

Virtual DOM is object representation of actual DOM.

AUGUST

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
01	02	03	04	05	06	07	08	09	10	11	12	13	14	
15	16	17	18	19	20	21	22	23	24	25	26	27	28	
29	30													

THURSDAY

Week 35

Day (240-125)

28

Appointments

Whenever state variable update, react triggers a reconciliation (R-render the component cycle).

```
useEffect ( () => {
```

```
  console.log ( "hello" ); // This runs when  
                             // component mounts
```

```
  return ( ) => console.log ( "bye" ); // This when  
  }, [ ] ); // component unmounts
```

Infinite Scrolling & how to do it is in code?
with explanation.

It uses Intersection Observer class & useRef.

Episode-07 Finding the Path

When the useEffect called?

The useEffect is called after the render of the component.

→ means it will run everytime when we update state or anything in component.
If no dependency array => useEffect is called on every render.

If empty dependency array => useEffect is called only on initial render

(just once when the component is rendered for first time)

→ It will not run everytime when we change state or anything in component.

2025

29

FRIDAY

Week 35

Day (241-124)

AUGUST

A	M	T	W	T	F	S	S	M	T	W	T	F	S	S
U						01	02	03	04	05	06	07	08	09
G	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	25	26	27	28	29	30	31							

Appointments

09 UseEffect does not cause re-render itself unless there is a state in it.

10

If dependency array is [btnName React] then useEffect is called after btnName React changes, although it will be ^{everytime} called on initial render.

12

13 Some useState tips

- 14 Never ~~use~~ ^{create} useState outside the body of component
- Always try to create useState at the top of body of component
- 16 Don't create useState inside any condition (like if-else) cause it
- 17 can have inconsistency as due to condition, there will be that state variable sometimes not.
- Don't create inside loop.

Making Routes in react application.

The library we will be using for this react router dom

For creating route, we have to create routing configuration

2025

AUGUST

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
01	02	03	04	05	06	07	08	09	10	11	12	13	14	
15	16	17	18	19	20	21	22	23	24	25	26	27	28	
29	30													

In App.js

Appointments

09 import { createBrowserRouter } from "react-router-dom";
RouterProvider

10
11 const AppLayout = () => {
12 return(
13 <div>
14 <Head />
15 <Body />
16 </div>
17);
18 }

19 } ← creating config

20 const appRouter = createBrowserRouter([
21 {
22 path: "/",
23 element: <AppLayout />
24 },
25 {
26 path: "/about",
27 element: <About />
28 }
29]]);

31 SUNDAY

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<RouterProvider router={appRouter} />);
→ Rendering config.

2025

01

MONDAY

Week 36 Day (244/121)

SEPTEMBER

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
	01	02	03	04	05	06	07	08	09	10	11	12	13	14
P	15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30													

Appointments

09 How to add error page for your Route?

10

{

11

path: "/"

element: <AppLayout />

12

errorElement: <Error />

}

13

There is a hook provided by react-router-dom.
14 called useRouteError

15 which gives more info about error

16 const err = useRouteError();

17 { err.status }
(404)

{ err.statusText }
(Not Found)

What if we want our Header should remain intact in every page.
In App.js

const AppLayout = () => {
return (

<div className = "app">

<Header />

<Body />

</div>

2025

};

Action Plan

Schedule your planFor Better future.

September
2025

Wk	36	37	38	39	40
----	----	----	----	----	----

What we want that `<Header/>` remain there but our `<Body/>` change for About page & other pages.

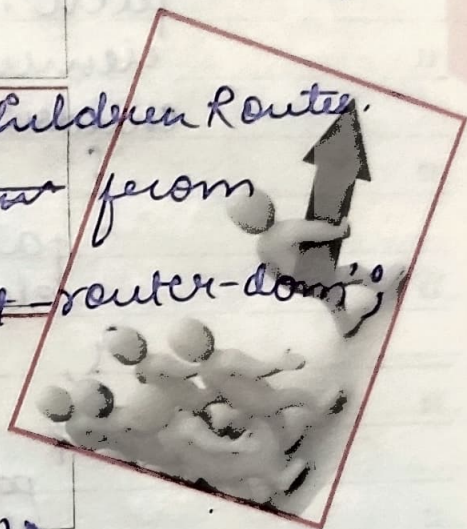
To do this we will use children Router and Outlet component ~~or~~ from `react-router-dom`

`import { Outlet } from 'react-router-dom'`
`const AppLayout = () => {`
`return (`

`<div className="app">`
`<Header />`
`<Outlet />`

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

Now in app router



**Be
Leader**

If your actions
inspire others to
dream more, learn
more or do more

You are a
Leader.

SEPTEMBER

OCTOBER

NOVEMBER

DECEMBER

02

TUESDAY

Week 36

Day (245-120)

SEPTEMBER

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
E	01	02	03	04	05	06	07	08	09	10	11	12	13	14
P	15	16	17	18	19	20	21	22	23	24	25	26	27	28
25	29	30												

Appointments

```

09 const appRouter = createBrowserRouter ([
10   {
11     path: "/"
12     element: <AppLayout />
13     children: [
14       {
15         path: "/"
16         element: <Body />
17       },
18       {
19         path: "/about"
20         element: <About />
21       },
22       {
23         path: "/contact"
24         element: <Contact />
25       }
26     ]
27     errorElement: <Error />
28   }
29 ] )

```

Now the outlet will be replaced by children a/c to routes.

For eg: - If we are on "/" then it will be replaced by <Body /> & if we are on "/about" then replaced by <About />

SEPTMBER

0 M T W T F S S 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

WEDNESDAY

Week 36

Day (246-119)

03

Appointments

Navigation in React

09

If we use `<a>` tag for navigation then the page will ~~refresh~~ reload & it will reload the page which will take time

11

So, we will use Link component from react-router-dom then the page will not reload it will just replace the things which ~~is~~ different from the page from we are navigating

14

`<Link to = "path to your page"> Home </Link>`

Two types of routing in web apps

- ① Client-Side Routing
- ② Server-Side Routing

⇒ ~~Server~~ - Side Routing

If we have an `index.html`, `about.html` if we click on anchor tag ("`/about`") it will reload the whole page, it sends a n/w call to `about.html`, fetches it and render on our webpage and reload the whole page.

⇒ Client-Side Routing

Here we doesn't make n/w all the ~~routes~~ components code is in our app so it just loads the component when we click on the route. The only n/w call we make is for api. ^{What we do here with} `<Link>` in routing.

04

THURSDAY

Week 36

Day (247-118)

SEPTEMBER

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
E	01	02	03	04	05	06	07	08	09	10	11	12	13	14
P	15	16	17	18	19	20	21	22	23	24	25	26	27	28
25	29	30												

Appointments

Dynamic Routing

10

To make a dynamic route we will use

11

```
const appRouter = createBrowserRouter([
  {
    path: "/restaurants/:resId"
    element: <RestaurantMenu />
  }
]);
```

12

13

14

To get the Id in the component we will use 'params' from react-router-dom

15

16

```
import { params } from "react-router-dom";

const { resId } = params;
```

17

Note :- Whenever you are using map function you have to use key to uniquely identify each item and it should on parent jsx.

```
restaurant.map((res) => {
  <Link key = "res.id" to = "/restaurants/:res.id"/>
  <RestaurantCard /> </Link>
});
```

2025

GraphQL is helps you with overfetching so
load only the data which is required.

SEPTEMBER
M T W T F S S M T W T F S
01 02 03 04 05 06 07 08 09 10 11 12
13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31

FRIDAY

Week 36

Day (248-117)

05

pointments Here Link is parent jex so we use key on it.

OCTOBER

DECEMBER