

React App with Routing



```
import React, { useEffect, useState } from "react";
import { useParams, useNavigate } from "react-router-dom";
import axios from "axios";

const ArticleDetails = () => {
  const { id } = useParams();
  const [article, setArticle] = useState(null);

  useEffect(() => {
    axios
      .get(http://localhost:5000/articles/${id})
      .then((response) => setArticle(response.data))
      .catch((error) => console.error(error));
  }, [id]);

  const navigate = useNavigate();
  const handleNavigate = () => {
    navigate("/");
  };

  return (
    <div className="bg-slate-600 min-h-screen py-8">
      <div className="container mx-auto px-4 ">
        {article ? (
          <div className="bg-white p-6 shadow-lg rounded-lg ">
            <img
              src={article.featuredImage}
              alt={article.title}
              className="rounded-lg mb-6"
            />
            <h1 className="text-4xl font-bold mb-6 text-gray-800">
              {article.title}
            </h1>
            <p className="text-gray-700 text-lg mb-4">{article.content}</p>
            <p className="text-gray-500 text-sm">
              Reported by: {article.reporterName}
            </p>
            <button
              onClick={handleNavigate}
              class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-6 rounded-md mt-5"
            >
              Back
            </button>
          </div>
        ) : (
          <div>
            <h1>Article Details</h1>
            <p>No article found for this ID.</p>
          </div>
        )}
      </div>
    </div>
  );
};
```

```

    </div>
  ) : (
    <p className="text-center text-gray-500">Loading article...</p>
  )}
</div>
</div>
);
};

export default ArticleDetails;

import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";

const Home = () => {
  const [articles, setArticles] = useState([]);

  useEffect(() => {
    axios
      .get("http://localhost:5000/articles")
      .then((response) => setArticles(response.data))
      .catch((error) => console.error(error));
  }, []);

  return (
    <div className="bg-zinc-600 min-h-screen py-8">
      <div className="container mx-auto px-4">
        <h1 className="text-4xl font-bold text-center mb-8 text-white">
          Articles
        </h1>
        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
          {articles.map((article) => (
            <div
              key={article.id}
              className="bg-white p-6 shadow-lg rounded-lg"
            >
              <div className="flex justify-center items-center">
                <img
                  src={article.featuredImage}
                  alt={article.title}
                  className="rounded-lg mb-4"
                />
              </div>
              <div>
                <h2 className="text-2xl font-semibold mb-2 text-gray-700">
                  {article.title}
                </h2>
                <p className="text-gray-600 text-sm mb-4">{article.summary}</p>
                <p className="text-gray-500 text-xs">By {article.reporterName}</p>
              </div>
            </div>
          ))}
        </div>
      </div>
    </div>
  );
};

```

```

    <Link
      to={"/articles/${article.id}}
      className="text-blue-500 hover:text-blue-700 font-medium mt-4 inline-block"
    >
      Read More →
    </Link>
  </div>
  )}
</div>
</div>
</div>
);
};

export default Home;

import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "../components/Home";
import ArticleDetails from "../components/ArticleDetails";

const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/articles/:id" element={<ArticleDetails />} />
      </Routes>
    </Router>
  );
};

export default App;

explain nicely

```



This code demonstrates a simple React application with routing using React Router. The app consists of two primary components: **Home** and **ArticleDetails**. Let's break it down:

App Component

The `App` component acts as the entry point of the application and sets up routing using `react-router-dom`.

Code Breakdown:

- **Router:** Wraps the application to enable routing.

- **Routes:** Defines routes for the application.
 - ``/`` - Renders the **Home** component.
 - ``/articles/:id`` - Renders the **ArticleDetails** component.

Home Component

The **Home** component displays a list of articles fetched from an API.

Key Features:

1. **State Management:**
 - ``articles`` is a state variable to store the fetched articles.
2. **API Call:**
 - Fetches articles from ``http://localhost:5000/articles`` using ``axios`` when the component mounts.
3. **Rendering Articles:**
 - Each article is displayed using a styled card.
 - A ``Link`` directs the user to the detailed page of the article using ``/articles/:id``.

Highlights:

- Articles are displayed in a grid layout using TailwindCSS classes.
- Each card shows the article's image, title, summary, and reporter name.
- Clicking "Read More" navigates to the **ArticleDetails** page for the selected article.

ArticleDetails Component

The **ArticleDetails** component displays detailed information about a single article.

Key Features:

1. **Dynamic Route Handling:**
 - Uses ``useParams`` to extract the article ID (``:id``) from the route.
2. **API Call:**
 - Fetches article details from ``http://localhost:5000/articles/${id}`` when the component mounts or when the ``id`` changes.
3. **Back Navigation:**
 - Includes a "Back" button to navigate to the homepage using ``useNavigate``.

Highlights:

- Displays the article's image, title, content, and reporter name in a well-structured card.
- Shows a loading message while the article data is being fetched.

Technical Features

1. **React Router:**
 - Provides navigation between pages without reloading the browser.
2. **Axios:**
 - Handles API requests to fetch data from a backend server.
3. **Dynamic Routes:**
 - Enables navigation to individual article pages with a unique ID.

4. TailwindCSS:

- Used for styling to ensure a clean and responsive layout.

Usage Flow

1. Homepage:

- Users land on the homepage (`/`), where all articles are listed.

2. View Details:

- Clicking "Read More" navigates to `/articles/:id` to view the article details.

3. Back Navigation:

- From the article details page, users can return to the homepage using the "Back" button.

This architecture is ideal for building scalable, user-friendly, and dynamic content-driven applications.



```
import React, { useEffect, useState } from "react";
import { useParams, useNavigate } from "react-router-dom";
import axios from "axios";

const ArticleDetails = () => {
  const { id } = useParams();
  const [article, setArticle] = useState(null);

  useEffect(() => {
    axios
      .get(`http://localhost:5000/articles/${id}`)
      .then((response) => setArticle(response.data))
      .catch((error) => console.error(error));
  }, [id]);

  const navigate = useNavigate();
  const handleNavigate = () => {
    navigate("/");
  };

  return (
    <div className="bg-slate-600 min-h-screen py-8">
      <div className="container mx-auto px-4 ">
        {article ? (
          <div className="bg-white p-6 shadow-lg rounded-lg ">
            <img
              src={article.featuredImage}
              alt={article.title}
              className="rounded-lg mb-6"
            />
            <h1 className="text-4xl font-bold mb-6 text-gray-800">
              {article.title}
            </h1>

```

```

    <p className="text-gray-700 text-lg mb-4">{article.content}</p>
    <p className="text-gray-500 text-sm">
      Reported by: {article.reporterName}
    </p>
    <button
      onClick={handleNavigate}
      class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-6 rounded-md mt-5"
    >
      Back
    </button>
  </div>
): (
  <p className="text-center text-gray-500">Loading article...</p>
)
</div>
</div>
);
};

```

```
export default ArticleDetails;
```

```

import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";

```

```

const Home = () => {
  const [articles, setArticles] = useState([]);

```

```

  useEffect(() => {
    axios
      .get("http://localhost:5000/articles")
      .then((response) => setArticles(response.data))
      .catch((error) => console.error(error));
  }, []);

```

```

  return (
    <div className="bg-zinc-600 min-h-screen py-8">
      <div className="container mx-auto px-4">
        <h1 className="text-4xl font-bold text-center mb-8 text-white">
          Articles
        </h1>
        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
          {articles.map((article) => (
            <div
              key={article.id}
              className="bg-white p-6 shadow-lg rounded-lg"
            >
              <div className="flex justify-center items-center">
                <img
                  src={article.featuredImage}

```

```

        alt={article.title}
        className="rounded-lg mb-4"
      />
    </div>

    <h2 className="text-2xl font-semibold mb-2 text-gray-700">
      {article.title}
    </h2>
    <p className="text-gray-600 text-sm mb-4">{article.summary}</p>
    <p className="text-gray-500 text-xs">By {article.reporterName}</p>
    <Link
      to={"/articles/${article.id}}
      className="text-blue-500 hover:text-blue-700 font-medium mt-4 inline-block"
    >
      Read More →
    </Link>
  </div>
  )}
</div>
</div>
</div>
);
};

```

```
export default Home;
```

```

import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "./components/Home";
import ArticleDetails from "./components/ArticleDetails";

```

```

const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/articles/:id" element={<ArticleDetails />} />
      </Routes>
    </Router>
  );
};

```

```
export default App;
```

explain nicely
with code line by line



Let's break down the given code line by line to explain it clearly.

App Component

This is the main entry point of the application, where routing is set up.

jsx

```
import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "../components/Home";
import ArticleDetails from "../components/ArticleDetails";
```

- `import` statements: Import React, React Router modules, and two components (**Home** and **ArticleDetails**) for routing.

jsx

```
const App = () => {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/articles/:id" element={<ArticleDetails />} />
      </Routes>
    </Router>
  );
};
```

- `<Router>`: Wraps the application to enable client-side routing.
- `<Routes>`: Contains individual `<Route>` elements to define the paths.
 - `/` renders the **Home** component.
 - `/articles/:id` renders the **ArticleDetails** component dynamically for a specific article ID.

jsx

```
export default App;
```

- Exports the `App` component for rendering in `index.js`.
-

Home Component

This component displays a list of articles fetched from an API.

jsx

```
import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import axios from "axios";
```


- `useEffect`: To fetch articles when the component mounts.
- `useState`: To manage the articles data state.
- `axios`: For making HTTP requests.
- `Link`: Provides navigation without reloading the page.

jsx

```
const Home = () => {
  const [articles, setArticles] = useState([]);
```

- Initializes `articles` as an empty array to store fetched data.

jsx

```
useEffect(() => {
  axios
    .get("http://localhost:5000/articles")
    .then((response) => setArticles(response.data))
    .catch((error) => console.error(error));
}, []);
```

- Fetches articles from `http://localhost:5000/articles` using `axios` and stores them in `articles`.
- The empty dependency array (`[]`) ensures this runs only on the initial render.

jsx

```
return (
  <div className="bg-zinc-600 min-h-screen py-8">
    <div className="container mx-auto px-4">
      <h1 className="text-4xl font-bold text-center mb-8 text-white">
        Articles
      </h1>
    </div>
  </div>
```

- Styles the page with TailwindCSS. Displays a title "Articles" in a styled header.

jsx

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
  {articles.map((article) => (
    <div key={article.id} className="bg-white p-6 shadow-lg rounded-lg">
```

- Uses a `map` function to render each article in a grid layout. Each article is styled with TailwindCSS.

jsx

```
<div className="flex justify-center items-center">
  <img
    src={article.featuredImage}
    alt={article.title}
    className="rounded-lg mb-4"
```

```

    />
  </div>

```

- Displays the article's featured image.

```
jsx
```

```

<h2 className="text-2xl font-semibold mb-2 text-gray-700">
  {article.title}
</h2>
<p className="text-gray-600 text-sm mb-4">{article.summary}</p>
<p className="text-gray-500 text-xs">By {article.reporterName}</p>

```

- Shows the article's title, summary, and reporter's name.

```
jsx
```

```

<Link
  to={` /articles/${article.id}`}
  className="text-blue-500 hover:text-blue-700 font-medium mt-4 inline-block"
>
  Read More →
</Link>

```

- `` navigates to `/articles/:id` for the selected article.
- Corrected the syntax for `to`: `\${}`.

```
jsx
```

```

    </div>
  )}
</div>
</div>
);
};

```

- Renders all articles in a grid layout.

```
jsx
```

```
export default Home;
```

- Exports the Home component.

ArticleDetails Component

Displays detailed information about a specific article.

jsx

```
import React, { useEffect, useState } from "react";
import { useParams, useNavigate } from "react-router-dom";
import axios from "axios";
```

- `useParams`: To extract the article ID (`:id`) from the route.
- `useNavigate`: To navigate back to the home page.
- `axios`: For fetching article details.

jsx

```
const ArticleDetails = () => {
  const { id } = useParams();
  const [article, setArticle] = useState(null);
```

- Extracts `id` from the URL and initializes `article` state as `null`.

jsx

```
useEffect(() => {
  axios
    .get(`http://localhost:5000/articles/${id}`)
    .then((response) => setArticle(response.data))
    .catch((error) => console.error(error));
}, [id]);
```

- Fetches article details from `http://localhost:5000/articles/${id}` whenever the `id` changes.

jsx

```
const navigate = useNavigate();
const handleNavigate = () => {
  navigate("/");
};
```

- Initializes `navigate` and defines a function `handleNavigate` to go back to the homepage.

jsx

```
return (
  <div className="bg-slate-600 min-h-screen py-8">
    <div className="container mx-auto px-4 ">
      {article ? (
        <div className="bg-white p-6 shadow-lg rounded-lg ">
```

- Displays a styled card for the article or a loading message.

```
jsx
```

```

    <img
      src={article.featuredImage}
      alt={article.title}
      className="rounded-lg mb-6"
    />
    <h1 className="text-4xl font-bold mb-6 text-gray-800">
      {article.title}
    </h1>
    <p className="text-gray-700 text-lg mb-4">{article.content}</p>
    <p className="text-gray-500 text-sm">
      Reported by: {article.reporterName}
    </p>

```

- Displays the article's image, title, content, and reporter's name.

```
jsx
```

```

    <button
      onClick={handleNavigate}
      className="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-6 rounded"
    >
      Back
    </button>
  </div>
) : (
  <p className="text-center text-gray-500">Loading article...</p>
)
}

```

- Shows a "Back" button to navigate home, and a loading message if data isn't fetched yet.

```
jsx
```

```

    </div>
  </div>
);
};

export default ArticleDetails;

```

- Exports the `ArticleDetails` component.

This code effectively demonstrates routing, API interaction, and dynamic content rendering in a React app.