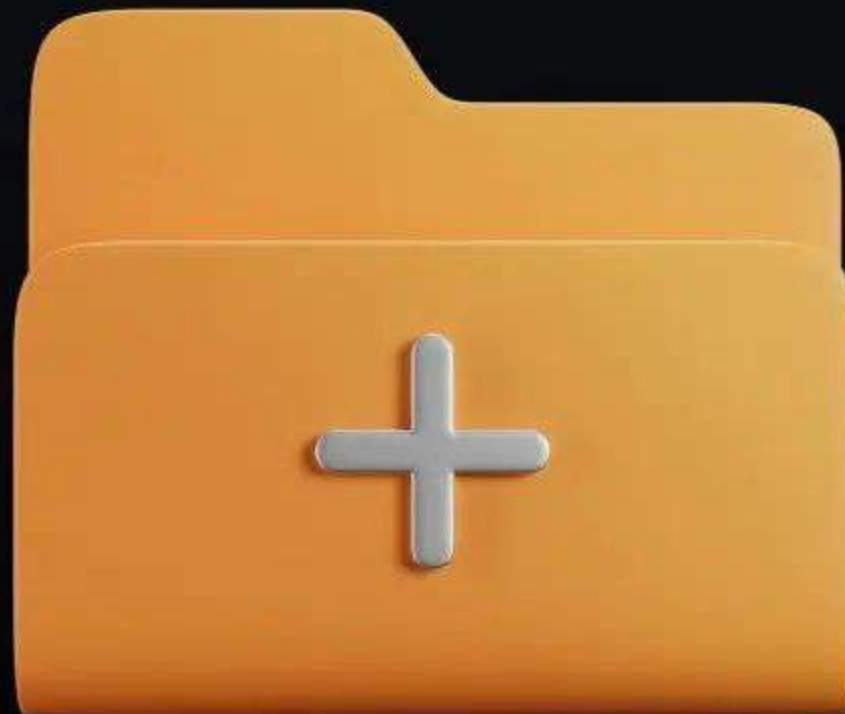




Folder Structure for Web Applications

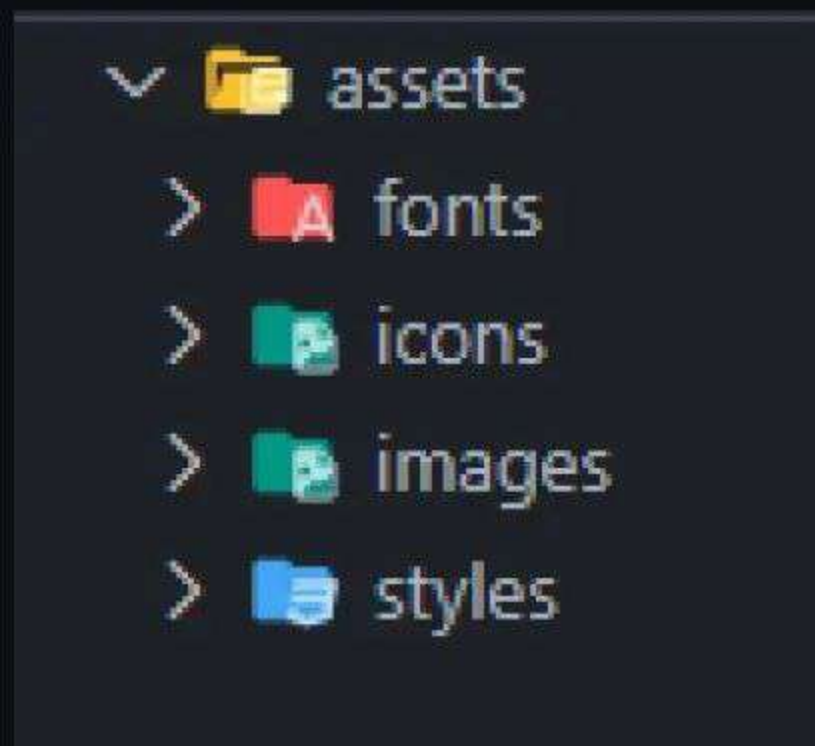


Hey Everyone!

It is critical to create a maintainable folder structure while developing web apps, having the right files in the correct folder helps organize your code and makes other developers have an idea of how the architecture of your web application is or will be during development.

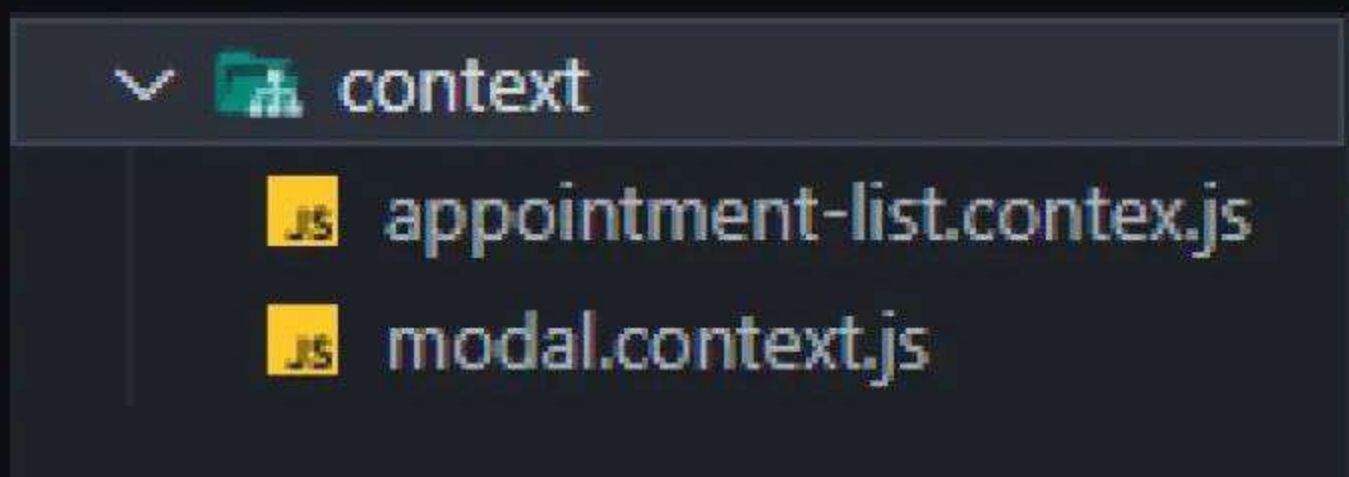
1. Assets

The assets folder contains all images, icons, css files, font files, etc. that will be used in your web application. Custom images, icons, paid fonts are being placed inside this folder.



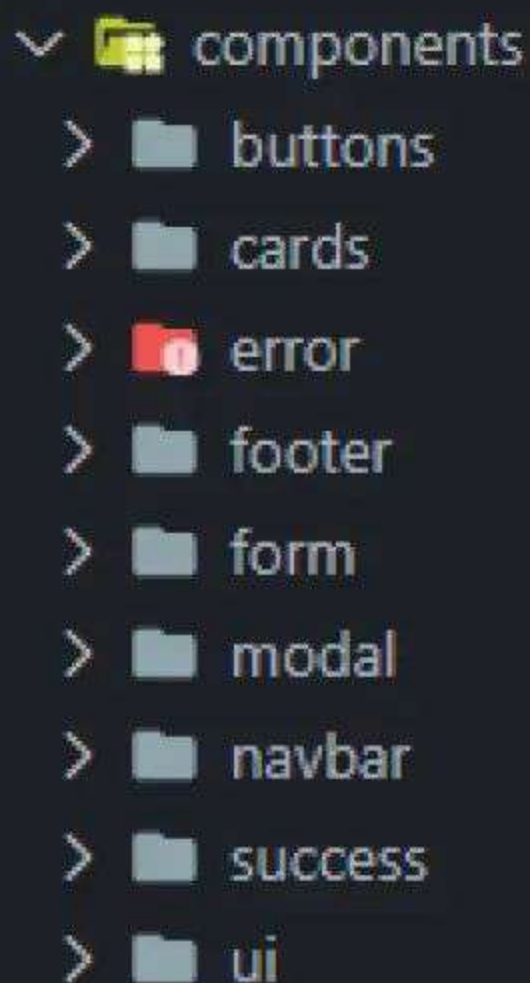
2. Context

When using React Js as your preferred frontend ui library, the context folder stores all your react context files that are used across components and multiple pages.



3. Components

The components folder holds the UI for your application. It contains all our UI components like navbar, footer, buttons, modals, cards, etc.



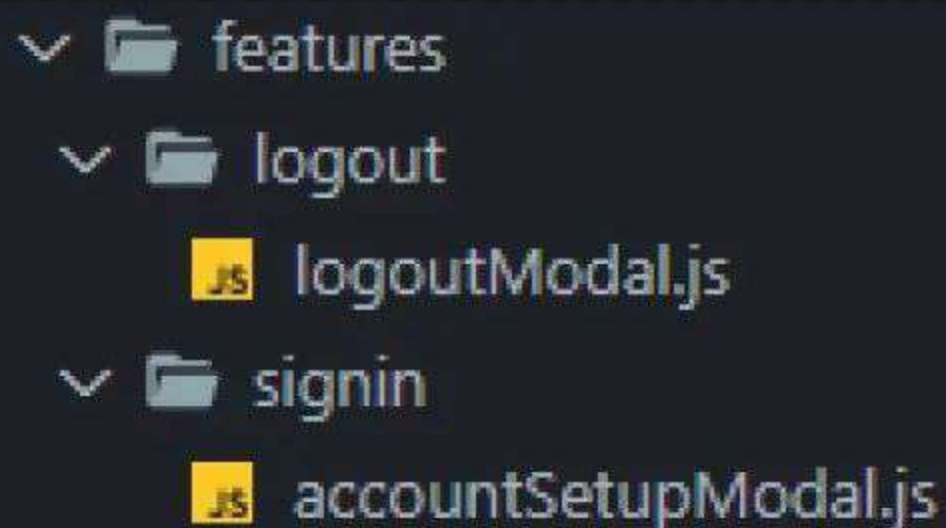
4. Data

The data folder is used for storing text data which will be used in different sections and pages as JSON files. Doing this will enable updating of information easier.

```
{...} navigation.json > ...  
{  
  "navigation": [  
    {  
      "name": "Home",  
      "path": "/"  
    },  
    {  
      "name": "About",  
      "path": "/about"  
    },  
    {  
      "name": "Services",  
      "path": "/services"  
    }  
  ]  
}
```

5. Features

This folder contains individual folder feature for each page (authentication, theme, modals). For example each page might have a modal feature.



```

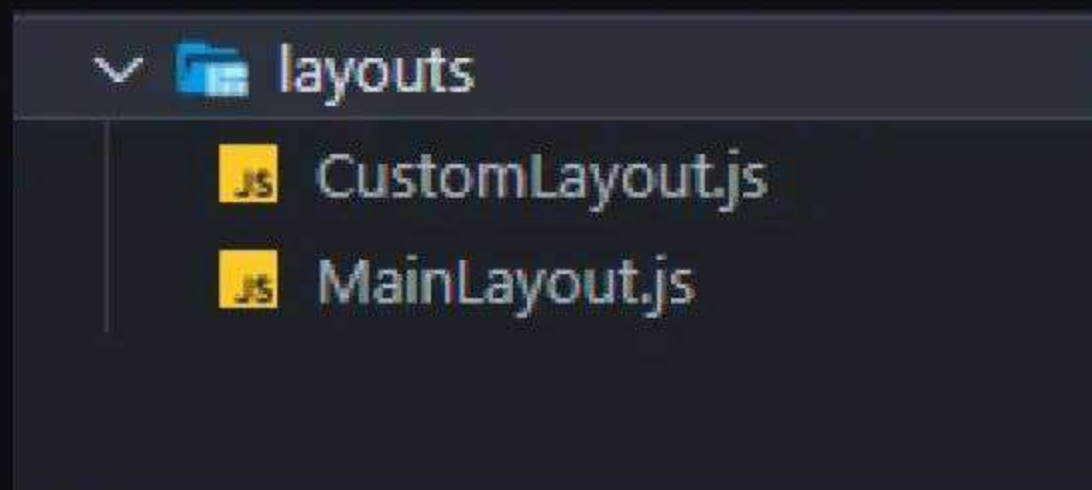
  features
  ├── logout
  │   └── logoutModal.js
  ├── signin
  │   └── accountSetupModal.js

```

A screenshot of a file explorer interface with a dark background. It shows a tree view of a directory structure. The root folder is 'features', which is expanded. Inside 'features', there are two subfolders: 'logout' and 'signin', both also expanded. The 'logout' folder contains a file named 'logoutModal.js'. The 'signin' folder contains a file named 'accountSetupModal.js'. Each file is preceded by a small yellow icon with the letters 'JS' in black.

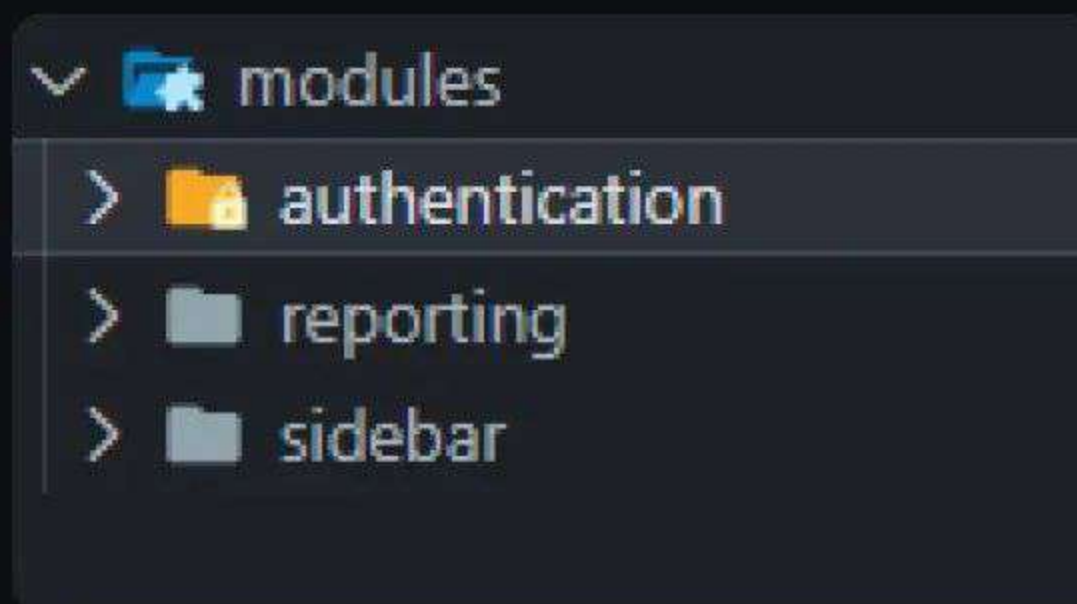
6. Layouts

When defining the general look and feel of the web page, the Layouts folder comes in handy. It is used to place layout-based components such as the sidebar, navbar, and footer. If your web application has many layouts, this folder is a fantastic place to save them.



7. Modules

Modules folder handles specific tasks in your application.





Programwithfun

Did you find this post a bit helpful?

If yes then giving a like and following this page
would mean world to me

Follow For More Coding & Web-Dev Related Content

