

React JS Training

(Intermediate to Advanced)



Narasimha

Sr. Corporate Trainer, Mentor
tnrao.trainer@gmail.com

Schedule for React JS Training

Day#	Date	Topic
Day-1	16-Apr-2024	State Management : Class Components
Day-2	17-Apr-2024	State Management : Functional Components, Hooks
Day-3	18-Apr-2024	Http Client Programming – AJAX Calls to APIs (Node JS)
Day-4	19-Apr-2024	Working with Forms, Validations, Services
Day-5	22-Apr-2024	Redux – State Management Library
Day-6	23-Apr-2024	Routing – SPA in React JS
Day-7	24-Apr-2024	Unit Testing React Application

Day6

Routing – SPA in React JS



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day6

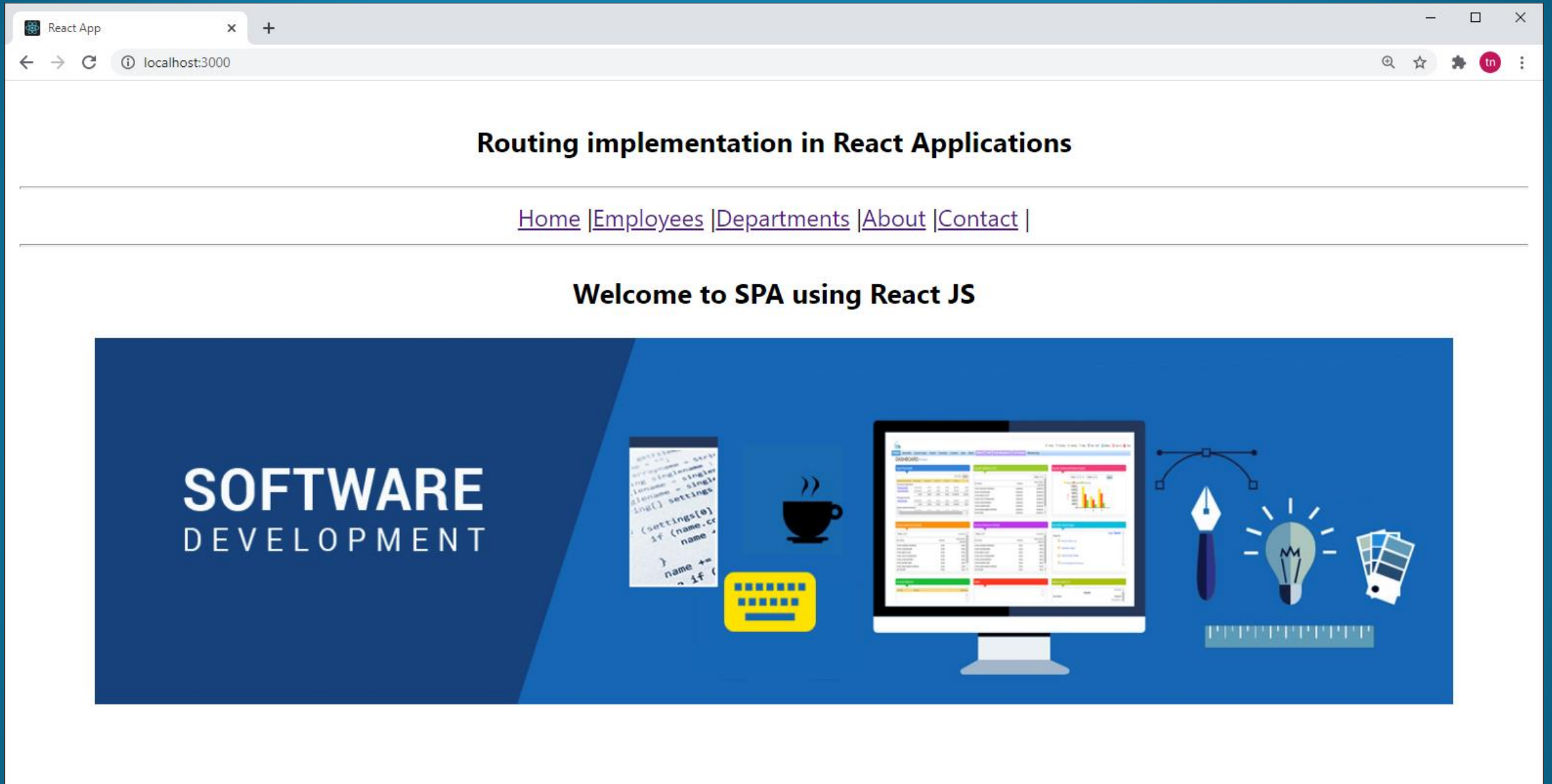
1. Implement SPA using React Routing
2. Navigations in React
3. Configure Routes in React Apps
4. Route Links in React Apps
5. Handling Invalid Routes
6. Handling Route Parameters
7. Handling Query String Parameters
8. Authenticate the Routes
9. Handling Authentication Tokens
10. Redirect to another route using code

Single Page Applications (Routing – Navigations)



Narasimha

Sr. IT Trainer/Consultant

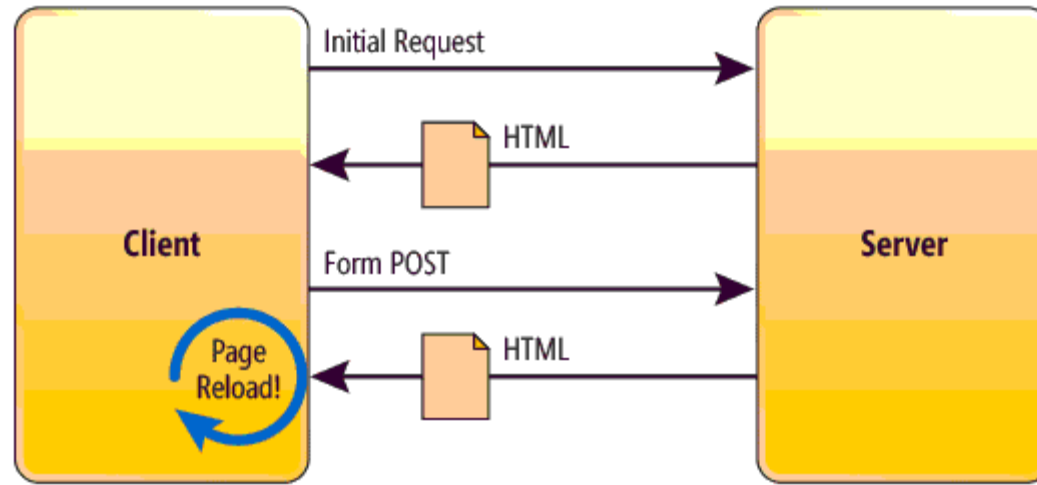


What is SPA?

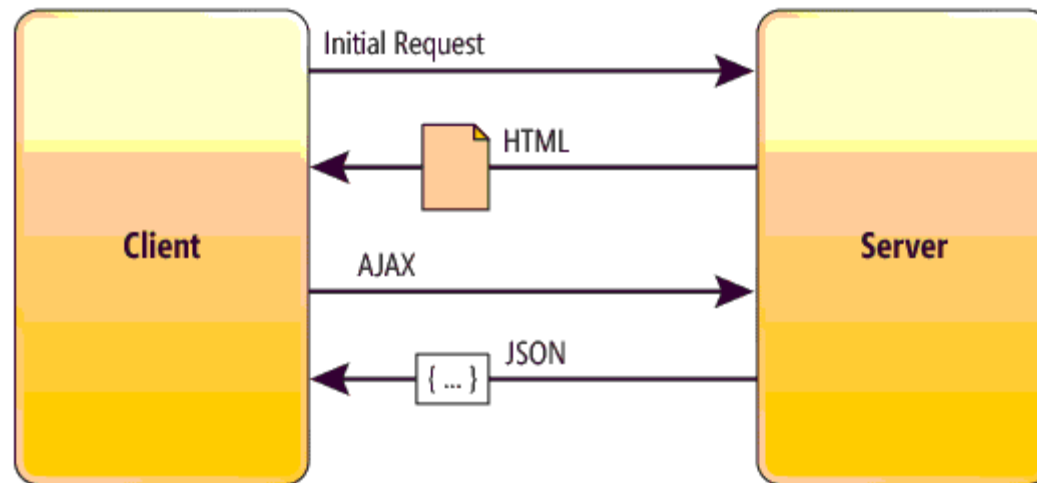
- A single-page application (SPA) is a web application or website that interacts with the user by dynamically rewriting the current web page with new content.
- It updates partial, instead of loading entire new pages
- The goal is faster transitions that make the website feel more like a desktop app.

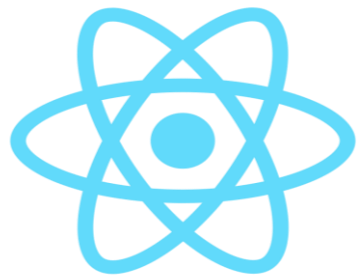


Traditional Page Lifecycle



SPA Lifecycle





Implement SPA in React

Routes in React

Narasimha

Sr. IT Trainer/Consultant

Steps to Implement

1. Install `react-router-dom` package
2. Import the required items in `index.js`
3. Configure the routes with corresponding components
4. Preparing Hyperlinks using `Link` tag
5. Provide routing details in `root.render()` method.

`index.js`



Implementation

Install react-router-dom module

```
npm install react-router-dom --save
```

Import the required items in index.js

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



Configure the routes and Prepare Hyperlinks

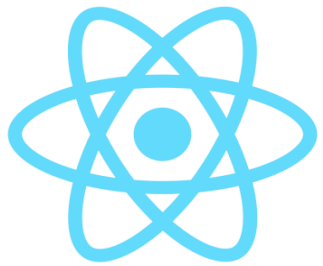
```
const routing = (  
  <Router>  
    <Link to="/">Home</Link>  
    <Link to="/Emps">Employees</Link>  
  
    <Routes>  
      <Route path="/" element= { <App /> } />  
      <Route path="/Emps" element= { <Emps /> } />  
      ....  
    </Routes>  
  </Router>  
)
```



Routing details in ReactDOM.render()

```
const root =  
ReactDOM.createRoot(document.getElementById('root'));  
  
root.render(  
  <React.StrictMode>  
    {routing}  
  </React.StrictMode>  
);
```





How to handle invalid route?

Narasimha

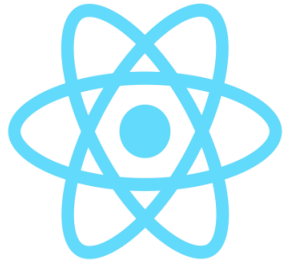
Sr. IT Trainer/Consultant

How to handle invalid route?

```
<Route path="*" element= { <NotFound /> } />
```

```
function NotFound(){  
  return <div>  
    .....  
  </div>;  
};  
export default NotFound;
```





How to handle route parameters?

Narasimha

Sr. IT Trainer/Consultant

How to handle route parameters?

```
<Route path="/Details/:id" element= { <Details /> } />
```

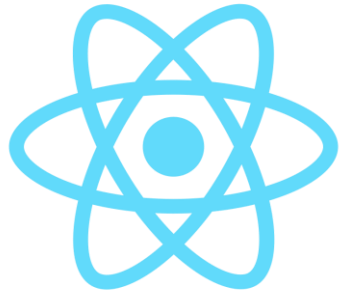
```
<Link to={"/Details" + item.Empno}>Details</Link>
```

```
import { Link, useParams } from 'react-router-dom';

function Details(props)
{
  const { id } = useParams();
  .....
}

export default Details;
```

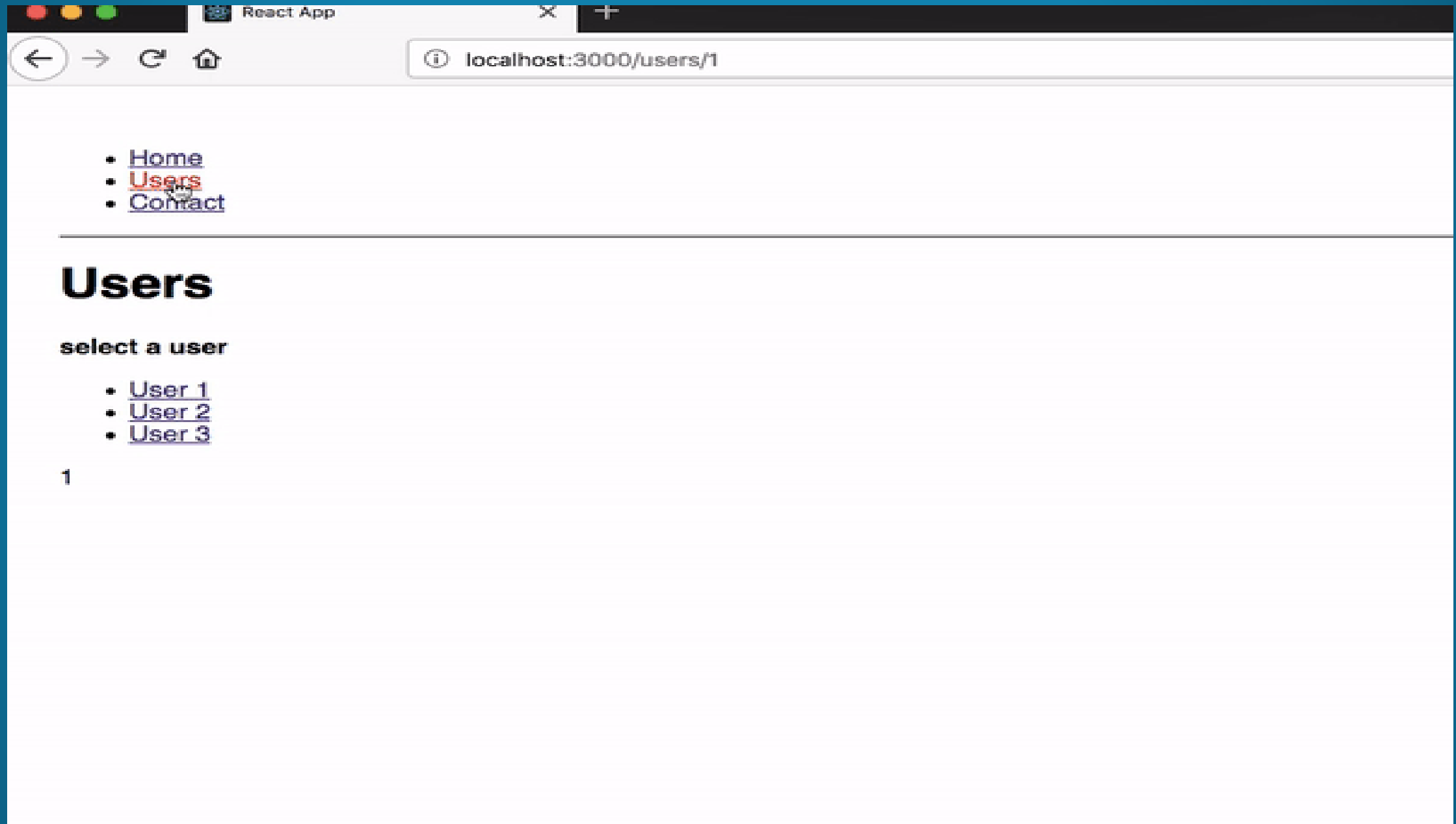


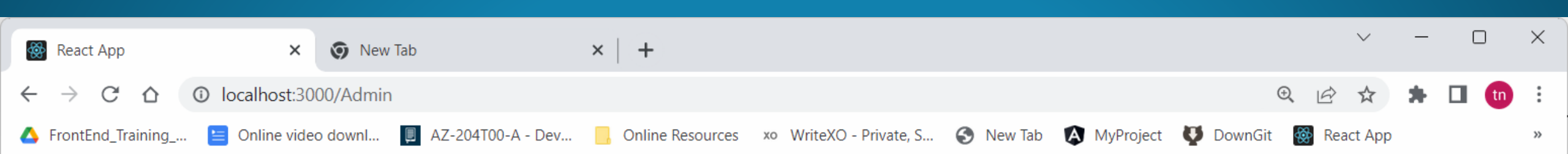


How to implement nested routing?

Narasimha

Sr. IT Trainer/Consultant





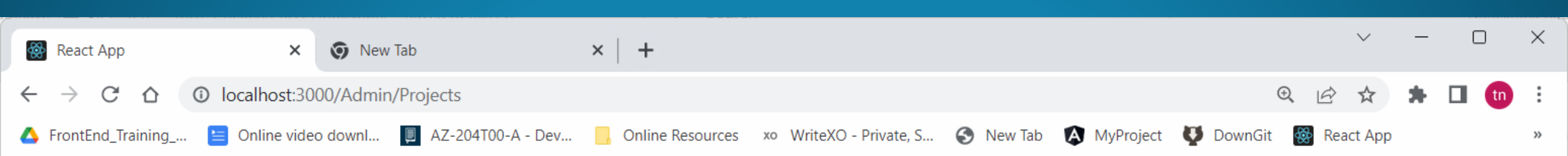
Routing implementation in React JS

[Home](#) | [Employees](#) | [Department](#) | [Contact Us](#) | [About Us](#) | [Invalid](#) | [Admin](#) |

This is Admin Component

[AdminHome](#) | [Projects](#) | [Customers](#) |

This is Admin Home Content



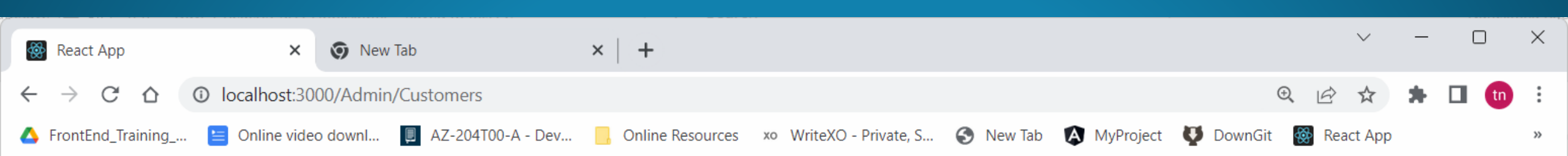
Routing implementation in React JS

[Home](#) | [Employees](#) | [Department](#) | [Contact Us](#) | [About Us](#) | [Invalid](#) | [Admin](#) |

This is Admin Component

[AdminHome](#) | [Projects](#) | [Customers](#) |

This is Projects Component



Routing implementation in React JS

[Home](#) | [Employees](#) | [Department](#) | [Contact Us](#) | [About Us](#) | [Invalid](#) | [Admin](#) |

This is Admin Component

[AdminHome](#) | [Projects](#) | [Customers](#) |

This is Customers Component

How to implement nested routing?

Index.js

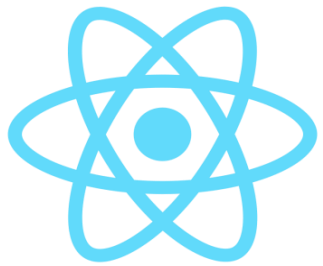
```
<Route path="/Admin" element= { <Admin /> }>  
  <Route index element= { <AdminHome /> } />  
  <Route path="Projects" element= { <Projects /> } />  
  <Route path="Customers" element= { <Customers /> } />  
</Route>
```



How to implement nested routing?

```
function Admin()  
{  
  return (  
    <div>  
      <h3>This is Admin Component</h3>  
      <Link to="">Admin Home</Link> |  
      <Link to="Projects">Projects</Link> |  
      <Link to="Customers">Customers</Link> |  
      <Outlet />  
    </div> );  
}
```





How to navigate to route using
code?

Narasimha

Sr. IT Trainer/Consultant

Programmatic Navigations

```
import { useNavigate } from "react-router-dom";  
let navigate = useNavigate();  
navigate("/Emps");
```

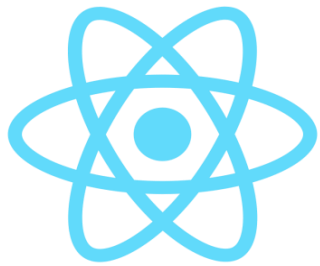


Practice Hands-Ons

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com



How to protect the routes?
How to configure authenticated
routes?

Narasimha

Sr. IT Trainer/Consultant

Authenticated Routes

- Configure the required routes with protection
- Handling Protected Routes
 - a. Verify the authentication token
 - b. Not Authenticated : Redirect to Login Route
 - c. Authenticated : Redirect to required route
- Login Component
 - a. Verify the user credentials
 - b. Valid : Create token, Redirect to required route



Dependent Concepts

- Redirect to routes programmatically
- Storing tokens using **Web Storage** (HTML5 API)
 - sessionStorage
 - localStorage
- Reading **route** parameters
- Reading **query string** params
- Usage of **props.children**



Practice Hands-Ons

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com



Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

