REACT FUTURE: REFACTORING THE ROUTES WITH ROUTE OBJECTS

- Centralized **route configuration by using the** `routes.js` file.
- Representing routes as objects provides a more structured and data-driven approach to defining your routing.
- We will use the `useRoutes` hook from `react-router-dom` to take an array of route objects and create route elements.
- We will add **`index: true`:** to specify the default child route when the parent route (`/` in this case) is matched.

1. Create a `routes.js` file

- Create a new file named `routes.js` to hold your route objects.

2. Define the Route Objects

- Translate the JSX `<Route>` elements into JavaScript objects with `path` and `element` properties.
- For nested routes, use the `children` property to represent the nested structure.

```
// routes.js
import Home from './Home';
import About from './About';
import Contact from './Contact';
import Team from './Team';
import Profile from './Profile';
import Layout from './Layout';

const routes = [
  {
    path: '/',
    element: <Layout />,
    children: [
      {
        index: true,  // For the default route at '/'
        element: <Home />
```

```
      },
      {
        path: 'about',
        element: <About />,
        children: [
          {
            path: 'team',
            element: <Team />
          }
        ]
      },
      {
        path: 'contact',
        element: <Contact />
      },
      {
        path: 'profile/:userId',
        element: <Profile />
      },
    ]
  }
];

export default routes;
```

3. Update `App.js`

- Import the `routes` array and the `useRoutes` hook from `react-router-dom`.
- Use `useRoutes(routes)` to create your route elements.

```
import React from 'react';
import { BrowserRouter as Router } from 'react-router-dom';
import { useRoutes } from 'react-router-dom';
import routes from './routes';

function App() {
```

```
  const element = useRoutes(routes);
  return
 (
    <Router>
      {element}
    </Router>
  );
}

export default App;
```