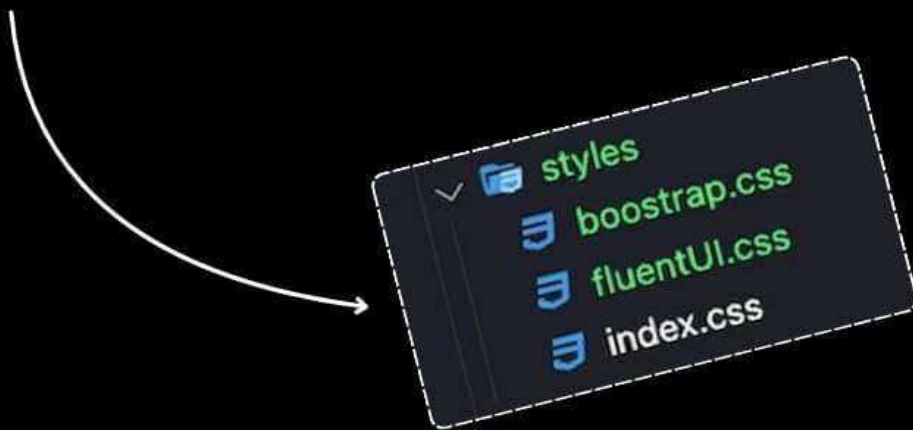


@uvais codes

How to Structure a React App



Swipe>>



Example structure

The structure of a project is very opinionated and there isn't really a right or wrong way as long as it's organized. However, having a set structure can be easier for beginners, so I will share the way I like to structure my projects just to get you started.



Assets

This folder will contain any static assets your app requires. Including images, fonts, PDFs, etc.



Style










The styles folder contains style css of the project. Here we can store component-based custom styles.



```
▼ styles
  bootstrap.css
  fluentUI.css
  index.css
```

Pages

Here you will keep all pages that correspond to a URL. This will make finding code for a particular page a little easier.

```
✓  pages
  >  dashboard
  ✓  product
    ✓  product-config
       index.ts
       ProductConfig.tsx
    ✓  product-information
       index.ts
       ProductInformation.tsx
```



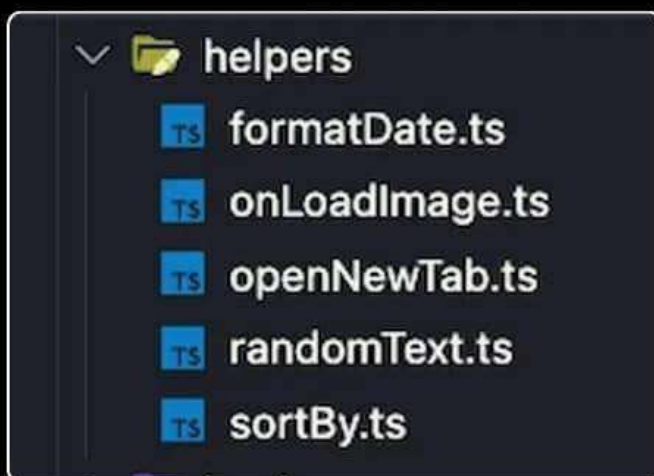
Components

Components are the building blocks of any react project. This folder contains collections UI components like button, card, modal, input, etc..., that can be used across various files in the project.



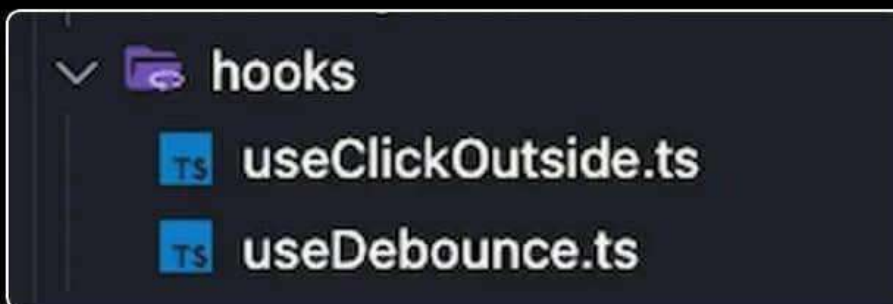
Helpers

The helpers folder contains utilities functions. You should be store pure functions on this folder. This mean is each function will have its own responsibilities.



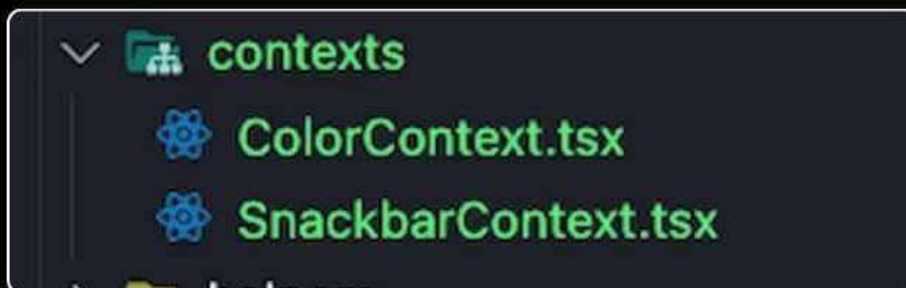
Hooks

The hooks folder contains the hooks function to can reuse code logic for all components, and it uses to separate between code logic and your components.



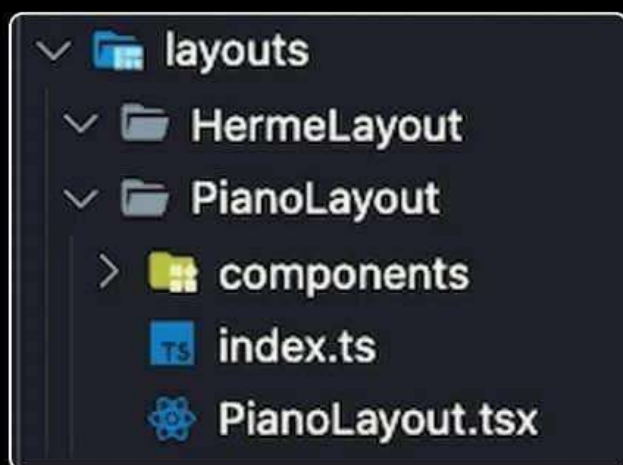
Contexts

The contexts folder stores all your React context files that used to share states between components each other. There are large projects you will have multiple context you use across your application.



Layouts

The layouts folder contains dynamic layout that you want display based on your client's information. This would be things like a sidebar, navbar, etc



Overview

```

└─ src
  └─ .jest
  └─ components
    └─ Card
    └─ Table
    └─ config
  └─ helpers
    └─ onLoadImage.ts
    └─ sortBy.ts
  └─ hooks
    └─ useClickOutside.ts
    └─ useDebounce.ts
  └─ layouts
    └─ HermeLayout
    └─ PianoLayout
  └─ libs
  └─ pages
    └─ dashboard
    └─ product
      └─ product-config
      └─ product-information
  └─ styles
```

This is an example of what the overall folder structure would look like.