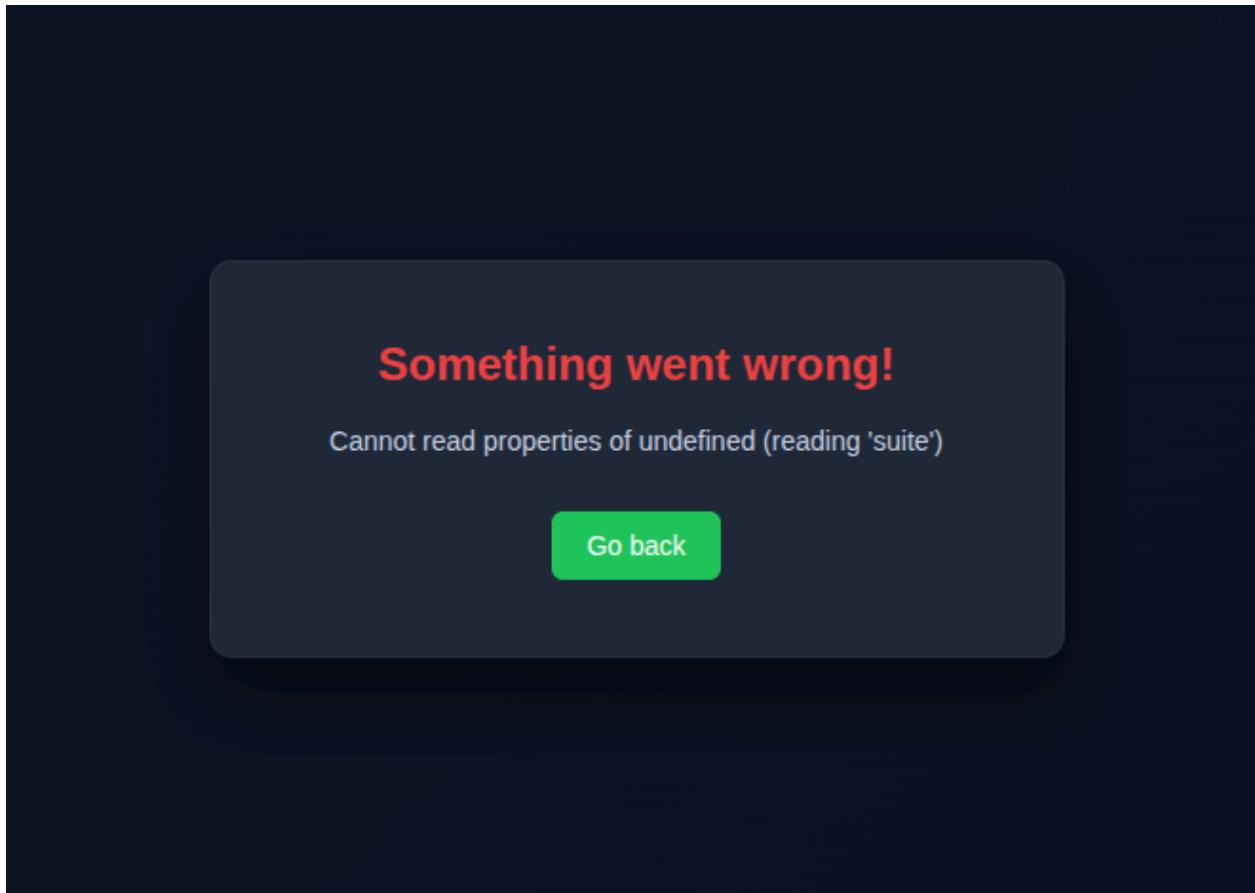
Loading...

Q9.) Add error page using [error.js](#)

A9.) NextJs

```
Module4 > Exercise5 > todo-app > app > ✎ error.jsx > ⚡ Error
1  "use client";
2  import { useRouter } from "next/navigation";
3  import "./error.css";
4  Q
5  export default function Error({ error }) {
6    const router=useRouter()
7    return (
8      <div className="error-page">
9        <div className="error-card">
10          <h2>Something went wrong!</h2>
11          <p>{error?.message}</p>
12          <button onClick={() => router.back()}>
13            Go back
14          </button>
15        </div>
16      </div>
17    );
18  }
19
```

Output



Q10.) Add not-found page if some route not exist inside the project.

A10.) NextJs

```
module4 > Exercise5 > todo-app > app > ⚡ not-found.jsx > ...
1  "use client";
2
3  import { useRouter } from "next/navigation";
4  import { useEffect, useState } from "react";
5  import "./not-found.css";
6
7  export default function Error404() {
8    const router = useRouter();
9    const [count, setCount] = useState(5);
10
11   useEffect(() => {
12     if (count === 0) {
13       router.push("/");
14       return;
15     }
16
17     const timer = setTimeout(() => {
18       setCount((prev) => prev - 1);
19     }, 1000);
20
21     return () => clearTimeout(timer);
22   }, [count, router]);
23
24   return (
25     <div className="notfound-page">
26       <div className="notfound-card">
27         <h1>404 - Page Not Found</h1>
28         <p>
29           Redirecting to homepage in{" "}
30           <span className="countdown">{count}</span> seconds...
31         </p>
32         </div>
33       </div>
34     );
35   }
36 }
```

Output

