Routing In React

We can do routing in react using a very famous library `react-router-dom`. React internally doesn't posses any direct of routing users from one page to another. That's where this library helps.

How to install react-router-dom

```
npm install rteact-router-dom
```

How to configure react-router-dom ?

To configure react-router dom we need to do a couple of steps:

- Wrap your App component inside a new BrowserRouter component provided by react-router-dom

```
// inside main.jsx
import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.jsx'

import './index.css'

import { BrowserRouter } from 'react-router-dom'



ReactDOM.createRoot(document.getElementById('root')).render(

        <React.StrictMode>

                <BrowserRouter>

                        <App />

                </BrowserRouter>
```

```
        </React.StrictMode>,
)
```

- Then in the app component we have to use two components `Routes` and `Route`. Both of these are given by react-router-dom. `Route` component defined on which path a particular component must be rendered. `Routes` component keeps all the collection of individual `Route` together.

```
import { Route, Routes } from "react-router-dom";
function App() {
        return (
                <Routes>
                        <Route path="/play" element={<PlayGame />} />
                        <Route path="/start" element={<StartGame />} />
                        <Route path="*" element={<div> not found </div>} />
                </Routes>
        );
}
```

- Here we have defined 2 routes, one if /start and another is /play. If the user goes to the /play route then `<PlayGame />` component is rendered, and for /start route `<StartGame />` component is rendered.
- If the user goes to a route that is specifically not defined here, then we show Not Found to them. This is achieved by saying `path="*"` in one of the routes.

And with this, react-router-dom is setup completely, now we can use Link component or navigator from react-router-dom to move the user from one page to another without any refresh of the page.

Link component

Link component helps us to put a hyperlink(just like anchor tag) on the dom, where is the user clicks then they will be redirected to a particular route and it's corresponding mapped component is rendered **without any page refresh.**

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

function PlayGame() {
        return (

                <div>

                        <h1>Play Game</h1>



                        <Link to='/start'> Start Game </Link>

                </div>

        );
}

export default PlayGame;
```

Here we are providing a link to `/start` route using the Link tag. When rendered on the dom it converts to an anchor tag, but doesn't refresh the page like an anchor tag.

Navigator in react-router-dom

If we don't want the user to click on a hyperlink to move to another page and instead move the user to different pages programatically, then we can use navigator object.
Let's say we have a form and we first do a form validation and then redirect the user to a page, for this usecase naviqator is a perfect choice.
TO use navigator we can call the inbuilt `useNavigate` hook inside react-router-dom.

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

import TextInputFormContainer from
"../../components/TextInputForm/TextInputFormContainer";
function StartGame() {
        const navigate = useNavigate(); // this created the navigator object

        function handleSubmit() {

                navigate('/play'); // navigator takes the route and moves us
there
```

```
        }

        return (
                <div>
                        <h1>Start Game</h1>
                        <TextInputFormContainer onSubmit={handleSubmit} />
                </div>
        );
}

export default StartGame;
```