

NEXT.JS

The React Framework for the Web



React is relatively **un-opinionated** about how you build and structure your applications.

There are **multiple ways** to build applications with React.

Next.js provides a framework to **structure your application**.

And **optimizations** that help make both the development process and final application **faster**.

Next.js included a **complete journey** to develop full stack web application.

NEXT.JS

Installation &
Project Structure



NEXT.JS

Installation &
Project Structure

npx create-next-app

npm run dev

npm run build

MY-APP

- app
- api\hello
- JS route.js
- ★ favicon.ico
- # globals.css
- JS layout.js
- JS page.js
- # page.module.css
- > node_modules
- public
- next.svg
- thirteen.svg
- vercel.svg
- .gitignore
- {} jsconfig.json
- JS next.config.js
- {} package-lock.json
- {} package.json
- ⓘ README.md

NEXT.JS

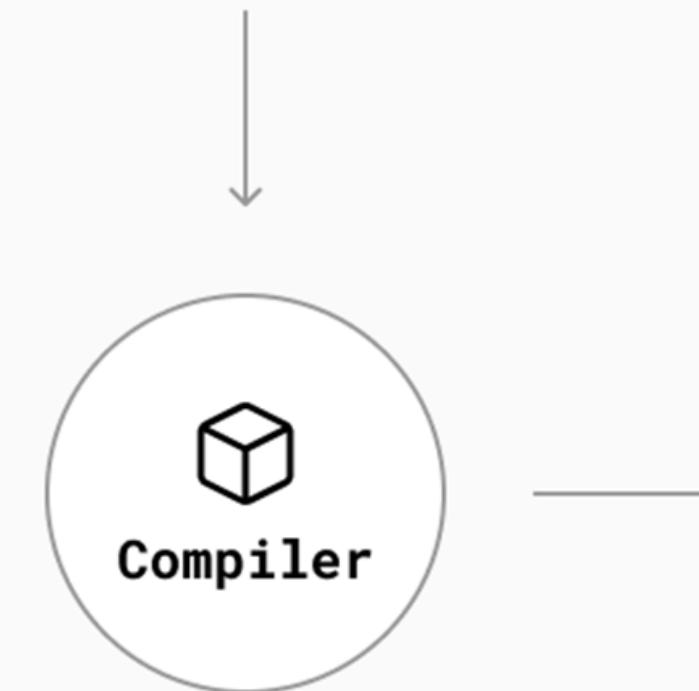
Developer Code & Compiled Code

Developer code

```
export default function HomePage() {  
  return <div>DX of the Future</div>  
}
```

Compiled code

```
"use strict";  
  
Object.defineProperty(exports,  
  "__esModule", {  
    value: true  
});  
exports.default = HomePage;  
  
function HomePage() {  
  return /*#__PURE__*/React.createElement(  
    "div", null, "DX of the Future"  
);  
}
```



NEXT.JS

Compiled Code & Minified Code

Compiled code

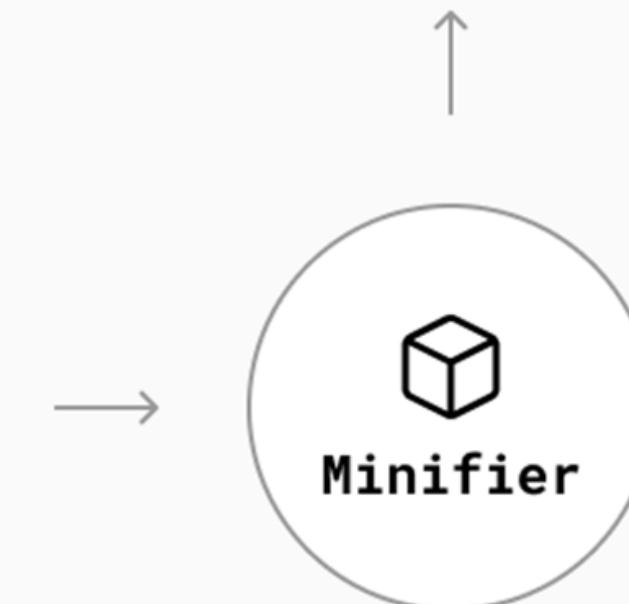
```
"use strict";

Object.defineProperty(exports,
 "__esModule", {
 value: true
});
exports.default = HomePage;

function HomePage() {
 return /*#__PURE__*/React.createElement(
 "div", null, "DX of the Future"
 );
}
```

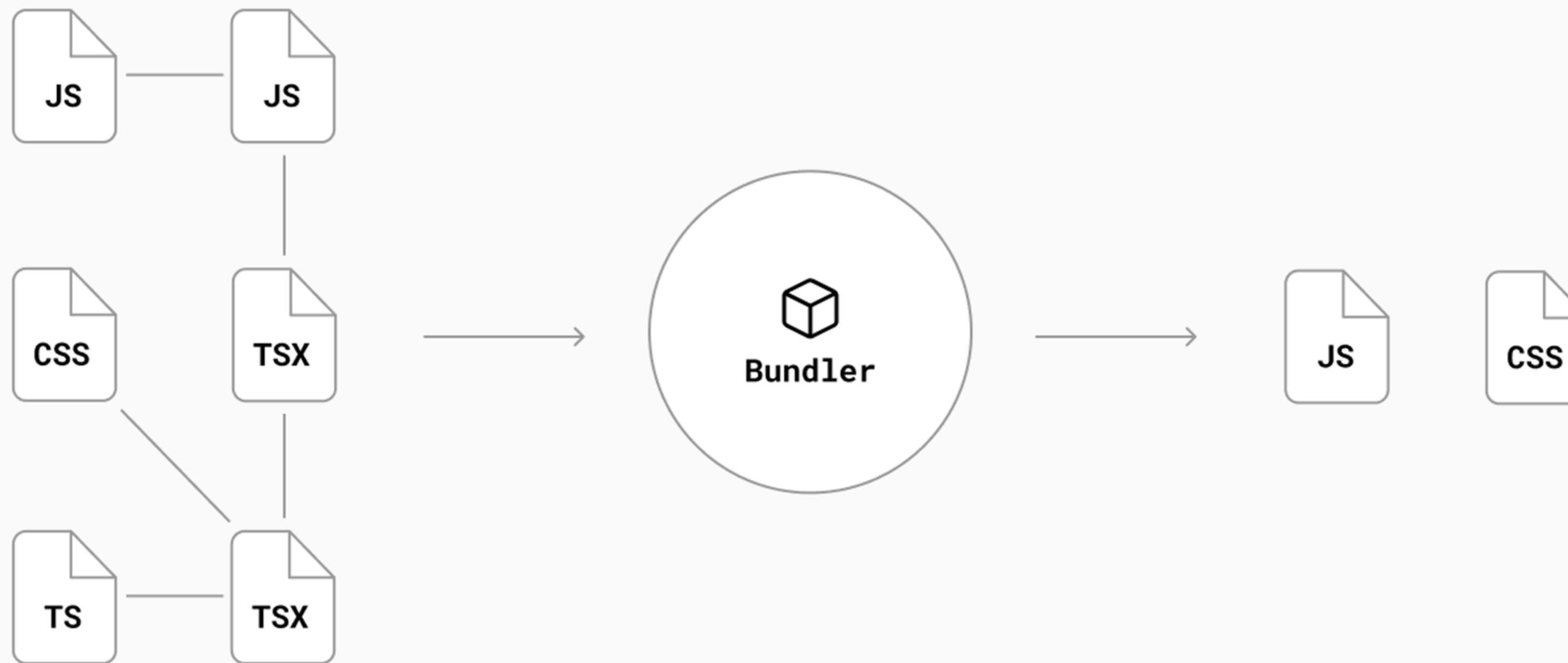
Minified code

```
"use strict";Object.defineProperty(exports,"__esModule",{value:0});exports.default=HomePage;function HomePage(){return React.createElement("div",null,"DX of the...")}
```



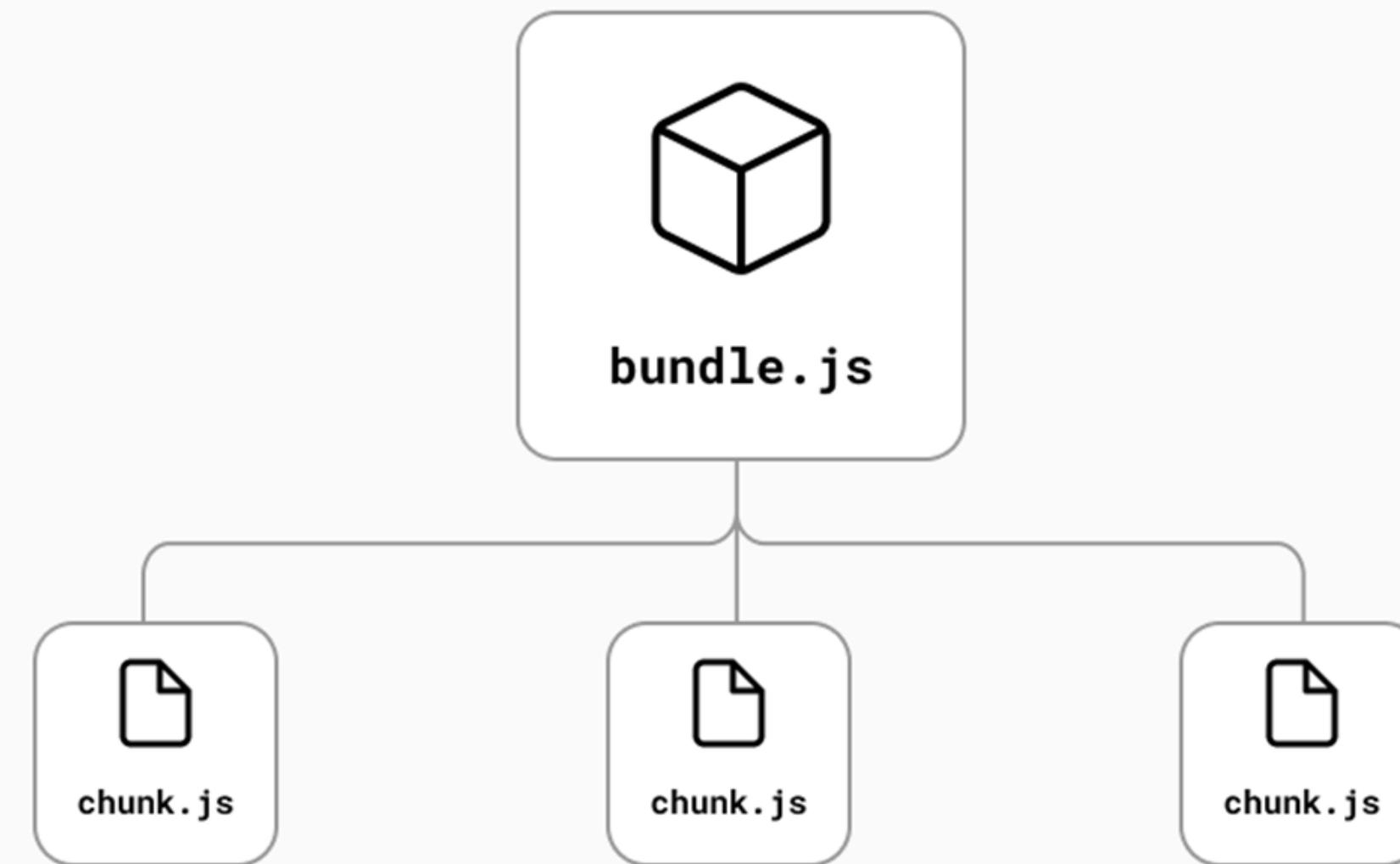
NEXT.JS

Bundler Concepts



NEXT.JS

Bundle Splitting



Functional Component In Next JS

- Components are independent and reusable bits of code.
- They serve the same purpose as JavaScript functions, but work in isolation and return HTML.
- Components come in two types, Class components and Function components
- In this tutorial we will concentrate on Function components.
- In older React code bases, you may find Class components primarily used.
- It is now suggested to use Function components along with Hooks.
- When creating a React component, the component's name MUST start with an upper case letter.

● ● ● page.js

```
1 import React from 'react';
2
3 const Page = () => {
4     return (
5         <div>
6             <h1>This is functional component</h1>
7         </div>
8     );
9 };
10
11 export default Page;
```

NEXT.JS

Serving Static File From Public Directory



page.js

```
1 import React from 'react';
2 const Page = () => {
3   return (
4     <div>
5       <img src={"assets/vercel.svg"} alt="img"/>
6     </div>
7   );
8 };
9 export default Page;
```

NEXT.JS

Global Style & Module Style

● ● ● page.js

```
1 import React from 'react';
2 import styles from './page.module.css'
3 const Page = () => {
4     return (
5         <div className="global-css">
6             <h1 className={styles.moduleCss}>This is functional component</h1>
7         </div>
8     );
9 };
10 export default Page;
```

● ● ● globals.css

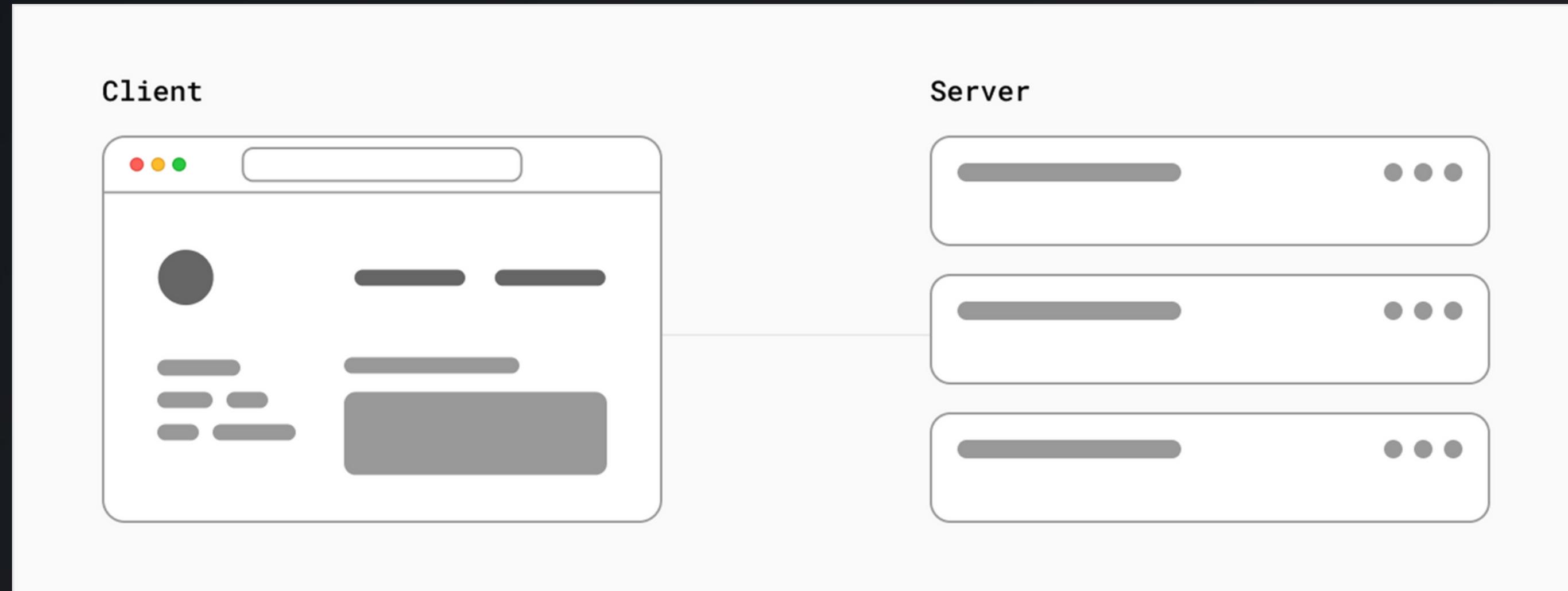
```
1 .global-css{
2     background: lightblue;
3 }
```

● ● ● page.module.css

```
1 .moduleCss{
2     color: red;
3 }
```

NEXT.JS

Client-Server Concepts



```
1  'use client'
2  import React, {useEffect, useState} from 'react';
3  const Page = () => {
4
5    let [data, setData] = useState([])
6
7    useEffect(()=>{
8      (async ()=>{
9        let response= await fetch("https://dummyjson.com/products")
10       let json = await response.json()
11       setData(json['products'])
12     })()
13   },[])
14
15
16  return (
17    <div>
18      {
19        data.map((item, index)=>{
20          return(
21            <div key={index}>
22              <h1>{item['title']}</h1>
23              <p>{item['price']}
24            </div>
25          )
26        })
27      }
28    </div>
29  );
30}
31
32 export default Page;
```

NEXT.JS

Client Side Rendering

Where the rendering of web content **occurs on the client's browser** using JavaScript

- Rendering on the client using JavaScript.
- Faster and interactive user experience.
- Data fetched dynamically on the client side.

```
2  async function getData() {
3      const data = await fetch('https://dummyjson.com/products')
4      const json= await data.json()
5      return json['products']
6  }
7
8  const Page =async () => {
9      const data = await getData()
10
11     return (
12         <div>
13             {
14                 data.map((item, index)=>{
15                     return(
16                         <div key={index}>
17                             <h1>{item['title']}</h1>
18                             <p>{item['price']}
19                         </div>
20                     )
21                 })
22             }
23         </div>
24     );
25 };
26
27 export default Page;
```

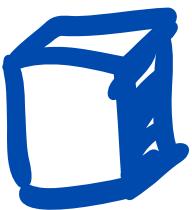
NEXT.JS

Server Side Rendering

Where web content is **generated on the server** and sent to the client's browser as fully rendered HTML

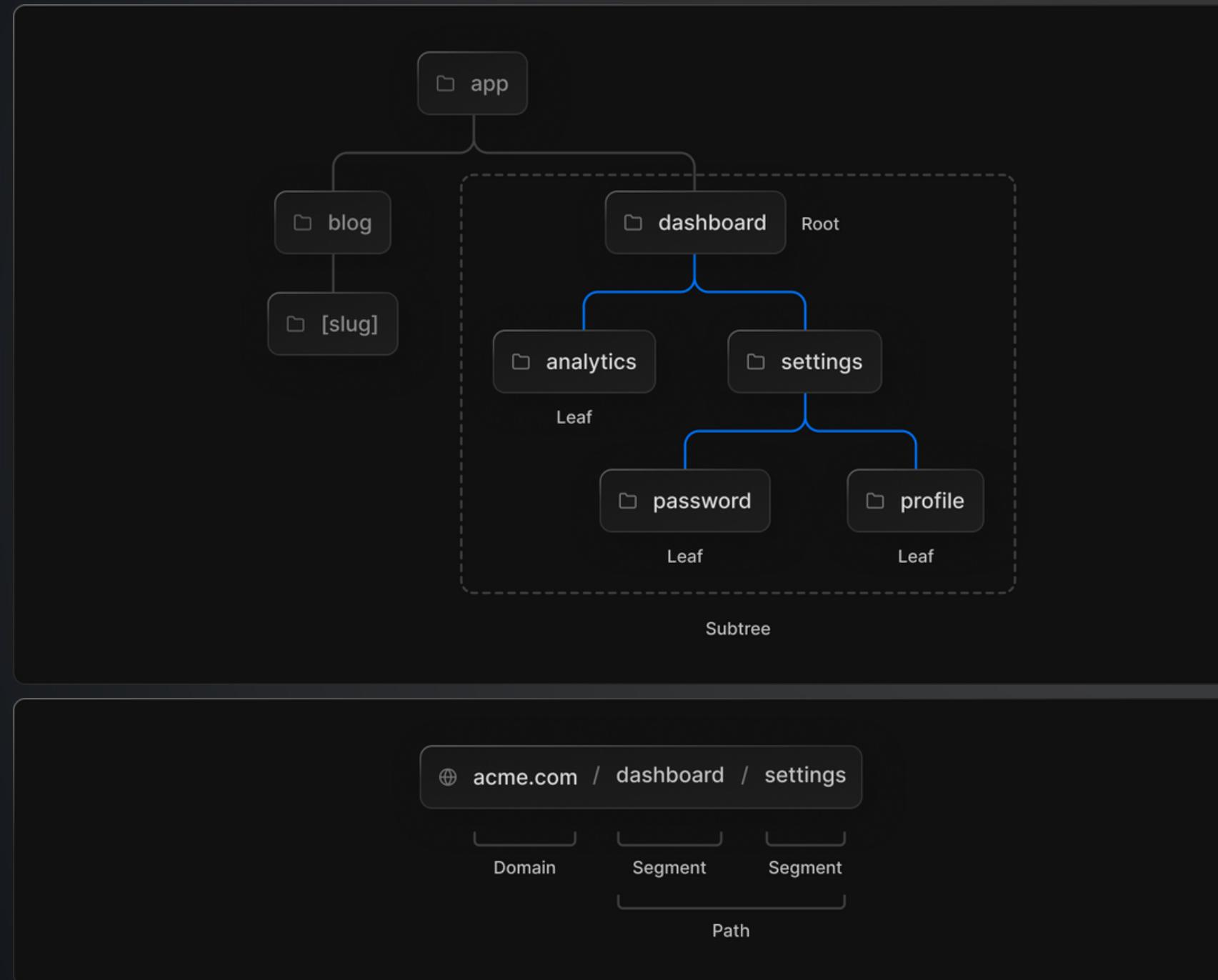
- Server generates fully rendered HTML.
- Increased server load during rendering.
- Better SEO due to crawlable content.

What do you need to do?	Server Component	Client Component
Fetch data.	✓	✗
Access backend resources (directly)	✓	✗
Keep sensitive information on the server (access tokens, API keys, etc)	✓	✗
Keep large dependencies on the server / Reduce client-side JavaScript	✓	✗
Add interactivity and event listeners (<code>onClick()</code> , <code>onChange()</code> , etc)	✗	✓
Use State and Lifecycle Effects (<code>useState()</code> , <code>useReducer()</code> , <code>useEffect()</code> , etc)	✗	✓
Use browser-only APIs	✗	✓
Use custom hooks that depend on state, effects, or browser-only APIs	✗	✓



NEXT.JS

Routing Terminology



TREE

A convention for visualizing a hierarchical structure

SUBTREE

Part of a tree, starting at a new root (first) and ending at the leaves (last)

ROOT

The first node in a tree or subtree, such as a root layout.

LEAF

Nodes in a subtree that have no children, such as the last segment in a URL path

URL SEGMENT

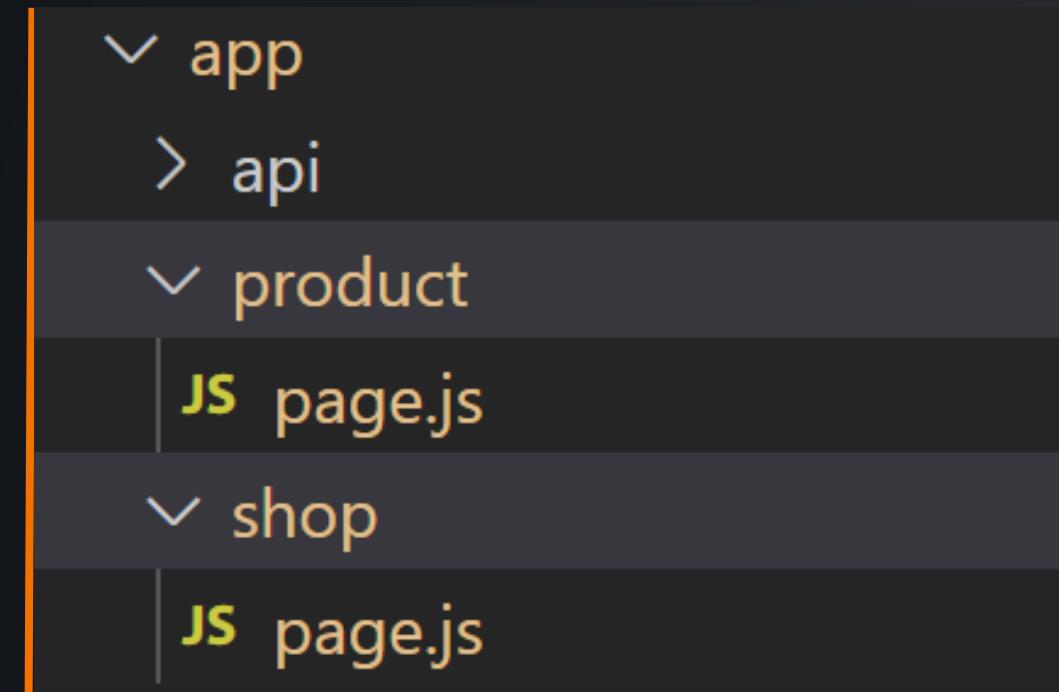
Part of the URL path delimited by slashes.

URL PATH

Part of the URL that comes after the domain

NEXT.JS

Basic Routing Example



● ● ● page.js

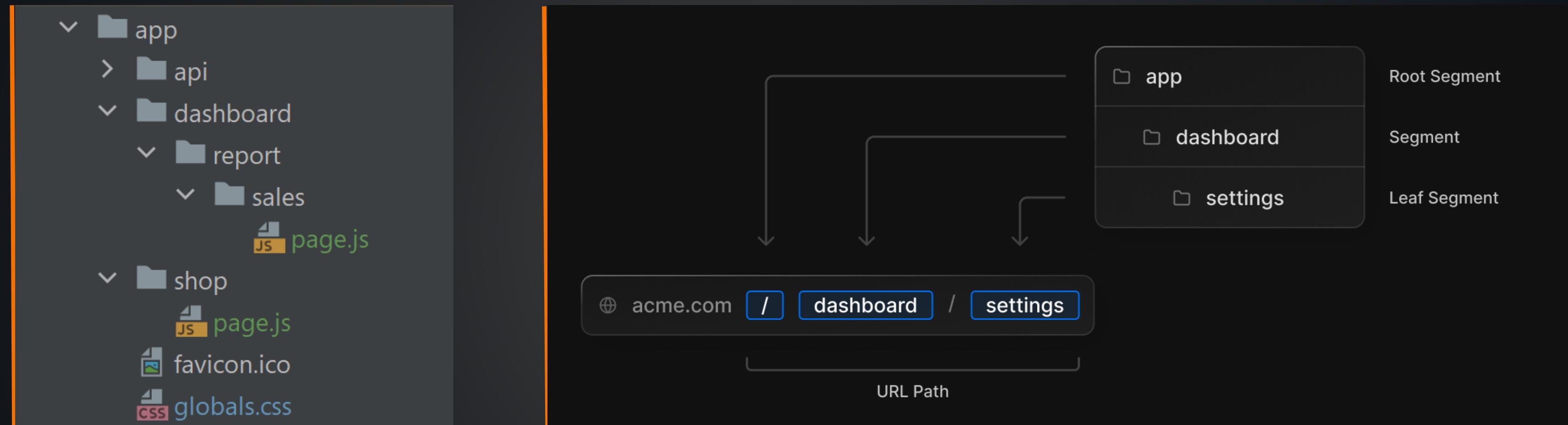
```
1 import React from 'react';
2 const Page = () => {
3     return (
4         <div>
5             <h1>This is shop page</h1>
6         </div>
7     );
8 };
9 export default Page;
```

● ● ● page.js

```
1 import React from 'react';
2 const Page = () => {
3     return (
4         <div>
5             <h1>This is product page</h1>
6         </div>
7     );
8 };
9 export default Page;
```

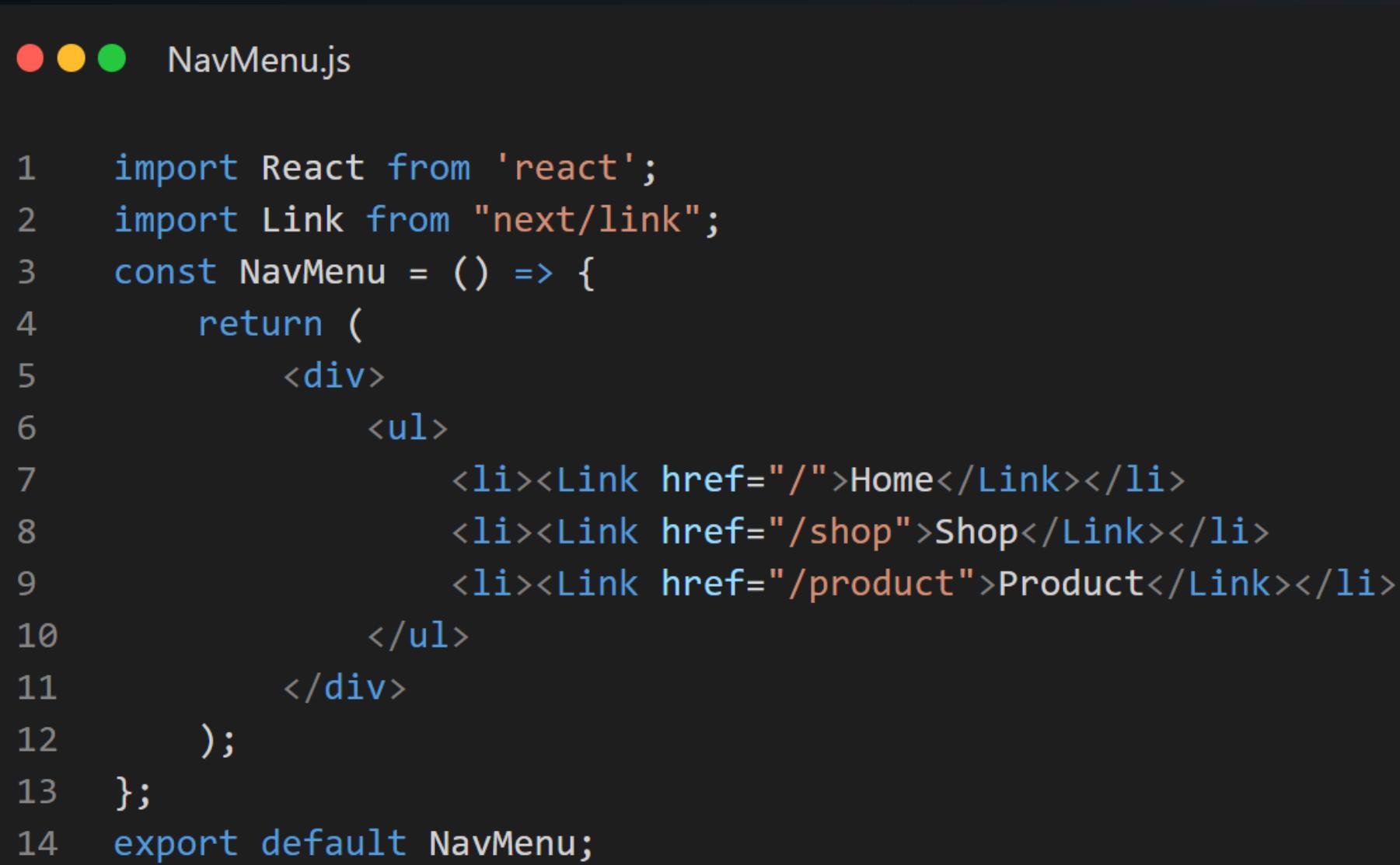
NEXT.JS

Basic Nested Routing Example



BASIC LINK COMPONENT

- <Link> is a React component that extends the HTML <a> element to provide **prefetching** and client-side navigation between routes.
- It is the **primary way** to navigate between routes in Next.js.



```
● ● ● NavMenu.js

1 import React from 'react';
2 import Link from "next/link";
3 const NavMenu = () => {
4     return (
5         <div>
6             <ul>
7                 <li><Link href="/">Home</Link></li>
8                 <li><Link href="/shop">Shop</Link></li>
9                 <li><Link href="/product">Product</Link></li>
10            </ul>
11        </div>
12    );
13 };
14 export default NavMenu;
```

ROUTING PROGRESS

● ● ● layout.js

```
1  'use client';

2

3  import { AppProgressBar as ProgressBar } from 'next-nprogress-bar';

4

5  export default function RootLayout({ children }) {
6      return (
7          <html lang="en">
8              <body>
9                  <ProgressBar
10                      height="4px"
11                      color="#ffffd00"
12                      options={{ showSpinner: false }}
13                  />
14                  {children}
15              </body>
16          </html>
17      )
18  }
19
```

MANAGING ACTIVE LINK

usePathname in next navigation

● ● ● Menu.js

```
1  'use client'
2  import React from 'react';
3  import Link from "next/link";
4  import { usePathname } from 'next/navigation';
5
6  const Menu = () => {
7      const currentRoute = ~/Desktop/next/my-app/app/profile • Contains
8          emphasized items
8      return (
9          <div>
10             <Link className={currentRoute === "/" ? "active-link" : "pending-link"} href={'/'}>Home</Link>
11             <Link className={currentRoute === "/profile" ? "active-link" : "pending-link"} href={'/profile'}>Profile</Link>
12             <Link className={currentRoute === "/product" ? "active-link" : "pending-link"} href={'/product'}>Product</Link>
13         </div>
14     );
15 };
16
17 export default Menu;
```

LINK COMPONENT QUERY

● ● ● NavMenu.js

```
6  <ul>
7    <li><Link href="/">Home</Link></li>
8    <li><Link href={{pathname: "/shop", query: {name: "URL Params From Link"}}}>Shop</Link></li>
9    <li><Link href="/product">Product</Link></li>
10 </ul>
```

● ● ● page.js

```
1 import React from 'react';
2 const Page = ({ searchParams }) => {
3   return (
4     <div>
5       <h1>This is shop page</h1>
6       {searchParams.name}
7     </div>
8   );
9 };
10 export default Page;
```

LINK COMPONENT QUERY

useSearchParams in next navigation

● ● ● NavMenu.js

```
6  <ul>
7    <li><Link href="/">Home</Link></li>
8    <li><Link href={{pathname: "/shop", query: {name: "URL Params From Link"}}}>Shop</Link></li>
9    <li><Link href="/product">Product</Link></li>
10 </ul>
```

● ● ● page.js

```
4  import { useSearchParams } from 'next/navigation'
5  const Page = () => {
6    const searchParams = useSearchParams()
7    const name = searchParams.get('name')
8    return (
9      <div>
10        <Menu/>
11        <h1>Profile Page {name}</h1>
12      </div>
13    );
14  };
```

PROPS FOR THE LINK COMPONENT

replace

Defaults to false. When true, next/link will replace the current history state instead of adding a new URL into the browser's history stack

```
<Link href="/dashboard" replace>Dashboard</Link>;
```

href

can also accept an object

```
<Link
  href={{
    pathname: '/about',
    query: { name: 'test' },
  }}
>
  About
</Link>
```

href

The path or URL to navigate to

```
<Link href="/dashboard">Dashboard</Link>
```

prefetch

Defaults to true. When true, next/link will prefetch the page (denoted by the href) in the background.

```
<Link href="/dashboard" prefetch={false}>Dashboard</Link>;
```

useRouter

The **useRouter hook** allows you to programmatically change routes inside Client Components.

router.push(href: string): Perform a client-side navigation to the provided route. Adds a new entry into the browser's history stack.

router.replace(href: string): Perform a client-side navigation to the provided route without adding a new entry into the browser's history stack

router.refresh(): Refresh the current route. Making a new request to the server, re-fetching data requests, and re-rendering Server Components.

router.prefetch(href: string): Prefetch the provided route for faster client-side transitions

router.back(): Navigate back to the previous route in the browser's history stack.

router.forward(): Navigate forwards to the next page in the browser's history stack