* First step in any project is (to build folder structure and instal [main] libraries)
* Because the team lead is the only one responsible, so he builds this structure and push it on (main branch) and then all team members drive there branches with the same structure.
* Put project Layouts (MainLayout, AuthLayout, ..)

Notes:-

* Remove (strict mode) from the main.tsx file
* Because it makes the (useEffect) executes many times
* Open a new terminal from frontend part -> cd .. –> cd Backend
* Folder that contains more than 1 component
* Starts with a (lowercase letter)
* (vite-plugin-svgr) is used to allow (svg) to be imported as a component inside vite projects
* (import + configure it) in the (vite config file)

Like => import svgr from "vite-plugin-svgr";

 plugins: [react(), svgr()],

* configure in (tsconfig.app.json)

Like => "types": ["vite-plugin-svgr/client"]

* (import "../../assets/logo.svg?react")

Put (?react) at the end of the import

* <Navbar> <li></li> <li></li> </Navbar>
* This is a custom header called navbar which can be used in multiple page (But each page has it own links)
* So I send links as (children) to the cusom navbar to render them there.
* Instead of make the parent component flex and align items there, I can align child component from inside it’s styles by (align-self: flex-end)
* \*\*\* To make (Footer) appears at the bottom of it’s parent component

1. Parent component =>

(display: flex; flex-direction: cloumn)

1. Footer (Child) => (margin-top: auto)

* vite-tsconfig-paths => Allows us to type (@)in paths instead of ../ or ./

(configure it like vite-plugin-svgr)

* All plugins installed on vite project have same

congiuration in (vite.config.ts, tsconfig.app.json)

((( but here tsconfig I put the routes I want )))

And (imported in the same way)

* < Nav.Link as={NavLink} to="/about-us">
* About
* </Nav.Link>

This code here means that (Nav.Link) is a component in (Reactstrap) and I want it (to act as [NavLik] component) in react-router-dom

* <NavLink

to=”/about-us”

key={index}

className={(navClass) =>

navClass.isActive ? "active\_\_menu" : ""

}

>

About Us

</NavLink>

(This (NavLink) component give (active) class for (the active route) (current route) and I can give this class some custom styles in my css file (or in global styles file)

* <Link to="/" replace={true}></Link>

Replace here make me when I click on back button on browser to not return to this page

\*\*(Because browser makes a stack of page LIFO so if I am in the page 3 on the stack and click back this returns me to page 2, and if I clicked forward i will return to page 3, here If I put {replace} on the link this removes page 3 from the stack to i can to return to this page)

(used usually with error pages)

* {
* path: "products/:prefix",
* element: <Products />,
* loader: ({ params }) => {
* if (
* typeof params.prefix !== "string" ||
* !/^[a-z]+$/i.test(params.prefix)
* ) {
* throw new Response("Bad Request", {
* statusText: "Category not found",
* status: 400,
* });
* }
* return true;
* },
* // Condition to confirm that the category prefix is a pure string, because there is not category with name of any other characters
* // So here we reduce backend calls to make the app more efficient
* // throw new Response() => This is how we return errors in react-router-dom (to navigate to dynamic error page)
* }
* if (
* typeof params.prefix !== "string" ||
* !/^[a-z]+$/i.test(params.prefix)
* ) {...}
* // Here I put a guard in typescript ( typeof params.prefix !== "string")
* // because in typescript (params.prefix) could be a string or undefined
* In eCommerce apps, it is not logic to call categories api every time I visit categories page  
  (Because it is not usual that categories are changing every day or every hour)

But it is logic to call products api each time because products are updated eventually

* Generics:-

Instead of building a cutom fuction that each time I pass the type of argument I pass (in ts)  
- I can build a gentric function that knows the type of argument (on executing not on building)

* // EX 1
* const getargs = <T,>(arg: T) => {
* return arg;
* };
* getargs<number>(5);
* getargs(5);
* // true also, because typescript is smart enough
* // to know the type of argument i pass
* // EX 2
* const getGenericArray = <T,>(list: T[]) => {
* return list;
* };
* getGenericArray(["string1", "string2"]);
* // EX 3
* type TList<T> = T[]; //type here (accepts an argument type)
* const getGenericArray = <T,>(list: TList<T>) => {
* return list;
* };
* // EX 4
* type TObject<T> = {arg: T}; //type here (accepts an argument type)

We type (T) here not (Type) because community used to do that, we type comma here <T,> to not make an error

* Extending an object:-
* type TObj = { id?: number; name: string; age: number };
* type hasID = { id?: number };
* const returnObject = <T extends hasID>(obj: T) => {
* return obj.id;
* };
* returnObject<TObj>({ id: 1, name: "testName", age: 30 });
* // if I remove (extends hasID) will be an error here , because it can not access and see (id) in object
* If I have a function in parent component, I want to execute it based on event (click) happens a child component
* // parent:-
* const User = () => {
* const deleteHandler = (id) => {
* // ....
* }
* <User delete={deleteHandler} />
* }
* // child:-
* const User = ({delete}) => {
* <div onClick={() => delete(id)}>delete</div>
* }
* Z
* Z
* Z
* z

Commands:-

npm i typescript -g

npm create vite@latest

npm install

npm i axios

npm i react-router-dom

npm i react-redux

npm i @reduxjs/toolkit

npm i react-bootstrap

npm i bootstrap

npm i vite-plugin-svgr

npm i vite-tsconfig-paths

* npm init - npm i json-server

Technologies:-

React

Typescript

Axios

React-router-dom

React-redux

Redux toolkit

React bootstrap

Bootstrap

vite-plugin-svgr

vite-tscongig-paths

-- json-server

Questions:-

1. What is (SWC), what is the difference between it and (Babel)