Rafce react development for creating the functional component

Complicated server setup

Handling user auth

Scaling

Worrying about the security

File management and optimizing the api performance

React and react routing with the outlet and conditional rendering

React query and the tan stack 5.0

These will allow for the auto caching

Refatching

Parallel queries

First class mutations

Loading and the state management

Tailwind and shat cn

Appwrite for the back end as the serveice tool and type script

Auth is for the team based and the permission based security

Add one or more than databases for a particular project.

Intialise the 5react application 🡪 npm create vite@latest ./

./ allows you to open inside the directory

Createvite 🡪 yes

React

Npm install 🡪 to install all the dependencies

Npm run dev 🡪 open the thing in the local development environment

Creates a simple folder structure for the application

The folder structure

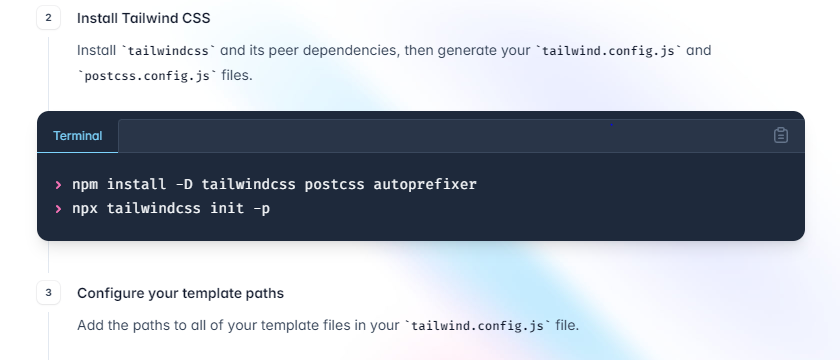
* Public
* 🡪 assets
* src (source directory)
* 🡪 \_auth
* 🡪🡪 AuthLayout.tsx
* 🡪🡪forms
* 🡪 🡪🡪 SigninForm.tsx
* 🡪🡪🡪 SignupForm.tsx
* 🡪 \_root
* 🡪app.css
* 🡪 app.tsx (will be the primary router component)
* 🡪 index.css
* 🡪 main.tsx (starting point of our application)
* 🡪 vite-env.d.ts
* Gitignore file

Create the main.tsx filw where thje routing for the other compenents are created

Main.tsx

App.tsx is created and the globals.css file is created for the styling

Here we are going to use v3.4.3 vite search inside the tailwind website

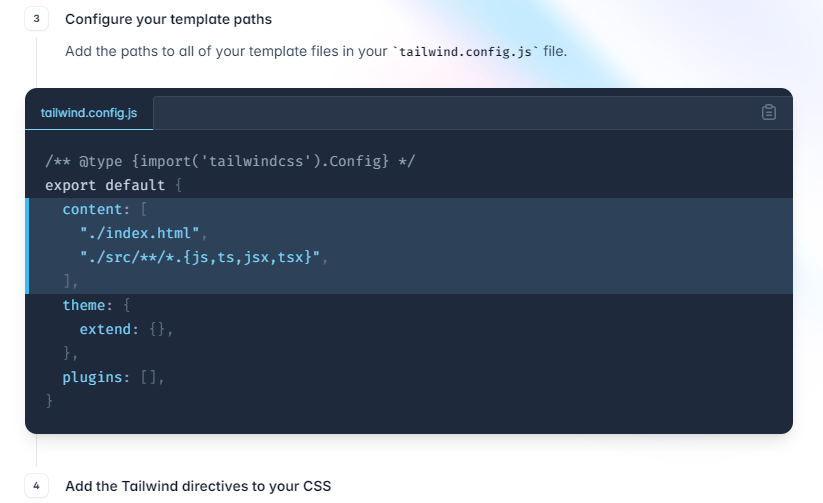


The following 2 commands are to be run

First will install all the required depencies

Second will initialize the files that are required for the styling

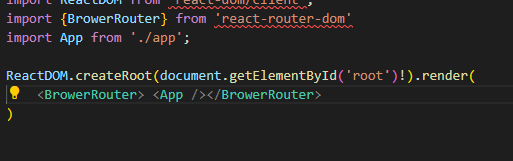
Configure the path



Npm install -D tailwindcss-animate

After all the setup is done the globals css style sheet is applied to the rest of the app

Pacakage that is neede for managing the routes is npm install react-router-dom



By wraping the app inside the browser router makes the app able to route

This section is going to go onto the app.tsx

Public routes

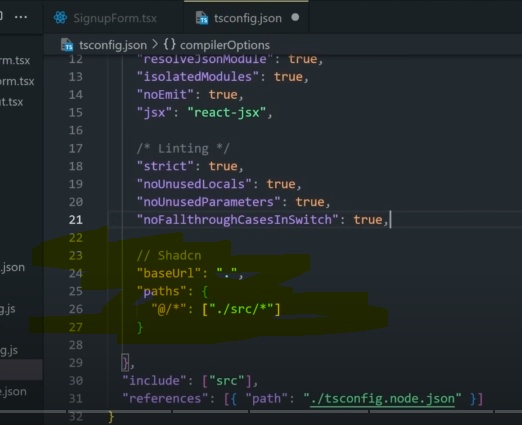
these are the sign up and the sign in and the sign up that every body are able to see

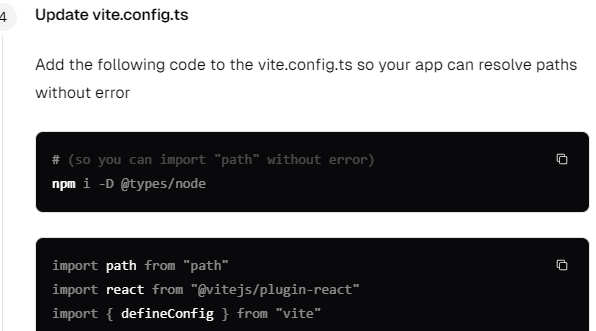
private routes are the routes that we are able to see only if we are able to sign in

inside the src directory we use \_auth and the \_root

AuthLayout will wrap the login page and the image at the side

Tsconfig.json





This will make the path errors go away

We are overwriting the vite.config.js

