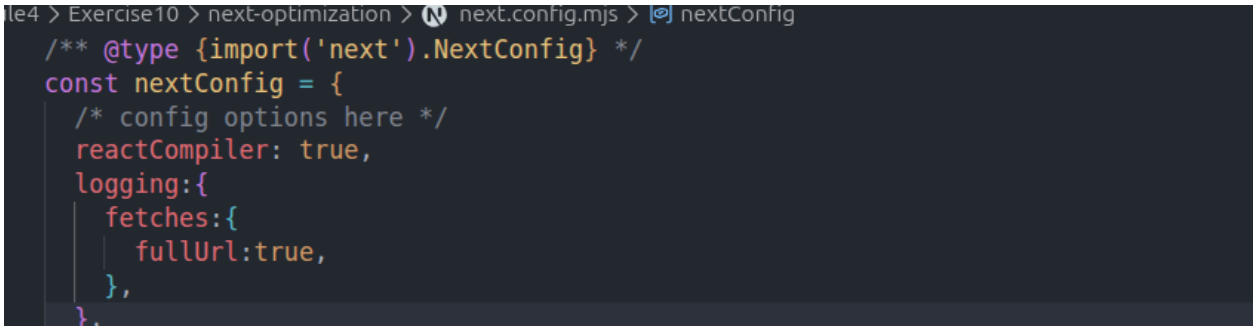
Module 4

Exercise 10(Optimization & Performance)

Q 1.) Enable logging in next js config to experience the memoization using fetch variations [‘force-cache’/ ‘no-store’]

A1.) Next.config



Nextjs

```
le4 > Exercise10 > next-optimization > app > Memoization > page.jsx > getCachedData > res
import './page.css'
async function getCachedData() {
  console.log("Fetching Forced cached data");

  const res = await fetch(
    "https://jsonplaceholder.typicode.com/users/2",
    { cache: "force-cache" }
  );

  return res.json();
}

async function getNoStoreData() {
  console.log("Fetching No Store Data");

  const res = await fetch(
    "https://jsonplaceholder.typicode.com/users/1",
    { cache: "no-store" }
  );

  return res.json();
}

export default async function Memoization() {
  const cached = await getCachedData();
  const noStore = await getNoStoreData();

  return (
    <div className="memo">
      <h2>Check Terminal</h2>

      <pre>{JSON.stringify(cached, null, 2)}</pre>
      <pre>{JSON.stringify(noStore, null, 2)}</pre>
    </div>
  );
}
```

Output

```
Fetching Forced cached data
Fetching No Store Data
 GET /Memoization 200 in 621ms (compile: 27ms, render: 593ms)
 | GET https://jsonplaceholder.typicode.com/users/2 200 in 369ms (cache skip)
 | | Cache skipped reason: (cache-control: no-cache (hard refresh))
 | GET https://jsonplaceholder.typicode.com/users/1 200 in 193ms (cache skip)
 | | Cache skipped reason: (cache: no-store)
Fetching Forced cached data
Fetching No Store Data
 GET /Memoization 200 in 402ms (compile: 2ms, render: 400ms)
 | GET https://jsonplaceholder.typicode.com/users/2 200 in 5ms (cache hit)
 | GET https://jsonplaceholder.typicode.com/users/1 200 in 350ms (cache skip)
 | | Cache skipped reason: (cache: no-store)
```

Q 2.) Create a revalidate action to purge the caching using revalidate tag.

A2.) action.js

Module4 > Exercise10 > next-optimization > app > action.js > revalidateTodoData

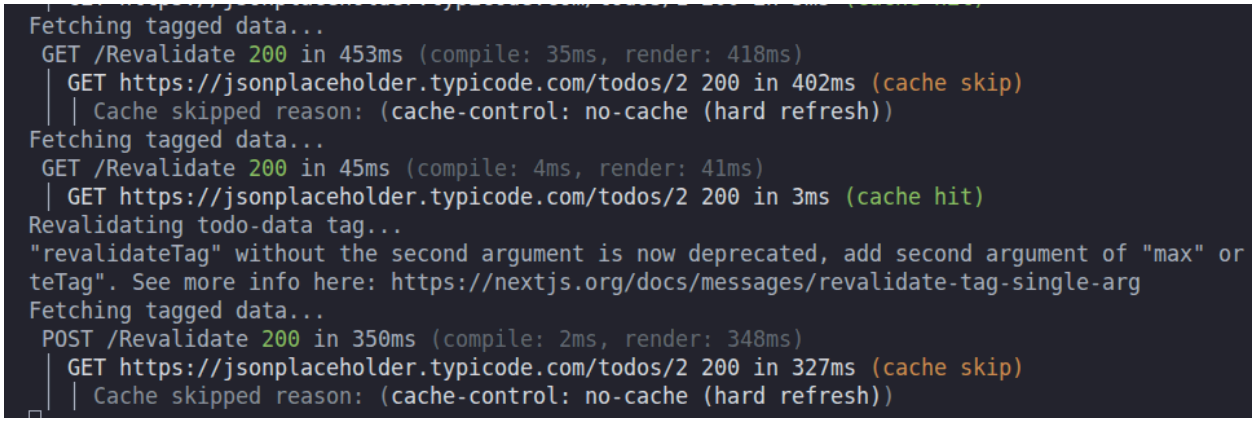
```
1  "use server";
2
3  import { revalidateTag } from "next/cache";
4
5  export async function revalidateTodoData() {
6    console.log("Revalidating todo-data tag...");
7    revalidateTag("todo-data");
8  }
```

Next.js

Module4 > Exercise10 > next-optimization > app > Revalidate > page.jsx > ...

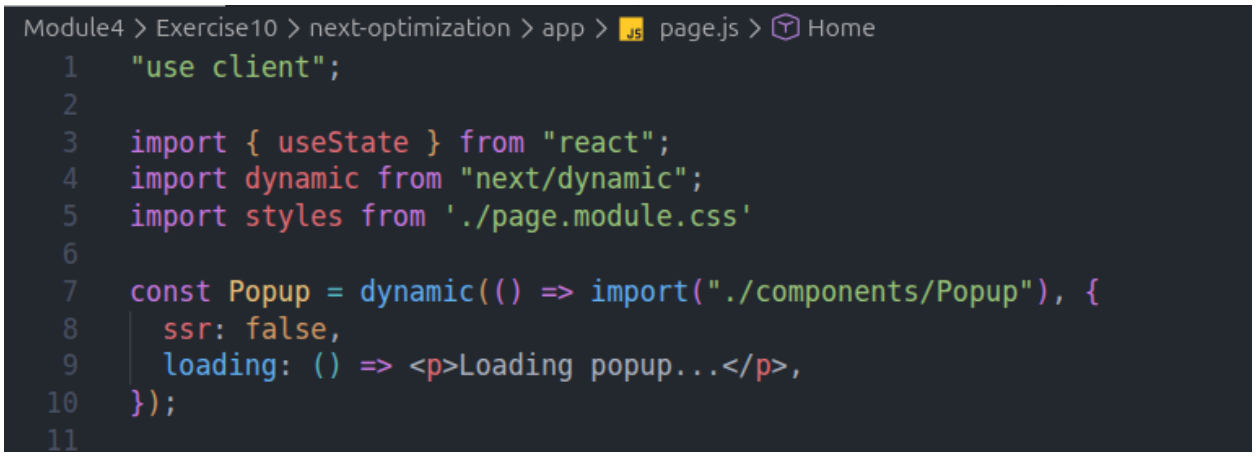
```
1  import { revalidateTodoData } from "../action";
2  import './page.css'
3  async function getData() {
4    console.log("Fetching tagged data...");
5
6    const res = await fetch(
7      "https://jsonplaceholder.typicode.com/todos/1",
8      {
9        cache: "force-cache",
10       next: {
11         tags: ["todo-data"],
12       },
13     }
14   );
15
16   return res.json();
17 }
18
19 export default async function Page() {
20   const data = await getData();
21
22   return (
23     <div className="tagged">
24       <h1>Tagged Cache Example</h1>
25
26       <pre>{JSON.stringify(data, null, 2)}</pre>
27
28       <form action={revalidateTodoData}>
29         <button type="submit">Revalidate Cache</button>
30       </form>
31     </div>
32   );
33 }
```

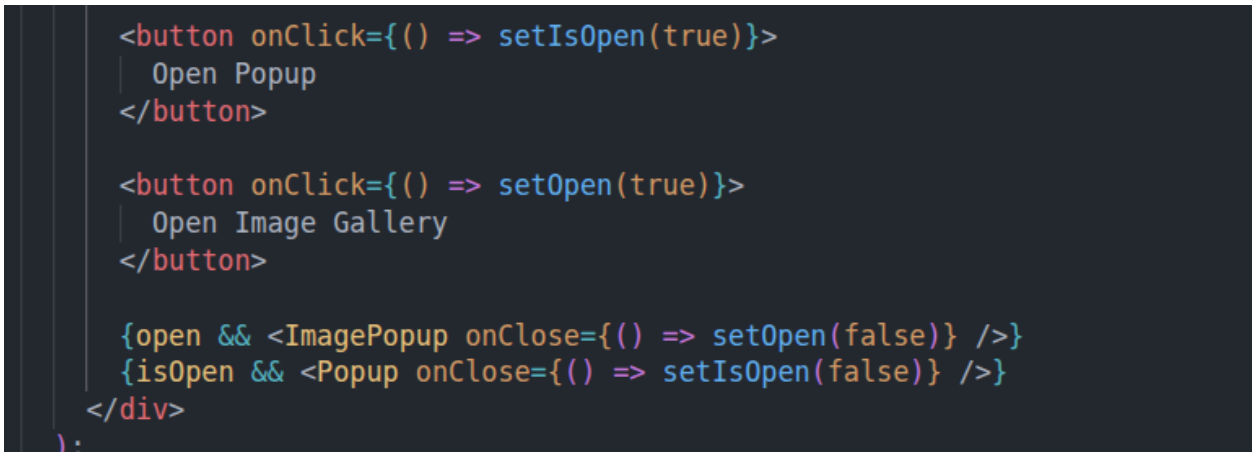
Output


Fetching tagged data...
GET /Revalidate 200 in 453ms (compile: 35ms, render: 418ms)
GET https://jsonplaceholder.typicode.com/todos/2 200 in 402ms (cache skip)
Cache skipped reason: (cache-control: no-cache (hard refresh))
Fetching tagged data...
GET /Revalidate 200 in 45ms (compile: 4ms, render: 41ms)
GET https://jsonplaceholder.typicode.com/todos/2 200 in 3ms (cache hit)
Revalidating todo-data tag...
"revalidateTag" without the second argument is now deprecated, add second argument of "max" or
teTag". See more info here: https://nextjs.org/docs/messages/revalidate-tag-single-arg
Fetching tagged data...
POST /Revalidate 200 in 350ms (compile: 2ms, render: 348ms)
GET https://jsonplaceholder.typicode.com/todos/2 200 in 327ms (cache skip)
Cache skipped reason: (cache-control: no-cache (hard refresh))

Q 3.) Create a pop-up component and a button , pop-up component should be imported dynamically and should open on button click.

A3.) Next.js



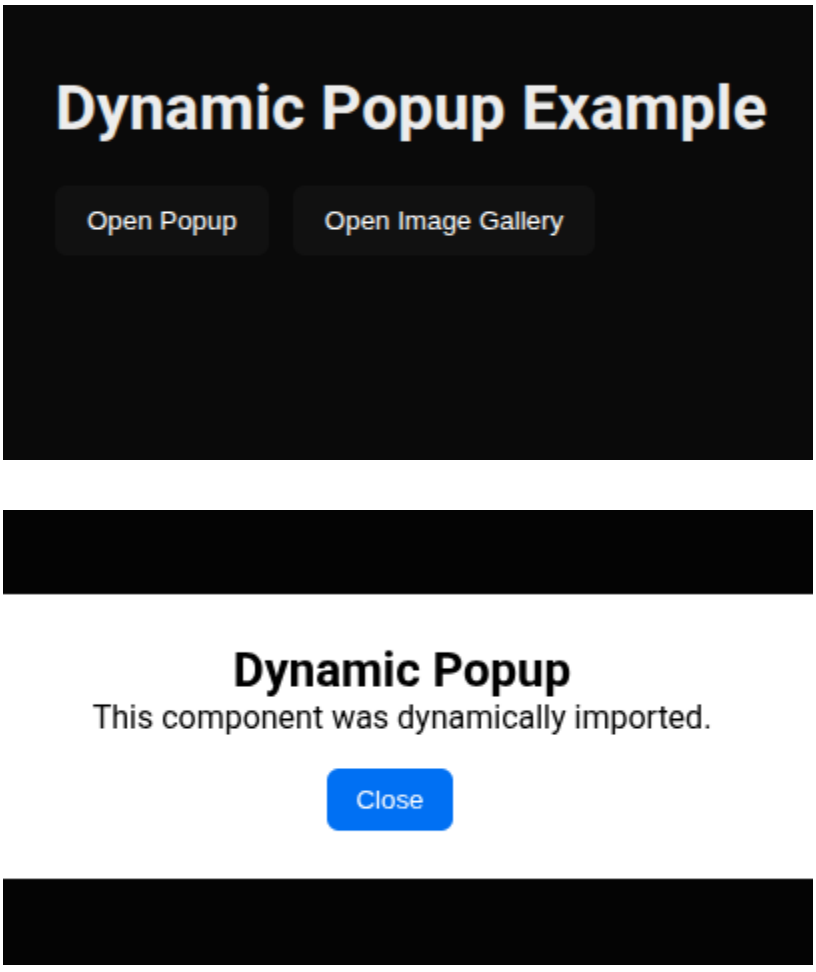


Popup.jsx

Module4 > Exercise10 > next-optimization > app > components > Popup.jsx > ...

```jsx
"use client";
import './Popup.css'
export default function Popup({ onClose }) {
  return (
    <div className="popup">
      <div>
        <h2>Dynamic Popup</h2>
        <p>This component was dynamically imported.</p>
        <button onClick={onClose}>Close</button>
      </div>
    </div>
  );
}
```

Output



Q 4.) Use a dummy api that provides images and use next image component to render it.

A4.) Nextconfig

```
  images: {
    remotePatterns: [
      {
        protocol: "https",
        hostname: "picsum.photos",
      },
    ],
  },
};
```

Page.jsx

```
const ImagePopup = dynamic(
  () => import("./components/ImagePopup"),
  { ssr: false }
);
```

```
    <button onClick={() => setIsOpen(true)}>
      Open Popup
    </button>

    <button onClick={() => setOpen(true)}>
      Open Image Gallery
    </button>

    {open && <ImagePopup onClose={() => setOpen(false)} />}
    {isOpen && <Popup onClose={() => setIsOpen(false)} />}
  </div>
);
```

ImagePopup.jsx

Module4 > Exercise10 > next-optimization > app > components > ImagePopup.jsx > ImagePopup

```jsx
"use client";

import { useEffect, useState } from "react";
import Image from "next/image";

export default function ImagePopup({ onClose }) {
  const [images, setImages] = useState([]);

  useEffect(() => {
    async function fetchImages() {
      const res = await fetch("https://picsum.photos/v2/list?page=1&limit=5");
      const data = await res.json();
      setImages(data);
    }

    fetchImages();
  }, []);

  return (
    <div className="img-popup">
      <div>
        <h2>Image Gallery</h2>

        <div>
          {images.map((img) => (
            <Image
              key={img.id}
              src={img.download_url}
              alt={img.author}
              width={200}
              height={150}
            />
          ))}
        </div>

        <button onClick={onClose}>Close</button>
```

Output

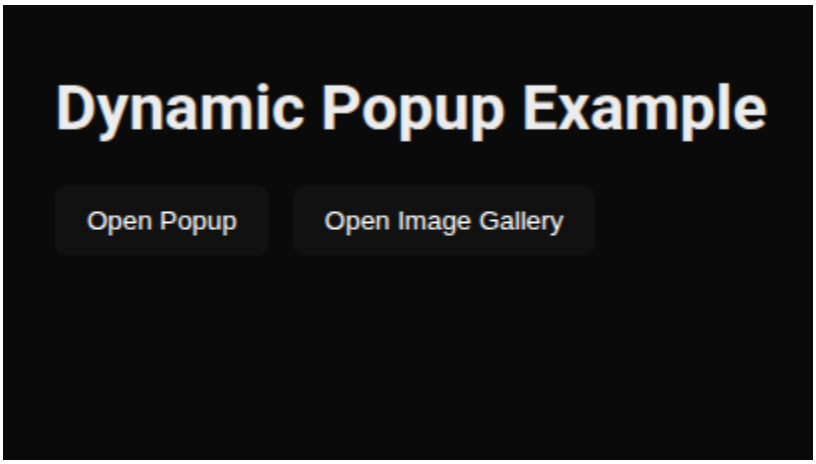
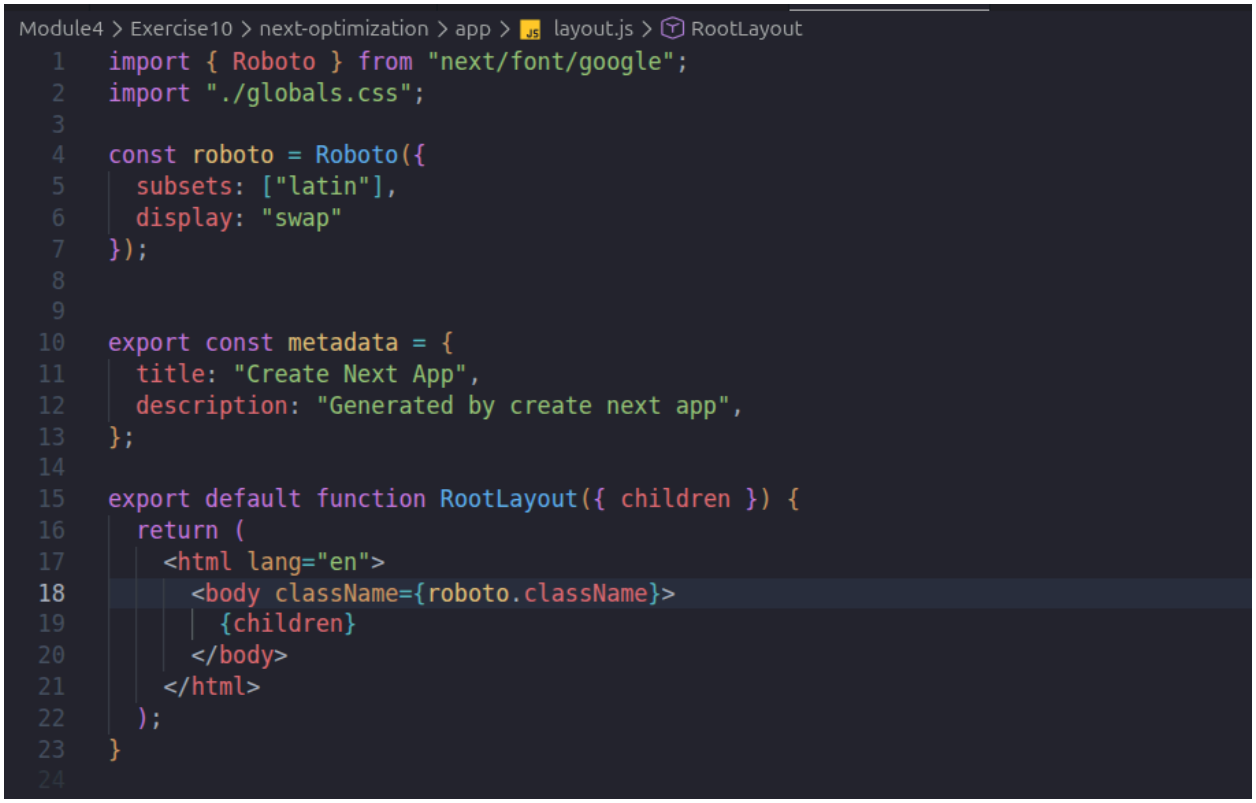


Image Gallery

Close

Q 5.) Change the font of pages implementing next/font.

A 5.)

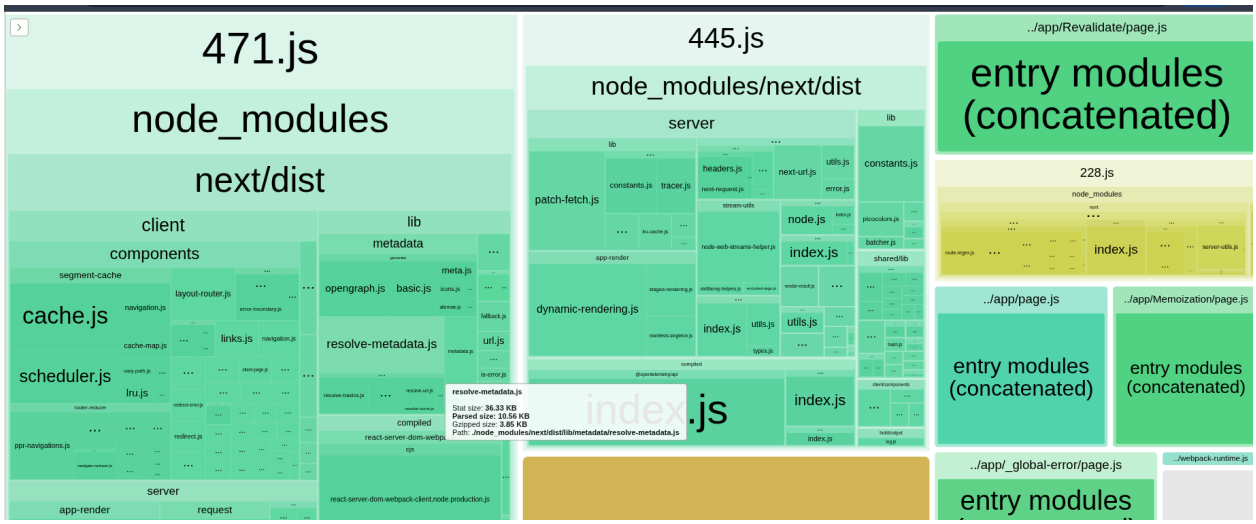


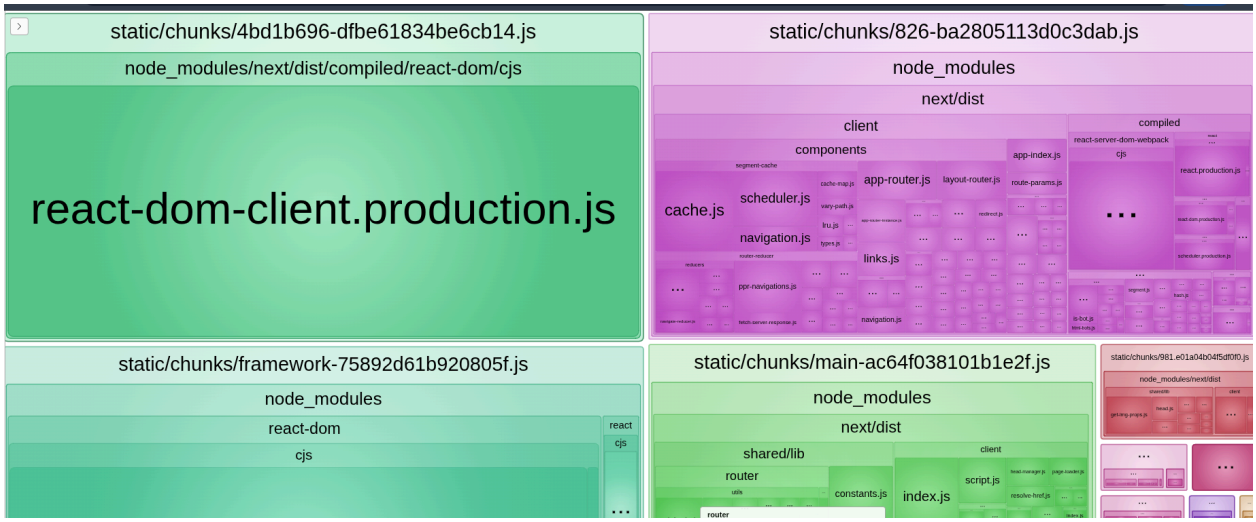
Q 6.) Install bundle analyzer and review the bundle of your project.

A6.)

```
import bundleAnalyzer from "@next/bundle-analyzer";

const withBundleAnalyzer = bundleAnalyzer({
  enabled: process.env.ANALYZE === "true",
});



export default withBundleAnalyzer(nextConfig);
```

Output




static/chunks/4bd1b696-dfbe61834be6cb14.js
node_modules/next/dist/compiled/react-dom/cjs
react-dom-client.production.js
static/chunks/826-ba2805113d0c3dab.js
node_modules
next/dist
client
components
cache.js
scheduler.js
navigation.js
app-router.js
layout-router.js
links.js
app-index.js
route-params.js
compiled
react-server-dom-webpack
cjs
react.production.js
static/chunks/framework-75892d61b920805f.js
node_modules
react-dom
cjs
react
cjs
static/chunks/main-ac64f038101b1e2f.js
node_modules
next/dist
shared/lib
router
client
script.js
index.js
constants.js
static/chunks/981.e01a04b04f5df0f0.js
node_modules/next/dist