

# REACT ROUTER

Teaching Faculty: Umur INAN

# REACT ROUTER

- React Router is a standard library for routing in React.
- It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

# BROWSER ROUTER

- BrowserRouter is a router implementation that uses the HTML5 history API(pushState, replaceState and the popstate event) to keep your UI in sync with the URL.
- It is the parent component that is used to store all of the other components.

# ROUTES

- It's a new component introduced in the v6 and a upgrade of the component.
- Routes are chosen based on the best match instead of being traversed in order.

# ROUTE

- Route is the conditionally shown component that renders some UI when its path matches the current URL.
- Supply the component to be rendered to the Route component as element props.
- We can also supply custom props to the components in each route that we wish to render.

## ROUTE

- If you want to display a 404 page when the path does not match with any of the routes, then you can define a route with path as \*.
- The order of Route components does not matter.

# ROUTE

```
<Routes>  
  <Route path="/users" element={<UserComponent />} />  
  <Route path="/movies" element={<Movies genre="Action" age="13" />} />  
</Routes>
```

## DYNAMIC ROUTE

```
<Routes>
  <Route path="/users" element={<UserComponent />} />
  <Route path="/movies" element={<Movies genre="Action" age="13" />} />
  <Route path="/users/:id" element={<User />} />
</Routes>
```



# NOT FOUND

```
<Routes>
  <Route path="/users" element={<UserComponent />} />
  <Route path="/movies" element={<Movies genre="Action" age="13" />} />
  <Route path="/users/:id" element={<User />} />
  <Route path="*" element={<NotFound />} />
</Routes>
```

## NESTED ROUTE

```
<Routes>
<Route path="/" element={<Home />} />
  <Route path="/students">
    <Route index element={<StudentList/>} />
    <Route path=":id" element={<Student />} />
    <Route path="new" element={<NewStudent />} />
  </Route>
<Route path="*" element={<NotFound />} />
</Routes>
```

# USENAVIGATE

- Redirects to another route.

```
const App = () => {  
  const navigate = useNavigate();  
  const onClick = () => {  
    navigate("/home");  
  }  
  return (  
    <div>  
      <button onClick={onClick}>Go Home</button>  
    </div>  
  );  
}
```

## LINK

- Link component is used to create links to different routes and implement navigation around the application.
- It works like HTML anchor tag.
- The `to` property will determine the URL to which the user needs to be navigated.

```
<Link to="/">Profile</Link>
```

## USEPARAMS

- To access the URL parameters of the current route.
- Returns an object that maps the names of URL Parameters to their values in the current URL.

```
let params = useParams();
```

```
params.id
```

```
params.title
```

## MAIN POINTS

- All websites have something in common: they are made of pages that share similar structures.
- A facet of SCl is that Order is found everywhere.
- Transcendental consciousness leads to increased adaptability and flexibility.