Build E-Commerece App

by React , TypeScript , Redux toolkit and BootStrap

**Features**

* **App Navigations**
* **Filtration**
* **Wishlist**
* **Shopping Cart**
* **Maximum add-to-cart**
* **Authentication**
* **Authorization**
* **Profile Page**
* **Place Order**
* **Skeletons and Animations**

**Typescript Experience**

* **Typescript with React**
* **TypeScript with Functions, Objects, Arrays, and**
* **Unions..etc**
* **Interfaces and Types**
* **Supporting Redux**
* **Generics**
* **Alias Paths**
* **Narrowing**
* **Tips and tricks**

**Technologies**

* **React**
* **Redux Toolkit**
* **Redux Persist**
* **Redux Selectors**
* **AXIOS**
* **React Router Dom**
* **React Hook Form**
* **ZOD Validation**

**Bounce**

* **Dark Mode**
* **Intersection Observer AF**
* **Multi-Language Support**
* **Toaster Notifications**

1. Check Node.js Version

A black background with white text

Description automatically generated

1. Install Vite Project (npm create vite@latest)

A screen shot of a computer program

Description automatically generated

1. npm I ( To install all backages )

Then You Can Check Your Project By This Command ( npm run dev )

Install Main Libraries

We Can install all Libraries in One line Code  
( npm i axios react-router-dom react-redux @reduxjs/toolkit react-bootstrap bootstrap )

A screenshot of a computer program

Description automatically generated

Folder Structure

We will delete this files  
A screenshot of a computer

Description automatically generated

We will Handle main.tsx , and delete <React.StrictMode> ,

because it causes double re-renders with useEffect

A screen shot of a computer

Description automatically generated

Then in SRC Directory we Will make

1-component Folder inside this Folder:

* common: we put here things wich don’t have category , Like (buttons , header , footer , sidebar) .
* feedback: we put here things like ( Toaster Notifications , Alert )
* forms: we put forms here like ( login , register )
* ecommerce:
* layout: Components of Layout like ()

2- layouts Folder inside this Folder:

( layouts like ( main layout , user layout ) )

3-routes Folder inside This Folder:

4-hooks folder inside This Folder:

5-pages folder inside This Folder:

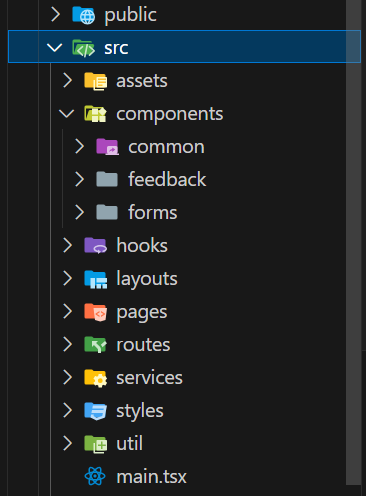
6-store folder inside This Folder: (I will use Redux so Project Global State)

7-services folder inside This Folder: (Global configuration by axios , api handler)

8- util folder inside This Folder :

( We need to create a Util when there is a function. We will use it a lot - to prevent repetition, we put it inside the Util and make it an import when needed )

9-styles folder inside This Folder : ( create global styles inside it )



Create your backend by JSON server

1. We will open Terminal and make a NewFolder (ecommerce-backend) in the main project folder
2. And open a Terminal and write ( npm init )

A screen shot of a computer

Description automatically generated

1. Then install json-server ( npm install json-server )

A screen shot of a computer

Description automatically generated

1. The open this folder in vs code , and make db.json file and put a dummy data in it

A screen shot of a computer program

Description automatically generated

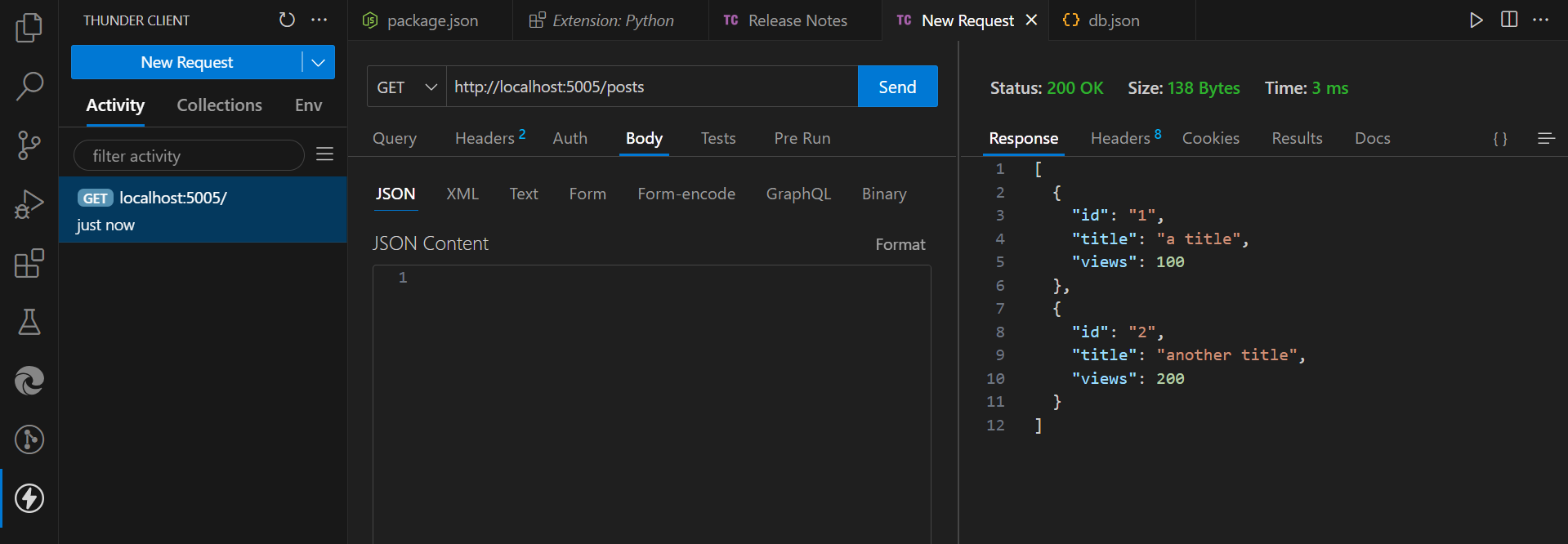
1. Then we’ll go to package json and handle npm start for this file (edit “Script” Key by adding “start” value)

A screen shot of a computer

Description automatically generated

1. Finally When I open terminal and write :

( npm start ) it will open a local-host page at port 5005 and this page will contain my data which in db.json



**Start The Project**

**Typescript aliases path:**

When I have these dots (extensions)

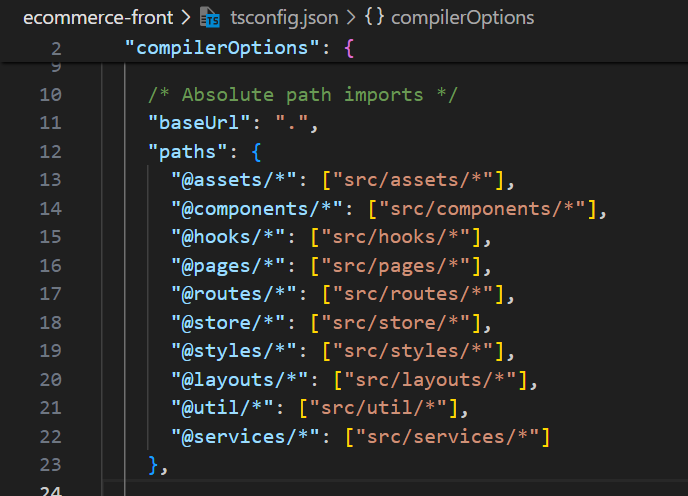


We should solve this problem by **aliases path** we will install ( npm I vite-tsconfig-paths )

•Then when I install Plugin We Should Go To ( vite.config.ts ) file and define it like This



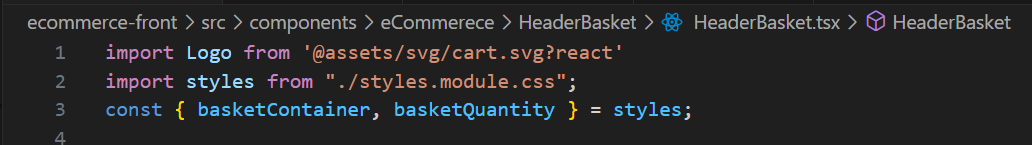
•Then we will sets the Paths in ( tsconfig.json ) file



‘ \* ’ here mean that all after the ‘ / ‘ , For Example:

Instead of [“ src/assets/svg/cart.svg ”] 🡺 [“ src/assets/\* ”]

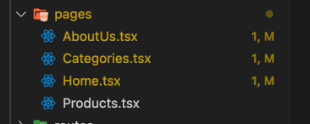
Congratulations  
We can now Do Paths like this



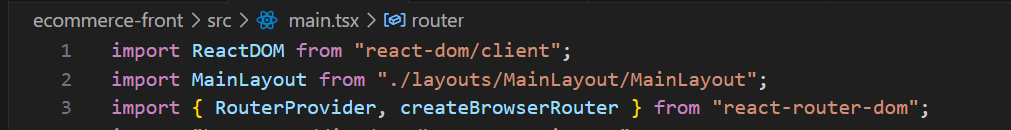


**Create main Routes**

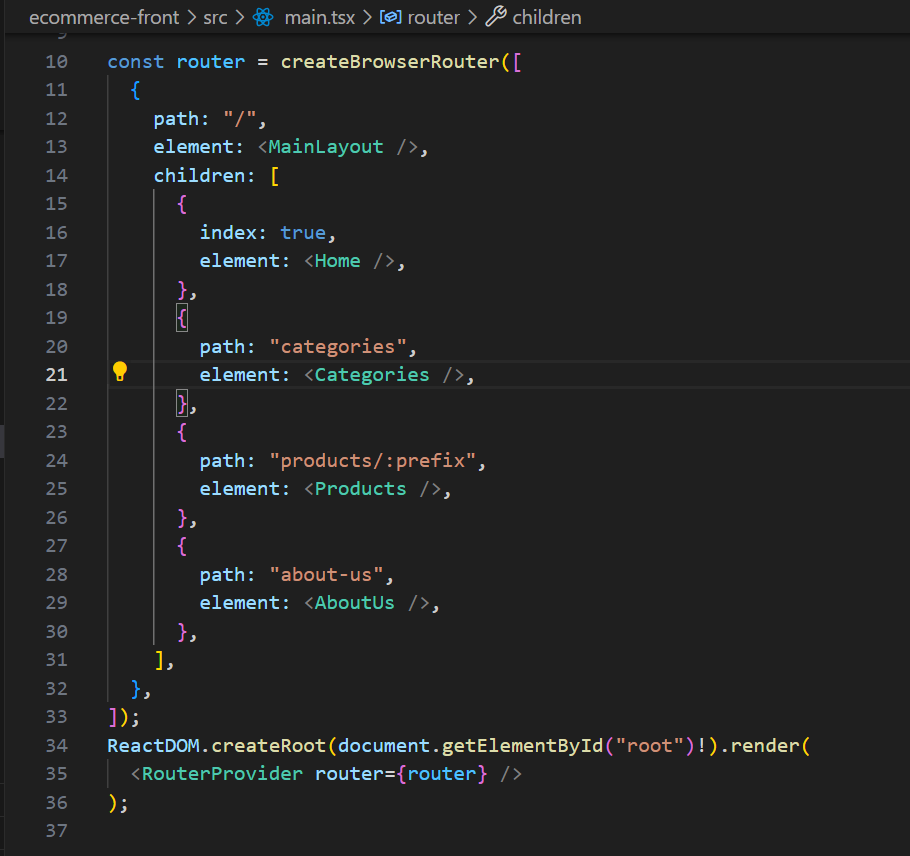
1. We Will Create Main Pages like ( Home , Products , Categories , AboutUs ) Accompained by “ .tsx “ extension in Pages Folder.



1. After Creation of Main Pages we will Call Two Important Function ( createBrowserRouter , RouterProvider ) in the File We will make the Route in ( main.tsx )



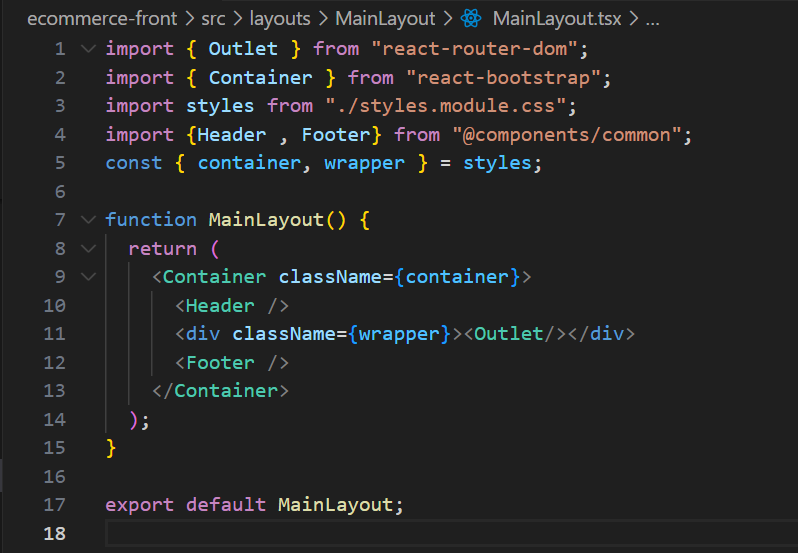
1. Then We will make a router variable and set createBrowserRouter with Array
2. We need to sets Layout that we need to render with its Paths like this



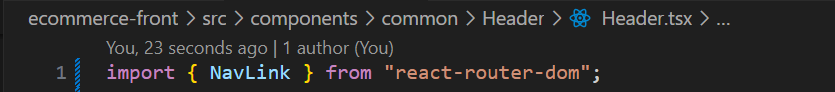
•Come To Show How we will Get the Pages that we route in the MainLayout Page

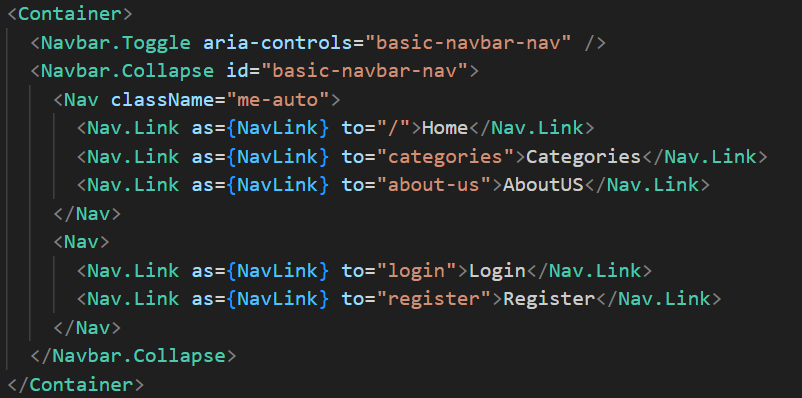
We will Import { OutLet } from "react-router-dom"

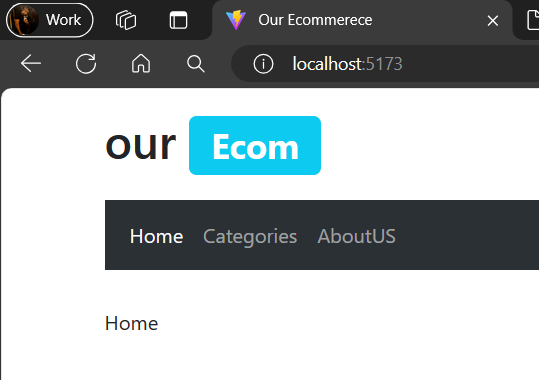
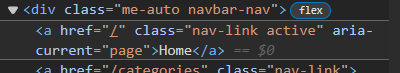
And it will Get The Page Directly



•Ok we need to Use NavLink from react-router-dom library because when we use it , It add an active class to the Link which we press



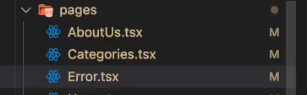


Then we can use this active class and make a special style for any link will be active

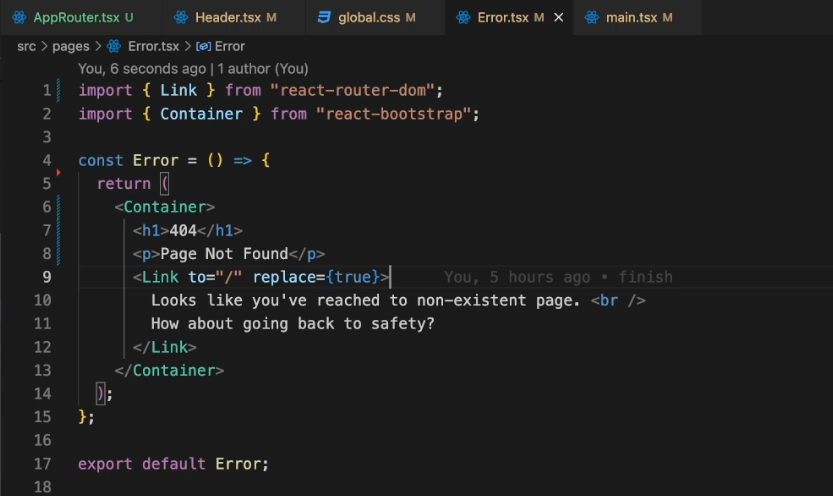
Create Error Page

•We Will create Error Page in Page Folder

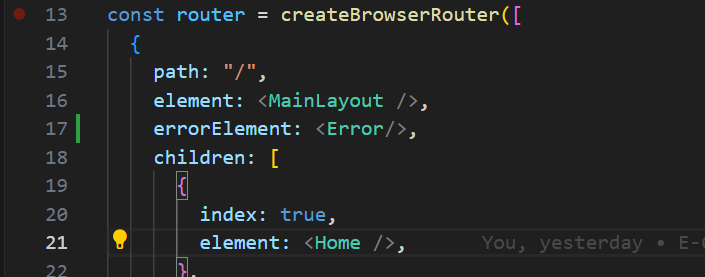


•Create The Page and We need To Create a Link When we press on it , we will go to the Home Page & cant return

* We will Link By { Link }
* And Can’t Return by: replace = { true } attribute



•We need now To Route This page with the components by: errorElement: < --- />



Client side router param guard

* We need to Protect Products Page , Because The Param that will be sent 🡪 must Be String
* So We need to Protect This Error From the Client Side and prevent any one sent any type of param except String
* We can use “ loader ” in the react-router-dom it can make me build a Condition or Guard and call API

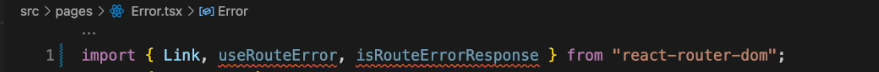


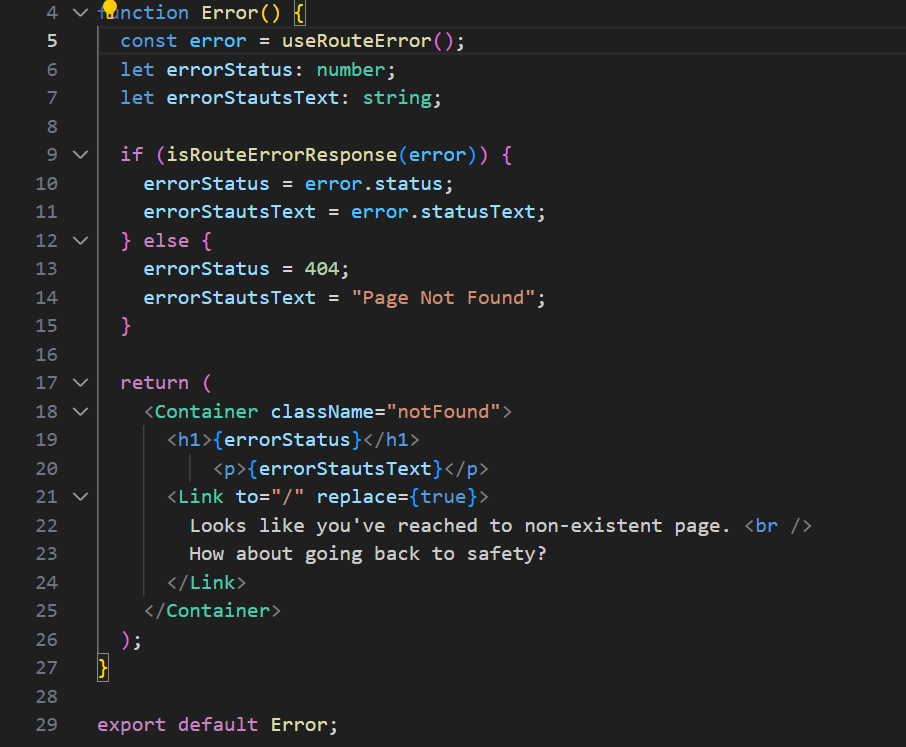
* Loader should return something (true) it mean show the page معناها كمل واظهر الصفحه
* We use regular expression code mean if the code cant contain letters from a-z the code will be error
* Throw new Response 🡪 react-router-dom
* Status:400 🡪 Bad Request
* Status:404 🡪 Not Found
* To avoid undefined error we write this line “ typeof params.prefix !== "string" || ” , Because ( params ) is an object and we cant check the object if it empty

React router dynamic error page

* We need to call Two parameters from react-router-dom:

1. useRouterError 🡪 لاستدعاء الايرور
2. isRouterErrorResponse 🡪 لو الايرور فعلا جايلي من React-router-dom





Category and Product components

* We create in Page Folder Two Files
  + Products.tsx
  + Categories.tsx

A screenshot of a computer

Description automatically generated

* Products.tsx 🡪 it’s a Page that will be map all the Products , and all the Products rendered in the component

A screenshot of a computer program

Description automatically generated

• <Col> & <Row> 🡺 Make Grid System

• Xs={6} 🡺 Support all screens ( big screens and mobile )

* Categories.tsx 🡪 it’s a Page that will be map all the Categories , and all the Categories rendered in the component

A screen shot of a computer

Description automatically generated

* We create Two Component in eCommerce Folder
  + Product.tsx
  + Category.tsx

A screenshot of a computer

Description automatically generated

* Category.tsx 🡪 it’s a Component include the name and image and title of every Category element will be import from database

A screen shot of a computer program

Description automatically generated

* Product.tsx 🡪 it’s a Component include the name and image and title of every Product element will be import from database

A screen shot of a computer program

Description automatically generated

<div> **Part 1** < **The End** /> <div/>