Child Routing In React

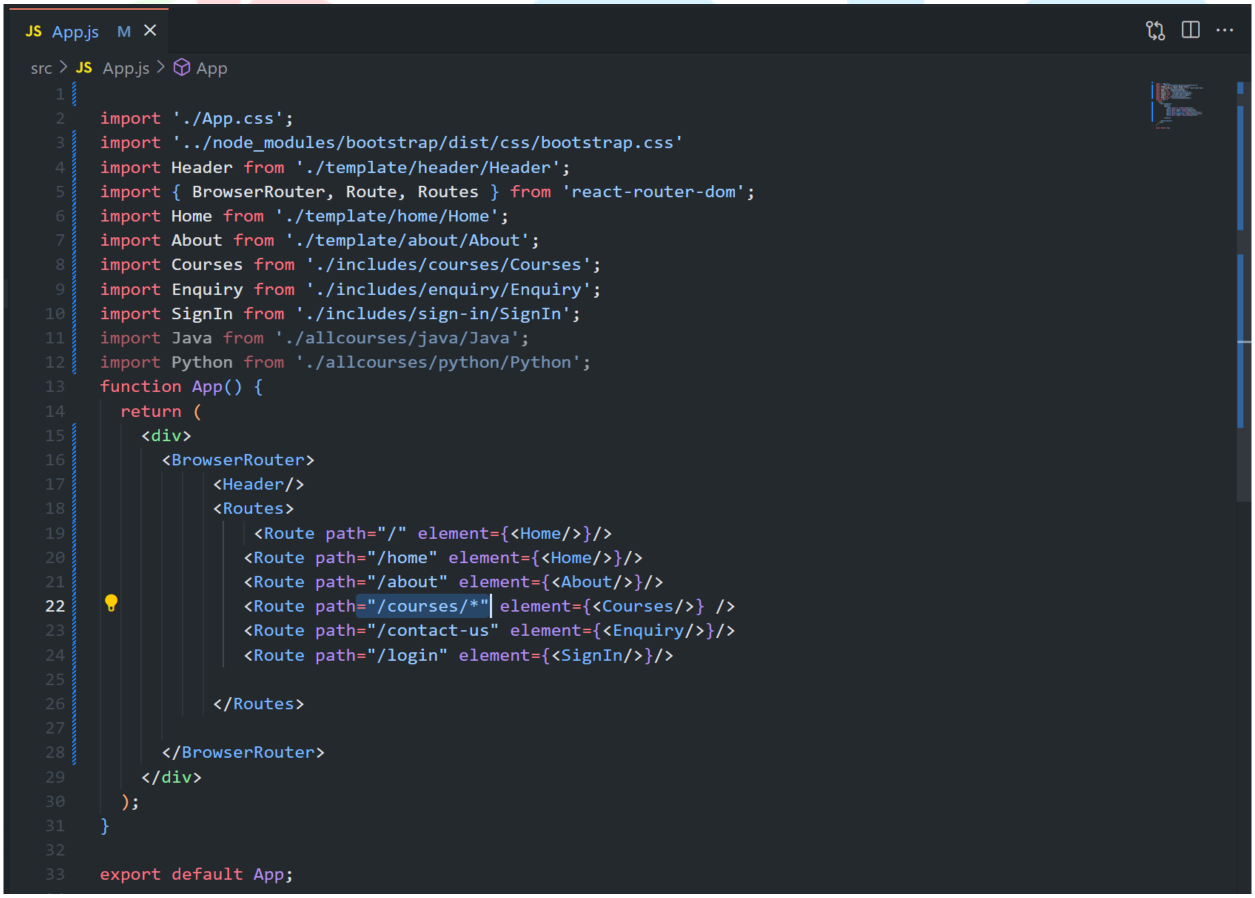
The term child routing refers to the concept on dynamically rendering the components on already dynamically rendered component.

In simple language child routing is nothing but navigation performed on existing route.

Example: -

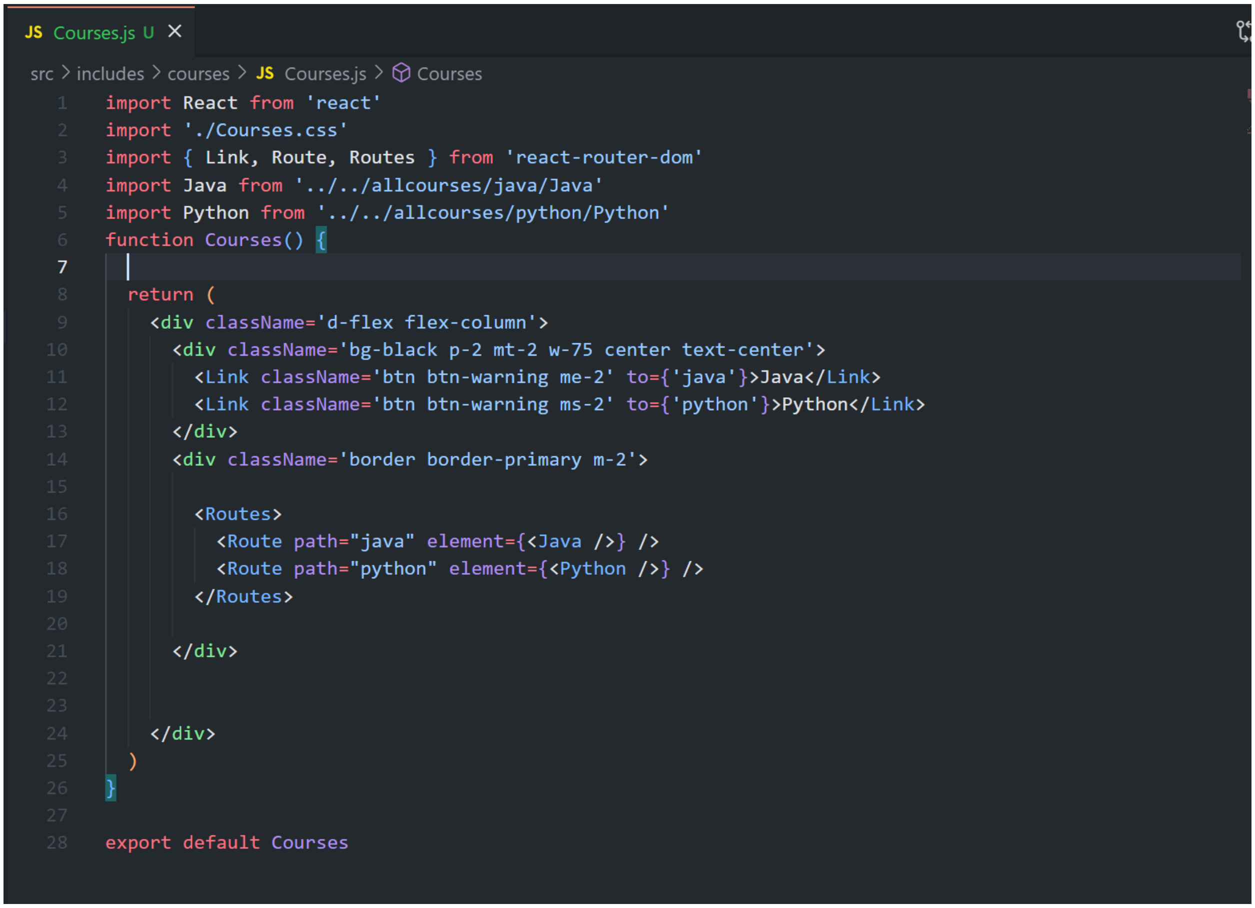
Step 1: - Create Java and Python component on your own.

Step 2: - Do below change in App.js



```

import './App.css';
import '../node_modules/bootstrap/dist/css/bootstrap.css'
import Header from './template/header/Header';
import { BrowserRouter, Route, Routes } from 'react-router-dom';
import Home from './template/home/Home';
import About from './template/about/About';
import Courses from './includes/courses/Courses';
import Enquiry from './includes/enquiry/Enquiry';
import SignIn from './includes/sign-in/SignIn';
import Java from './allcourses/java/Java';
import Python from './allcourses/python/Python';
function App() {
  return (
    <div>
      <BrowserRouter>
           <Header/>
           <Routes>
               <Route path="/" element={<Home/>}/>
              <Route path="/home" element={<Home/>}/>
              <Route path="/about" element={<About/>}/>
              <Route path="/courses/*" element={<Courses/>} />
              <Route path="/contact-us" element={<Enquiry/>}/>
              <Route path="/login" element={<SignIn/>}/>

           </Routes>

      </BrowserRouter>
    </div>
  );
}

export default App;
```

Step 3: - Do below changes in Courses.js



```
import React from 'react'
import './Courses.css'
import { Link, Route, Routes } from 'react-router-dom'
import Java from '../../allcourses/java/Java'
import Python from '../../allcourses/python/Python'
function Courses() {

  return (
    <div className='d-flex flex-column'>
      <div className='bg-black p-2 mt-2 w-75 center text-center'>
        <Link className='btn btn-warning me-2' to={'java'}>Java</Link>
        <Link className='btn btn-warning ms-2' to={'python'}>Python</Link>
      </div>
      <div className='border border-primary m-2'>

        <Routes>
          <Route path="java" element={<Java />} />
          <Route path="python" element={<Python />} />
        </Routes>

      </div>


    </div>
  )
}

export default Courses
```

Step 4: - check output.

React Component Lifecycle

The component lifecycle refers to the time period of component loads into the DOM to it removed from the DOM.

In simple language component lifecycle is the period of component rendered on browser window.

Component lifecycle has three phases, which are as below.

1. **Mount Phase: -**
   When the component started getting rendered into the DOM then that lifecycle phase is called as mount phase.

2. **Update Phase: -**
   When the value of state or props of a component get changed then at that moment component is in update phase of its life cycle. In this phase component gets rerender.

3. **Unmount Phase: -**

   When the component stated getting removed from the DOM, then that lifecycle phase is called as unmount phase.

CJC
Complete Java Classes

JSON-Server & JSON-Database

Q) What is JSON databases?

Json data based is use to store semi-structured data. json databased is not expensive to perform operation on it as it is a document (file) type database.

Q) What is JSON-Server?

JSON-Server is used in front-end development to achieve, fast development. As it removes the dependency from actual backed server.

Json-server reads the data of a given Json-database and accordingly generates several endpoints to perform the CRUD operation on it.

Imagine we have a db.json file, in this file we have courses information in Json data-format, then the endpoint which are generated by Json-Server are as below.

1. Post course to add/save new course.
2. Get course to get all courses.
3. Get course/{id} to get single course.
4. Put course/{id} to update particular course.
5. Patch course/{id} to partially update the particular course.
6. Delete course/{id} to delete the particular course.

Installation command: - **npm install json-server**

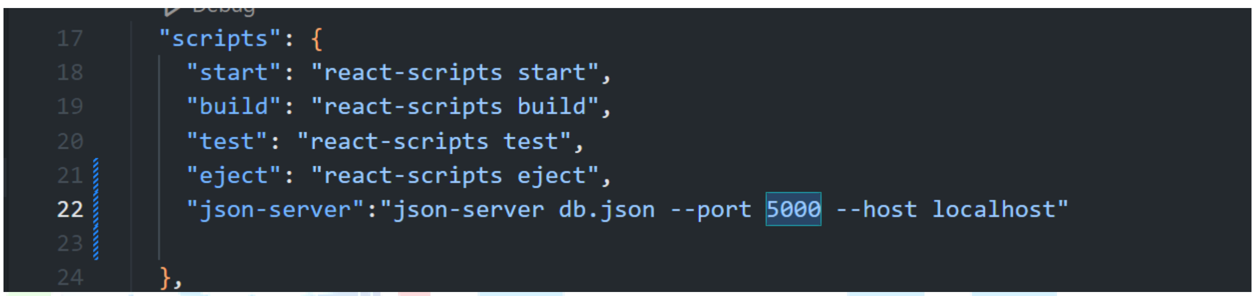
CJC Complete Java Classes

Q) How to run Json-server?

1st Way: - on terminal use below command.

```
Json-server –watch db.json –port 500
```

2nd Way: - in package.json add the script as shown below.



```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "json-server":"json-server db.json --port 5000 --host localhost"

},
```

CJC
Complete Java Classes