## CLIENT-SIDE ROUTING

Client-side routing is a web development technique where the routing or navigation of a web application is handled primarily on the client side (in the user's web browser) rather than on the server. In traditional server-side routing, when you request a new page or URL, the server processes the request and sends back the appropriate HTML page. In contrast, with client-side routing, the browser is responsible for rendering the appropriate content for different URLs or routes.

## KEY CHARACTERISTICS OF CLIENT-SIDE ROUTING

1. **Single Page Applications** (SPAs): Client-side routing is commonly used in SPAs, where the initial HTML page is loaded, and subsequent content updates are achieved through JavaScript. The entire application logic, including routing, is loaded when the user first visits the website.

2. **Asynchronous Navigation:** When a user clicks on a link or enters a URL, the web application uses JavaScript to update the page content without a full page reload. This results in smoother and faster transitions between different sections of the application.

3. **URL Manipulation:** Client-side routing typically relies on manipulating the browser's URL without causing a full page reload. This is often done using the HTML5 History API, which allows developers to change the URL and handle browser history without triggering a new server request.

4. **Route Handling:** Client-side routing libraries or frameworks (e.g., React Router, Vue Router, Angular Router) are commonly used to define and manage routes. These libraries enable developers to associate specific components or views with different URLs, making it easier to build and maintain complex web applications.

5. **Enhanced User Experience:** Client-side routing can lead to a more responsive and interactive user experience, as it reduces the time spent waiting for server responses. Users can navigate between pages without noticeable delays, which can be particularly beneficial for web applications that need to mimic the feel of native apps.

**Challenge with client-side routing**

**SEO Considerations:** One challenge with client-side routing is search engine optimization (SEO). Because the initial page load may contain minimal content, search engines might have difficulty indexing the entire site. ***Developers often implement server-side rendering (SSR) or other SEO-friendly techniques to address this issue.***

**Install React Router:** Install the React Router package in your project by running the following command:**npm install react-router-dom**

## BrowserRouter

BrowserRouter is a component provided by the React Router library, which is a used for adding routing and navigation functionality to React applications and enable client-side routing in web application**.**

1. **Client-Side Routing:** In a web application, when you navigate from one page to another, the browser traditionally sends a request to the server, which then responds with a new HTML page. This approach is called server-side routing. However, modern web applications often use client-side routing, where navigation happens on the client side without making requests to the server. This provides a smoother and more responsive user experience.

2. **URL Handling:**BrowserRouter is a key part of client-side routing in React applications. It allows you to manage the URL of your web application and handle route changes without a full page reload. It listens to changes in the URL and updates the view of your React application accordingly.

3. **Route Matching:**BrowserRouter is typically used in conjunction with other components like Route, Routes, and Link. The Route component allows you to specify which component should render when a specific route is matched in the URL. The Switch component is used to render the first Route that matches the current URL. The Link component is used for creating links that trigger route changes when clicked.

**Syntax**

```
<BrowserRouter>
<Routes>
<Route path="/" element={<Home />} />
<Route path="/about" element={<About />} />
<Route path="/contact" element={<Contact />} />
</Routes>
</BrowserRouter>
```

## <Routes>

The <Routes> component is part of the react-router-dom library, introduced in React Router version 6. It's used to define and configure the routing structure for your application. <Routes> is a replacement for the older <Switch> component and offers more flexibility and features for declarative routing in React applications. It supports:

1. **Declarative Routing:** <Routes> allows you to declare the routing configuration for your application in a more structured and declarative manner. You can define the routes, nested routes, and layout components within a hierarchical structure.

2. **Nested Routes:** You can easily define nested routes, making it simpler to manage and represent the structure of your application. Nested routes are useful when you have layouts or components that are shared across multiple routes.

3. **Component-Based Routing:** Routes are associated with components that should be rendered when a particular route is matched. This aligns with React's component-based architecture and makes it easy to encapsulate the functionality for each route within its associated component.<Routes>

```
    <Route path="/your-path"  element={<YourComponent />}    />
        {/* Additional routes */}
</Routes>
```

## Route

A Route is a fundamental component provided by routing libraries like react-router-dom in React applications. Its primary purpose is to define and configure how different parts of your application should be rendered based on the current URL or route. In other words, a Route component associates a specific URL pattern with a corresponding React component to be rendered when that URL is matched.
**Basic Route Syntax :<Route path="/your-path" component={YourComponent} />**

e.g.1
**App.js**

```
import {BrowserRouter, Routes, Route} from"react-router-dom"
functionApp() {
  return<div>
   <BrowserRouter>
    <Routes>
     <Routepath="/home"element={<h1>home page</h1>}>
    </Route>
    </Routes>
   </BrowserRouter>
  </div>
}
exportdefault App
```

**Route for multiple component:**we can use multiple components
e.g.
**components/Home.js**

```
functionHome(){
  return (<>
  <h1> Home Page</h1>
  </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
  return<>
  <h1> About Page</h1>
  </>
}
exportdefault About;
```

**App.js**

```
import {BrowserRouter, Routes, Route} from"react-router-dom"
import Home from"./Components/Home"
import About from"./Components/About"
functionApp() {
  return<div>
    <BrowserRouter>
      <Routes>
        <Routepath="/home"element={<Home/>}/>
        <Routepath="/about"element={<About/>}/>
        </Routes>
    </BrowserRouter>
  </div>
}
exportdefault App
```

## THE HOME PAGE (SETUP DEFAULT ROUTE )

***The home page, often referred to as the landing page or main page, is the initial page that users see when they visit a website or application***. It serves as the primary entry point and typically provides an overview or introduction to the website's content or features.

e.g.

**components/Home.js**

```
functionHome(){
  return (<>
  <h1> Home Page</h1>
  </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
  return<>
  <h1> About Page</h1>
  </>
}
```

```
exportdefault About;
```

**App.js**

```
import {BrowserRouter, Routes, Route} from"react-router-dom"
import Home from"./Components/Home"
import About from"./Components/About"
functionApp() {
  return<div>
    <BrowserRouter>
     <Routes>
       <Routepath="/"element={<Home/>}/>
       <Routepath="/about"element={<About/>}/>
       </Routes>
    </BrowserRouter>
  </div>
}
exportdefault App
```

### LINK COMPONENT

The Link component is part of the react-router-dom library, and it is used to create navigation links in a React application that uses client-side routing. to navigate between different pages or sections of a website or web application.

The Link component allows you to create links to different routes or pages within your application without causing a full page reload. It is a critical component for building single-page applications (SPAs) and improving the user experience by providing seamless navigation.

**Syntax :**

```
import { Link } from 'react-router-dom';
// Usage in JSX
<Link to="/target-route">Link Text</Link>
```

**Attributes:**

1. **to (required):** Specifies the target route or URL to which the link should navigate. This can be a string representing the path or a location object. For example, to="/about" or to={{ pathname: '/about', search: '?id=123' }}.

2. replace: An optional boolean attribute. If true, it replaces the current entry in the navigation history, effectively performing a "replace" action instead of a "push" action.

3. state: An optional object that allows you to pass state data to the target route. This state data can be accessed in the target route's component.

**e.g. - 1**

```
importReactfrom'react';
import{ BrowserRouter, Routes, Route, Link } from'react-router-dom';
```

```
functionApp() {
  return (
    <div>


      <BrowserRouter>
      <Linkto="/home">Home</Link><br/>
      <Linkto="/about">About</Link>
      <Routes>
        <Routepath="/home"element={<Home/>}/>
        <Routepath="/about"element={<About/>}/>
      </Routes>
     </BrowserRouter>


    </div>
  );
}

functionHome() {
  return (
    <div>
      <h1>Home Page</h1>
      <p>This is my Home Page</p>
    </div>
  );
}

functionAbout() {
  return (
    <div>
      <h1>About Page</h1>
      <p>This is my About Page</p>
    </div>
  );
}

exportdefaultApp;
```

**e.g. – 2**

**components/Home.js**

```
functionHome(){
   return (<>
   <h1> Home Page</h1>
   </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
   return<>
   <h1> About Page</h1>
   </>
}
exportdefault About;
```

**App.js**

```
import {BrowserRouter, Routes, Route, Link }from"react-router-dom"
import Home from"./Components/Home"
import About from"./Components/About"
functionApp() {
   return<div>
     <BrowserRouter>
     <Linkto ="/about"> About</Link>
     <br/>
     <Linkto ="/"> Home</Link>
       <Routes>
         <Routepath="/"element={<Home/>}/>
         <Routepath="/about"element={<About/>}/>
         </Routes>
     </BrowserRouter>
   </div>
}
exportdefault App
```

**e.g.-2**

```
import React from 'react';
import { BrowserRouter as Router, Route, Link } from 'react-router-dom';
function App() {
  const pageName = 'products';
  return (
<Router>
<div>
<h1>Navigation Examples</h1>
        {/* Using a String Path */}
<Link to="/about">Go to About Page</Link><br />
        {/* Using a Location Object */}
<Link to={{ pathname: '/about', search: '?id=123' }}>About Page with ID 123</Link><br />
        {/* Using Dynamic Route with Variables */}
<Link to={`/${pageName}`}>Go to {pageName} Page</Link><br />
</div>
</Router>
  );
}

export default App;
```

### Creation of Navigation

creating a navigation bar separately as its own component promotes code reusability, modularity, consistency, and flexibility. It simplifies development, maintenance, and updates, enhancing the user experience and facilitating scalability as your project evolves.

**e.g. -1**

App.js

```
import{ BrowserRouter, Routes, Route } from"react-router-dom";
import Home from"./Components/Home";
import About from"./Components/About";
import Navbar from"./Components/Navbar";
functionApp() {
  return (
    <div>
      <BrowserRouter>
      <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
        </Routes>
      </BrowserRouter>
    </div>
  );
}
exportdefault App;
```

```
Navbar.js
import {Link} from"react-router-dom"
functionNavbar(){
   return (
   <>
      <Linkto = "/about"> About</Link>
      <br/>
      <Linkto = "/"> Home</Link>
   </>
   )
}
export default Navbar
```

**e.g.-2**

**components/Home.js**

```
functionHome(){
   return (<>
   <h1> Home Page</h1>
   </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
   return<>
   <h1> About Page</h1>
   </>
}
exportdefault About;
```

**Navbar.js**

```
import{ Link} from"react-router-dom"
functionNavbar(){
   return (
   <>
      <Linkto = "/about"> About</Link>
      <br/>
      <Linkto = "/"> Home</Link>
      {/* <ul>
         <li><Link to = "/about"> About</Link></li>
         <li><Link to = "/"> Home</Link></li>
      </ul> */}

   </>
```

```
    )
}
export default Navbar
```

## useParams Hook

In React, the useParams hook is a built-in hook provided by the react-router-dom package. It allows you to access and retrieve the parameters from the URL in a React component.

**e.g.-1**

App.js

```
import{ BrowserRouter, Routes, Route } from"react-router-dom";
import Home from"./Components/Home";
import About from"./Components/About";
import Navbar from"./Components/Navbar";
import User from"./Components/User";
functionApp() {
  return (
    <div>
      <BrowserRouter>
        <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          {/* <Route path="/user" element={<User />} /> */}

          <Routepath="/user/:name"element={<User/>}/>
        </Routes>
      </BrowserRouter>
    </div>
  );
}
exportdefault App;
```

Navbar.js

```
import {Link} from"react-router-dom"
functionNavbar(){
    return (
    <>
        <Linkto = "/about"> About</Link>
        <br/>
        <Linkto = "/"> Home</Link>
        <br/>
        <Linkto = "/user"> User</Link>
    </>
    )
}
export default Navbar
```

**User.js**

```
import{ useParams } from"react-router-dom"
functionUser(){
    const params = useParams()
    const {name} = params
    console.log(name)
    return (<div>
        {/* <h1>Users page</h1> */}
        <h1>{name}'s' page</h1>
    </div>)
}

exportdefault User
```

**components/Home.js**

```
functionHome(){
    return (<>
    <h1> Home Page</h1>
    </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
    return<>
    <h1> About Page</h1>
    </>
}
exportdefault About;
```

**e.g.-2**

**Navbar.js**

```
import{ Link} from"react-router-dom"
functionNavbar(){
   return (
   <>

     <ul>
         <li><Linkto = "/about"> About</Link></li>
         <li><Linkto = "/"> Home</Link></li>
         <li><Linkto = "/user/naincy">naincy</Link></li>
         <li><Linkto = "/user/astha">astha</Link></li>
         <li><Linkto = "/user/kanishka">kanishka</Link></li>
      </ul>

   </>
   )
}
```

### SETUP 404 PAGE:

In React, a 404 page is a specific page that is displayed when a user tries to access a URL that does not exist or is not found within the application. It is a way to handle and gracefully inform users that the requested page or route is unavailable.PageNotFound.js

e.g.1

```
functionPageNotFound(){
return<div>
   <h1> 404 Page</h1>
   <p> page not found</p>

</div>
}
exportdefaultPageNotFound
```

App.js

```
import{ BrowserRouter, Routes, Route } from"react-router-dom";
import Home from"./Components/Home";
import About from"./Components/About";
import Navbar from"./Components/Navbar";
import User from"./Components/User";
importPageNotFoundfrom"./Components/PageNotFound";
functionApp() {
  return (
   <div>
     <BrowserRouter>
       <Navbar/>
```

```
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          {/* <Route path="/user" element={<User />} /> */}
          <Routepath="/user/:name"element={<User/>}/>



          <Routepath="/*"element={<PageNotFound/>}/>
        </Routes>
      </BrowserRouter>
    </div>
  );
}
exportdefault App;
```

Navbar.js

```
import{ Link} from"react-router-dom"
functionNavbar(){
    return (
    <>
        <ul>
            <li><Linkto = "/about"> About</Link></li>
            <li><Linkto = "/"> Home</Link></li>
            <li><Linkto = "/user/naincy">naincy</Link></li>
            <li><Linkto = "/user/astha">astha</Link></li>
            <li><Linkto = "/user/kanishka">kanishka</Link></li>
        </ul>
     </>
    )
}
exportdefault Navbar
```

User.js

```
import{ useParams } from"react-router-dom"
functionUser(){
    const params = useParams()
    const {name} = params
    console.log(name)
    return (<div>
        {/* <h1>Users page</h1> */}
        <h1>{name}'s' page</h1>
    </div>)
}
exportdefault User
```

**components/Home.js**

```
functionHome(){
    return (<>
    <h1> Home Page</h1>
    </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
    return<>
    <h1> About Page</h1>
    </>
}
exportdefault About;
```

## PROGRAMMATIC NAVIGATION: NAVIGATE

The Navigate component is used for programmatic navigation within your application. It allows you to navigate to a specific URL or route programmatically.

App.js

```
import {BrowserRouter, Routes, Route, Navigate }from"react-router-dom"
import Navbar from"./Components/Navbar"
import Home from"./Components/Home"
import About from"./Components/About"
importPageNotFoundfrom"./Components/PageNotFound"


functionApp() {
    return<div>
      <BrowserRouter>
      <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          {/* <Route path="/*" element={ <PageNotFound /> }/> */}
          {/* setup programmatic Navigation using navigate component  */}
          <Routepath="/*"element={<Navigateto ="/"/>  }/>
          </Routes>
      </BrowserRouter>
    </div>
}
export default App
```

```
functionPageNotFound(){
return<div>
   <h1> 404 Page</h1>
   <p> page not found</p>

</div>
}
exportdefaultPageNotFound
Navbar.js
import {Link} from"react-router-dom"
functionNavbar(){
   return (
   <>
      <Linkto = "/about"> About</Link>
      <br/>
      <Linkto = "/"> Home</Link>
   </>
   )
}
export default Navbar
```

**components/Home.js**

```
functionHome(){
   return (<>
   <h1> Home Page</h1>
   </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
   return<>
   <h1> About Page</h1>
   </>
}
exportdefault About;
```

## STYLING OF REACT COMPONENT

Styling a React component can be done using CSS, inline styles, or CSS-in-JS libraries. Here are a few approaches to consider:

**CSS Classes:** You can apply CSS styles to React components by assigning CSS classes to them. Define your styles in a separate CSS file or within a <style> tag, and then add the appropriate class name to the component using the className prop.

### Separate CSS file

App.js

```
import{ BrowserRouter, Routes, Route , Navigate} from"react-router-dom";
import Home from"./Components/Home";
import About from"./Components/About";
import Navbar from"./Components/Navbar";
import User from"./Components/User";
importPageNotFoundfrom"./Components/PageNotFound";
functionApp() {
  return (
    <div>
      <BrowserRouter>
        <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          {/* <Route path="/user" element={<User />} /> */}
          <Routepath="/user/:name"element={<User/>}/>



          <Routepath="/*"element={<Navigateto ="/"/>  }/>

        </Routes>
      </BrowserRouter>
    </div>
  );
}
exportdefault App;
```

**components/**NavBar.css

```
  .navbar{
  text-align: right;
}
.navbarli{
    font-weight: bold;
    display: inline-block;
    padding: 20px;
    background-color: blue;
    color :silver;
    font-weight: bold;

  }
  .navbar-link {
    color :white;
    font-weight: bold;
  }
```

**components/**Navbar.js

```
import{ Link } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
  return (
    <>
      <ulclassName="navbar">
        <li>
          <LinkclassName="navbar-link"to="/">
            Home
          </Link>
        </li>
        <li>
          <LinkclassName="navbar-link"to="/about">
            About
          </Link>
        </li>
        <li>
          <LinkclassName="navbar-link"to="/user/kanishka">
            naincy
          </Link>
        </li>
        <li>
          <Linkto="/user/astha">astha</Link>
        </li>
        <li>
          <Linkto="/user/kanishka">kanishka</Link>
        </li>
      </ul>
    </>
  );
}
exportdefault Navbar;
```

**components/**User.js

```
import{ useParams } from"react-router-dom"
functionUser(){
   const params = useParams()
   const {name} = params
   console.log(name)
   return (<div>
      {/* <h1>Users page</h1> */}
      <h1>{name}'s' page</h1>
   </div>)
}

exportdefault User
```

**components/Home.js**

```
functionHome(){
   return (<>
   <h1> Home Page</h1>
   </>)
}
exportdefault Home;
```

**components/About.js**

```
functionAbout(){
   return<>
   <h1> About Page</h1>
   </>
}
exportdefault About;
```

**Note:**

we cannot apply the csson react component such as <Link> ,<NavLink> by using tag name. however, we can access the style using the className.

### WITHIN A <STYLE>ATTRIBUTE :APPLYING INLINE CSS

e.g-1

```
import{ Link, NavLink } from"react-router-dom";
functionNavbar() {
   consttextStyles = {
      color: 'white',
      backgroundColor :'black'
     };
```

```
  return (
   <>
     <ulclassName="navbar">
     <li><Linkstyle={textStyles}  to = "/user/naincy">naincy</Link></li>
     </ul>
   </>
  );
}
exportdefault Navbar;
```

e.g-2

```
import{ Link,NavLink} from"react-router-dom"
functionNavbar(){
    return (
   <>
       <ulclassName="navbar">
           <li><Linkstyle={{color: 'red', fontWeight: 'bold', backgroundColor :'blue'}}  to =
"/user/naincy">naincy</Link></li>
       </ul>

   </>
    )
}
exportdefault Navbar
```

## Setup active link

To set up an active link in React, you can use the React Router library, specifically the ***NavLink*** component, to apply a specific style or class to the active link based on the current URL.

When the user navigates to a specific route, React Router will automatically apply the "active-link" class to the corresponding NavLink element, allowing you to style the active link differently from other links.

**Navbar.js**

e.g.-1

```
import{ NavLink } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
  return (
   <>
     <ulclassName="navbar">
       <li>
         <NavLinkclassName="navbar-link"to="/user/naincy">
           {" "}
```

```
          naincy
        </NavLink>
      </li>
      <li>
        <NavLinkclassName="navbar-link"to="/user/astha">
          {" "}
          astha
        </NavLink>
      </li>
      <li>
        <NavLinkclassName="navbar-link"to="/user/kanishka">
          {" "}
          kanishka
        </NavLink>
      </li>
    </ul>
  </>
 );
}
exportdefault Navbar
```

e.g.-2

```
import{ NavLink } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
 return (
  <>
   <ulclassName="navbar">
    <li>
     <NavLink
      // style={(isActive)=>{return{backgroundColor:"green"}}}
      style={({isActive})=>{return {backgroundColor: isActive ? "green" :"red"}}}
      className="navbar-link"to="/user/naincy">
      {" "}
      naincy
     </NavLink>
    </li>
    <li>
     <NavLinkclassName="navbar-link"to="/user/astha">
      {" "}
      astha
     </NavLink>
    </li>
    <li>
     <NavLinkclassName="navbar-link"to="/user/kanishka">
```

```
          {" "}
          kanishka
         </NavLink>
        </li>
       </ul>
      </>
    );
}
exportdefault Navbar
```

## Search param hook

The **useSearchParam**s hook is used to read and modify the query string in the URL for the current location. Like React's own useState hook, useSearchParams returns an array of two values: the current location's search params and a function that may be used to update them. Just as React'suseState hook, setSearchParams also supports functional updates. Therefore, you may provide a function that takes a searchParams and returns an updated version.

Filter.js

```
import{ useSearchParams } from"react-router-dom"
functionFilter(){
   const [searchParams,setSearchParams] = useSearchParams();
   console.log(searchParams.get('age'))
   const age = searchParams.get('age')
   console.log(searchParams.get('city'))
   const city = searchParams.get('city')
   return (
      <div>
         <h1> Filter Page </h1>
         <h1>Age is :{age}</h1>
         <h1>City is :{city}</h1>
         {/* <input type="text"
onChange={(event)=>{setSearchParams({age:event.target.value,city:"jaipur"})}} ></input>
         <button onClick={(e)=>{setSearchParams({age:40})}}> set age in query params</button>
            */}
      </div>
      )
}
exportdefaultFilter ;
```

**App.js**

```
import {BrowserRouter, Routes, Route,Navigate } from"react-router-dom"
import Navbar from"./Components/Navbar"
import Home from"./Components/Home"
import About from"./Components/About"
import User from"./Components/User"
import Filter from"./Components/Filter"
importPageNotFoundfrom"./Components/PageNotFound"
functionApp() {
  return<div>
   <BrowserRouter>
   <Navbar/>
    <Routes>
     <Routepath="/"element={<Home/>}/>
     <Routepath="/about"element={<About/>}/>
     <Routepath="/filter"element={<Filter/>}/>
     <Routepath="/user/:name"element={<User/>}/>
     {/* <Route path="/*" element={ <PageNotFound /> }/> */}
     <Routepath="/*"element={<Navigateto ="/"/>  }/>
     </Routes>
   </BrowserRouter>
  </div>
}
export default App
```

**Navbar.js**

```
import{ NavLink } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
 return (
  <>
   <ulclassName="navbar">
    <li>
     <NavLink
      className="navbar-link"to="/home">
      {" "}
      home
     </NavLink>
    </li>
    <li>
     <NavLinkactiveClassName="active-link"className="navbar-link"to="/about">
      {" "}
      about
     </NavLink>
```

```
        </li>
        <li>
          <NavLinkclassName="navbar-link"to="/filter">
            {" "}
            filter
          </NavLink>
        </li>
      </ul>
    </>
  );
}
exportdefault Navbar
```

## USENAVIGATE HOOK

The useNavigate hook is a hook provided by React Router that allows you to programmatically navigate to different routes within your React components. It is available in React Router v6.
e.g.
Home.js

```
import{ Link, useNavigate } from"react-router-dom";
functionHome() {
  const navigate = useNavigate();


  constnavtopage = (url) => {
    navigate(url);
  };
  return (
    <>
      <h1> Home Page</h1>
      <Linkto="/about">goto about us </Link>
      <buttononClick={() =>navtopage("/about")}>goto about page</button>
      <buttononClick={() =>navtopage("/filter")}>goto filter page</button>
    </>
  );
}
exportdefault Home
```

**App.js**

```
import {BrowserRouter, Routes, Route,Navigate } from"react-router-dom"
import Navbar from"./Components/Navbar"
import Home from"./Components/Home"
import About from"./Components/About"
import User from"./Components/User"
import Filter from"./Components/Filter"
importPageNotFoundfrom"./Components/PageNotFound"
functionApp() {
  return<div>
   <BrowserRouter>
   <Navbar/>
    <Routes>
     <Routepath="/"element={<Home/>}/>
     <Routepath="/about"element={<About/>}/>
     <Routepath="/filter"element={<Filter/>}/>
     <Routepath="/user/:name"element={<User/>}/>
     {/* <Route path="/*" element={ <PageNotFound /> }/> */}
     <Routepath="/*"element={<Navigateto ="/"/>  }/>
     </Routes>
   </BrowserRouter>
  </div>
}
export default App
```

**Navbar.js**

```
import{ NavLink } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
 return (
  <>
   <ulclassName="navbar">
    <li>
     <NavLink
      className="navbar-link"to="/home">
      {" "}
      home
     </NavLink>
    </li>
    <li>
     <NavLinkactiveClassName="active-link"className="navbar-link"to="/about">
      {" "}
      about
     </NavLink>
    </li>
    <li>
     <NavLinkclassName="navbar-link"to="/filter">
      {" "}
      filter
     </NavLink>
    </li>
   </ul>
```

```
    </>
  );
}
exportdefault Navbar
```

## Nested Routing

Nested routing, also known as nested routes or nested routing components, is a technique in React Router that allows you to define and render routes within other routes. It enables you to create hierarchical structures in your application where components can be nested and rendered based on the URL.

With nested routing, you can have components with their own routes and rendering logic that are nested within other components. This is useful when you want to encapsulate specific functionality or views within parent components and handle their routing independently.

e.g.

App.js

```
import{ BrowserRouter, Routes, Route, Navigate } from"react-router-dom";
import Home from"./Components/Home";
import About from"./Components/About";
import Navbar from"./Components/Navbar";
import Filter from"./Components/Filter";
import Contact from"./Components/Contact";
importCompnayfrom"./Components/Company";
import Channel from"./Components/Channel";
import Other from"./Components/Other";
functionApp() {
  return (
    <div>
      <BrowserRouter>
        <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          <Routepath="/filter"element={<Filter/>}/>
          <Routepath="/contact/"element={<Contact/>}>
            <Routepath="company"element={<Compnay/>}></Route>
            <Routepath="channel"element={<Channel/>}></Route>
            <Routepath="other"element={<Other/>}></Route>
            </Route>

        </Routes>
      </BrowserRouter>
    </div>
  );
}
exportdefault App;
```

```
Navbar.js
import{ NavLink } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
  return (
    <>
      <ulclassName="navbar">
        <li>
          <NavLink
            className="navbar-link"to="/home">
            {" "}
            home
          </NavLink>
        </li>
        <li>
          <NavLinkactiveClassName="active-link"className="navbar-link"to="/about">
            {" "}
            about
          </NavLink>
        </li>
        <li>
          <NavLinkclassName="navbar-link"to="/contact">
            {" "}
            contact us
          </NavLink>
        </li>
        <li>
          <NavLinkclassName="navbar-link"to="/filter">
            {" "}
            filter
          </NavLink>
        </li>
      </ul>
    </>
  );
}
exportdefault Navbar
```

Contact.js

```
import{ Link, Outlet } from"react-router-dom"
functionContact(){
    return( <div>
        <h1> contact us page </h1>
        <Linkto="company">Compnay</Link>
        <Linkto="channel"> Channel </Link>
        <Linkto="other"> other </Link>
        <Outlet/>
    </div>
    )
}
exportdefault Contact
```

Company.js

```
functionCompnay(){
    return( <div>
        <h1>Compnay us page </h1>

    </div>
    )
}
exportdefaultCompnay
```

Channel.js

```
functionChannel() {
  return (
    <div>
      <h1> You Tube channel page</h1>
    </div>
  );
}
exportdefault Channel;
```

Other.page

```
functionOther() {
  return (
    <div>
      <h1> You Tube Other page</h1>
    </div>
  );
}
exportdefault Other;
```

### <Outlet />

In React Router, the <Outlet /> component is used as a placeholder to render child components within the parent component associated with a particular route.

When using nested routes in React Router, a parent component often includes an <Outlet /> component. The purpose of the <Outlet /> component is to render the child components defined in the nested routes.

### useLocation

useLocation is a hook provided by React Router that allows you to access the current location object within a component.

The location object represents the current URL of the application and contains information about the path, search parameters, hash, and other details related to the URL.

e.g.-1

```
import{ Link, useLocation } from"react-router-dom";
functionHome(){
   const location = useLocation()
   console.log(location)
   return (<>
   <h1> Home Page</h1>
   <Linkto ="/about">goto about us  </Link>
   </>)
}
exportdefault Home;
```

e.g.-2

App.js

```
import {BrowserRouter, Routes, Route,Navigate } from"react-router-dom"
import Navbar from"./Components/Navbar"
import Home from"./Components/Home"
import About from"./Components/About"
import User from"./Components/User"
functionApp() {
   return<div>
```

```
      <BrowserRouter>
      <Navbar/>
        <Routes>
          <Routepath="/"element={<Home/>}/>
          <Routepath="/about"element={<About/>}/>
          <Routepath="/user/:name"element={<User/>}/>
          {/* <Route path="/*" element={ <PageNotFound /> }/> */}
          <Routepath="/*"element={<Navigateto ="/"/>  }/>
          </Routes>
      </BrowserRouter>
    </div>
}
export default App
```

Navbar.js

```
import{ Link } from"react-router-dom";
import"./Navbar.css";
functionNavbar() {
  return (
    <>
      <ulclassName="navbar">
        <li>
          <LinkclassName="navbar-link"to="/home">
            {" "}
            home
          </Link>
        </li>
        <li>
          <LinkclassName="navbar-link"to="/about">
            {" "}
            about
          </Link>
        </li>
        <li>
          <LinkclassName="navbar-link"to="/contact">
            {" "}
            contact us
          </Link>
        </li>
        <li>
          <Linkto="/user/astha"state={{name:'amit'}}>astha</Link>
        </li>
        <li>
          <Linkto="/user/kanishka">kanishka</Link>
        </li>
      </ul>
    </>
  );
}
exportdefault Navbar;
```

User.js

```
import{ useParams,useLocation } from"react-router-dom"
functionUser(){
    const params = useParams()
    const {name} = params
    const location = useLocation()
    console.log(location)
    return (<div>
        <h1>Users page</h1>
        <h1>{name}'s' page</h1>
    </div>)
}

exportdefault User
```

## PROTECTED ROUTING

Protected routing is a technique used to restrict access to certain routes or pages in a web application. It ensures that only authenticated or authorized users can access specific routes or content.

Protected routes typically require users to be logged in or meet certain authorization criteria before they can access the protected content. This helps secure sensitive information, restrict unauthorized access, and maintain the privacy and integrity of the application

**App.js**

```
import {BrowserRouter, Routes, Route, Navigate }from"react-router-dom"
import Home from"./Components/Home"
import About from"./Components/About"
import Navbar from"./Components/Navbar"
import Filter from"./Components/Filter"
importPageNotFoundfrom"./Components/PageNotFound"
import User from"./Components/User"
import Contact from"./Components/Contact"
importCompnayfrom"./Components/Company"
import Channel from"./Components/Channel"
import Login from"./Components/Login"
import Protected from"./Components/Protected"
functionApp() {
  return<div>
   <BrowserRouter>
  <Navbar/>
    <Routes>
      <Routepath="/"element={<ProtectedComponent={Home}/>}/>
      <Routepath="/about"element={<About/>}/>
      <Routepath="/user/:name"element={<User/>}/>
      <Routepath="/*"element={<Navigateto ="/"></Navigate>}/>
      <Routepath="/filter"element={<Filter/>}/>
      <Routepath="/contact/"element={<Contact/>}>
       <Routepath="company"element={<Compnay/>}></Route>\
       <Routepath="channel"element={<Channel/>}></Route>\

      </Route>
      <Routepath="/login"element={<Login/>}/>
    </Routes>
   </BrowserRouter>
  </div>
}
exportdefault App
```

**login.js**

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

functionLogin(){
  const navigate = useNavigate()
  const login =()=>{
    localStorage.setItem('login',true)
    navigate("/")

  }
  return( <div>
    <h1> login page</h1>
```

```
      <inputtype="text"/>
      <inputtype="text"/>
       <buttononClick={login}>Login </button>

    </div>
    )
}
exportdefault Login
```

**Protected.js**

```
import{ useEffect } from"react"
import{ Link,Navigate, useNavigate } from"react-router-dom"

function Protected(props){
    const {Component} = props
    const navigate = useNavigate()
    useEffect (()=>{
        let login = localStorage.getItem('login')
        if( !login){
            navigate('/login')
        }

    })
    return( <div>
        abc
       <Component/>


    </div>
    )
}
exportdefault Protected
```

## DYNAMIC ROUTING

**Dynamic routing in React allows you to create routes that change based on user input or data. It's about showing different content or components depending on the URL.** Dynamic routing in React refers to the ability to create and manage routes that change based on user input, data, or other dynamic factors. React is a JavaScript library commonly used for building single-page applications (SPAs), and dynamic routing is a fundamental aspect of creating interactive and flexible web applications. Dynamic routing allows you to respond to user interactions and display different content or components based on the current route or URL parameters.

**src/product.Data.js**

```
const products = [
{ id: 1, name: "Product 1", description: "Description for Product 1" },
{ id: 2, name: "Product 2", description: "Description for Product 2" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },
{ id: 3, name: "Product 3", description: "Description for Product 3" },

  ];
  export default products;
```

**src/ProductList.js**

```
import React from 'react';
import { Link } from 'react-router-dom';
import products from './productsData';

const ProductList = () => {
  return (
<div>
<h2>Product List</h2>
<ul>
        {products.map(product => (
<li key={product.id}>
<Link to={`/products/${product.id}`}>{product.name}</Link>
</li>
        ))}
</ul>
</div>
  );
};

export default ProductList;
```

**src/ProductDetail.js**

```
import React from 'react';
import { useParams } from 'react-router-dom';
import products from './productsData';

const ProductDetail = () => {
  const { id } = useParams();
  const product = products.find(p => p.id === parseInt(id));

  if (!product) {
    return <div>Product not found</div>;
  }

  return (
<div>
<h2>Product Detail</h2>
<p>Name: {product.name}</p>
<p>Description: {product.description}</p>
</div>
  );
};

export default ProductDetail;
```

**src/App.js**

```
import React from 'react';
import { BrowserRouter as Router, Route, Link, Routes } from 'react-router-dom';
import ProductList from './ProductList';
import ProductDetail from './ProductDetail';

function App() {
  return (
<Router>
<div>
<nav>
<ul>
<li>
<Link to="/products">Products</Link>
</li>
</ul>
</nav>
</div>
<Routes>
<Route path="/products" element={<ProductList />} />
<Route path="/products/:id" element={<ProductDetail />} />
</Routes>
</Router>
  );
}
export default App;
```

## DYNAMIC ROUTING

**Dynamic routing in React allows you to create routes that change based on user input or data. It's about showing different content or components depending on the URL .**Dynamic routing in React refers to the ability to create and manage routes that change based on user input, data, or other dynamic factors. React is a JavaScript library commonly used for building single-page applications (SPAs), and dynamic routing is a fundamental aspect of creating interactive and flexible web applications. Dynamic routing allows you to respond to user interactions and display different content or components based on the current route or URL parameters.

src/product.Data.js

```
constproducts = [
  { id:1, name:"Product 1", description:"Description for Product 1" },
  { id:2, name:"Product 2", description:"Description for Product 2" },
  { id:3, name:"Product 3", description:"Description for Product 3" },
  { id:3, name:"Product 3", description:"Description for Product 3" },
  { id:3, name:"Product 3", description:"Description for Product 3" },
  { id:3, name:"Product 3", description:"Description for Product 3" },
  { id:3, name:"Product 3", description:"Description for Product 3" },
  { id:3, name:"Product 3", description:"Description for Product 3" },

 ];
 exportdefaultproducts;
```

src/ProductList.js

```
importReactfrom'react';
import{ Link } from'react-router-dom';
importproductsfrom'./productsData';

constProductList = () => {
 return (
  <div>
   <h2>Product List</h2>
   <ul>
    {products.map(product=> (
     <likey={product.id}>
      <Linkto={`/products/${product.id}`}>{product.name}</Link>
     </li>
    ))}
   </ul>
  </div>
 );
};
```

```
exportdefaultProductList;
```

src/ProductDetail.js

```
importReactfrom'react';
import{ useParams } from'react-router-dom';
importproductsfrom'./productsData';

constProductDetail = () => {
  const{ id } = useParams();
  constproduct = products.find(p=>p.id === parseInt(id));

  if(!product) {
    return<div>Product not found</div>;
  }

  return (
    <div>
      <h2>Product Detail</h2>
      <p>Name: {product.name}</p>
      <p>Description: {product.description}</p>
    </div>
  );
};

exportdefaultProductDetail;
```

src/App.js

```
importReactfrom'react';
import{ BrowserRouterasRouter, Route, Link, Routes } from'react-router-dom';
importProductListfrom'./ProductList';
importProductDetailfrom'./ProductDetail';

functionApp() {
  return (
    <Router>
      <div>
        <nav>
          <ul>
            <li>
              <Linkto="/products">Products</Link>
            </li>
          </ul>
        </nav>
      </div>
      <Routes>
        <Routepath="/products"element={<ProductList/>}/>
        <Routepath="/products/:id"element={<ProductDetail/>}/>
      </Routes>
    </Router>
  );
}

exportdefaultApp;
```