

# JOBSHEET 4

Omar Al-Maktary  
TI-3H 1941720237

Experiment:

1- Create a new react project:



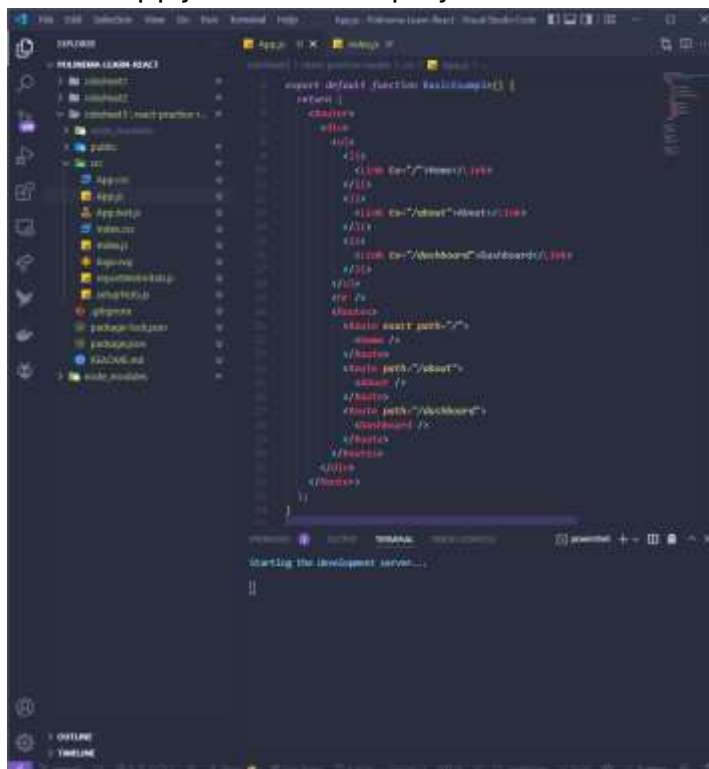
```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <h1>Hello React!</h1>
    </div>
  );
}

export default App;
```

npm start  
Starting the development server...

2- Edit file App.js and run the project:



```
import ReactDOM from 'react-dom';

ReactDOM.render(
  <div>
    <h1>Hello React!</h1>
  </div>,
  document.getElementById('root')
);
```

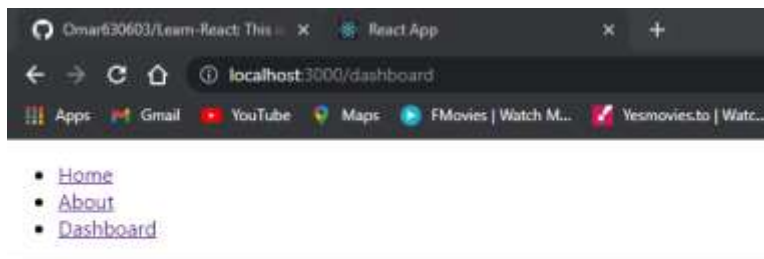
npm start  
Starting the development server...

- 3- Edit the routes because of an error that Route tag can't have Elements inside it so we use the new way:

```
<Routes>
  <Route exact path="/">
    <Home />
  </Route>
  <Route path="/about">
    <About />
  </Route>
  <Route path="/dashboard">
    <Dashboard />
  </Route>
</Routes>
```

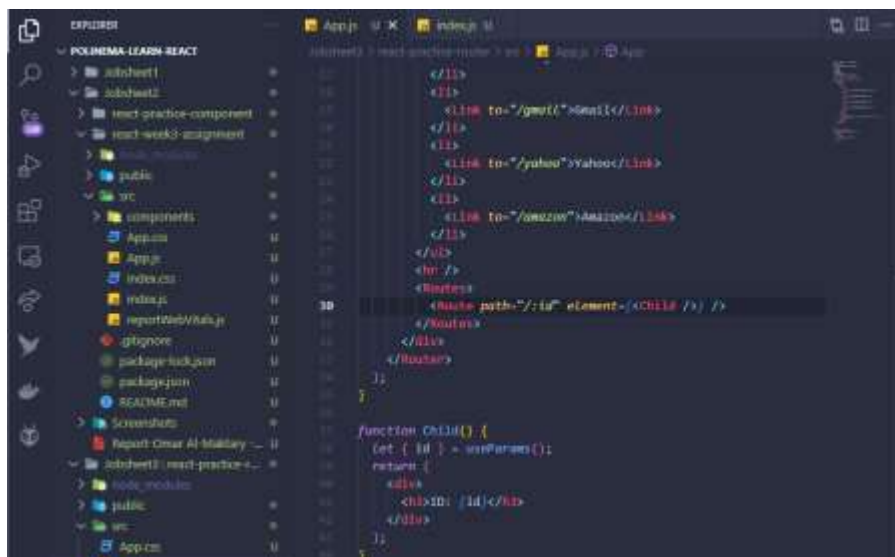
```
<Routes>
  <Route exact path="/" element={<Home />}></Route>
  <Route path="/about" element={<About />}></Route>
  <Route path="/dashboard" element={<Dashboard />}></Route>
</Routes>
```

- 4- Results:

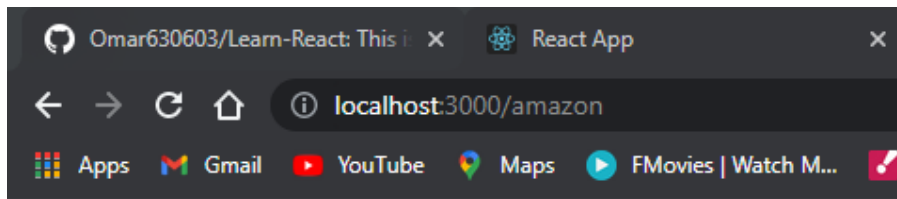


## Dashboard

- 5- URL Parameters



6- Results:



- [Netflix](#)
  - [Gmail](#)
  - [Yahoo](#)
  - [Amazon](#)
- 

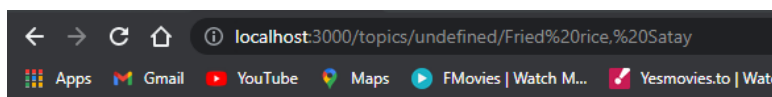
## ID: amazon

7- Use Nesting Router:

The method in the jobsheet doesn't work so I did some search and this is the results:



8- Results:



- [Home](#)
  - [Topics](#)
- 

## Topics

- [Food](#)
- [Travelling](#)
- [Review](#)

## Fried rice, Satay

9- Use Redirects (Auth):

In this experiment we learn how to configure authentication in ReactJS.

There have been some changes to the code since I'm using react v6:

```
import React from "react";
import {
  BrowserRouter as Router,
  Link,
  Route,
  Routes, //in the new router-dom version, Switch is used
  Navigate, //In older versions, Redirect is used
  useNavigate, //In older versions, useHistory is used
  useLocation,
} from "react-router-dom";

export default function AuthExample() {
  return (
    <Router>
      <div>
        <AuthButton />
        <ul>
          <li>
            <Link to="/">Home Page</Link>
          </li>
          <li>
            <Link to="/public">Public Page</Link>
          </li>
          <li>
            <Link to="/private">Private Page</Link>
          </li>
        </ul>
        <Routes>
          <Route path="/" element={<Home />}></Route>
          <Route path="/public" element={<PublicPage />}></Route>
          <Route path="/login" element={<LoginPage />}></Route>
          <Route
            path="/private"
            element={
              <PrivateRoute>
                <ProtectedPage />
              </PrivateRoute>
            }
          ></Route>
        </Routes>
      </div>
    </Router>
  );
};
```

```

}
function Home() {
  return (
    <div>
      <h2>Home</h2>
    </div>
  );
}

const fakeAuth = {
  isAuthenticated: false,
  authenticate(cb) {
    fakeAuth.isAuthenticated = true;
    setTimeout(cb, 100);
  },
  signout(cb) {
    fakeAuth.isAuthenticated = false;
    setTimeout(cb, 100);
  },
};

function AuthButton() {
  let navigate = useNavigate();
  let location = useLocation();
  let { from } = location.state || { from: { pathname: "/" } };
  return fakeAuth.isAuthenticated ? (
    <div>
      <h2>Welcome! </h2>
      <button
        onClick={() => {
          fakeAuth.signout(() => navigate("/", { from }));
        }}
      >
        Sign out
      </button>
    </div>
  ) : (
    <p>You are not logged in.</p>
  );
}

function PrivateRoute({ children }) {
  let auth = fakeAuth;
  let location = useLocation();

  if (!auth.isAuthenticated) {

```

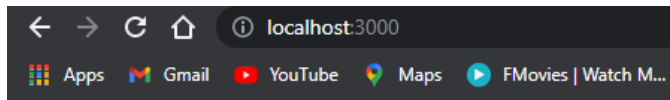
```

    return <Navigate to="/login" state={{ from: location }} />;
  }

  return children;
}
function PublicPage() {
  return (
    <div>
      <h2>Public</h2>
    </div>
  );
}
function ProtectedPage() {
  return (
    <div>
      <h2>Private</h2>
    </div>
  );
}
function LoginPage() {
  let navigate = useNavigate();
  let location = useLocation();
  let { from } = location.state || { from: { pathname: "/" } };
  let login = () => {
    fakeAuth.authenticate(() => {
      navigate("/private", { from });
    });
  };
  return (
    <div>
      <p>You must log in to view the page at {from.pathname}</p>
      <button onClick={login}>Log in</button>
    </div>
  );
}

```

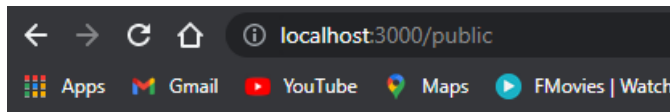
10- Results:



You are not logged in.

- [Home Page](#)
- [Public Page](#)
- [Private Page](#)

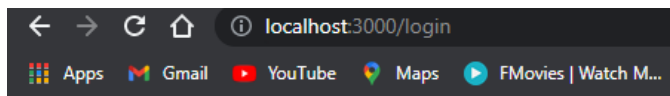
## Home



You are not logged in.

- [Home Page](#)
- [Public Page](#)
- [Private Page](#)

## Public

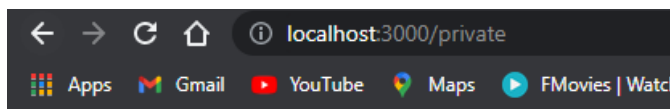


You are not logged in.

- [Home Page](#)
- [Public Page](#)
- [Private Page](#)

You must log in to view the page at /private

[Log in](#)



## Welcome!

[Sign out](#)

- [Home Page](#)
- [Public Page](#)
- [Private Page](#)

## Private