Development - Working with template
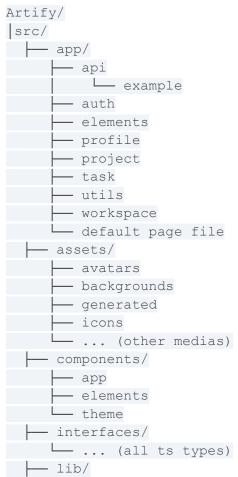
Components Overview:
Below is an overview of the common components provided in [Your Next.js Template]:

App Components: These components are essential for the overall structure and layout of your application. They include header, footer, navigation menus, and sidebar components that provide a consistent user experience throughout the application.

UI Elements: This category encompasses a wide range of reusable UI elements such as buttons, inputs, cards, modals, and alerts. These elements are designed with flexibility in mind, allowing you to easily customize their appearance and behavior to suit your project's requirements.

Theme Components: These components facilitate theming and styling customization within your application. They include components for managing color schemes, typography, spacing, and other design-related aspects, enabling you to create visually appealing interfaces with ease.

Folder structure overview

```
Artify/
|src/
  ├── app/
      ├── api
      │      └── example
      ├── auth
      ├── elements
      ├── profile
      ├── project
      ├── task
      ├── utils
      ├── workspace
      └── default page file
  ├── assets/
      ├── avatars
      ├── backgrounds
      ├── generated
      ├── icons
      └── ... (other medias)
  ├── components/
      ├── app
      ├── elements
      └── theme
  ├── interfaces/
      └── ... (all ts types)
  ├── lib/
```

```
        └── ... (other small re-useable handlers)
    ├── model/
        ├── store
        └── ... (data (chat, project) files)
    ├── modules/
        ├── pages
        ├── profile
        ├── project
        ├── prompt
        ├── sidebar
        └── task
    ├── utils/
        └── ... (elements utils (color, icons etc))
├── package.json
├── next.config.js
├── tailwindcss.config.ts
└── ... (other configuration files)
```

app/: This directory contains pages and logic related to the application, such as API integration, authentication, and default page files.

api/: Contains modules for interacting with external APIs. Each API should have its own directory with relevant files.

auth/: Contains authentication-related components and logic.

elements/: Houses reusable UI elements that can be used across different parts of the application.

profile/: Components related to user profiles.

project/: Components and logic related to projects.

task/: Components and logic related to tasks and kanban. It also have details page for each task

utils/: Utility pages and helpers used throughout the application.

workspace/: Components and logic related to the workspace environment.

assets/: Holds static assets such as images, icons, and backgrounds.

components/: Contains reusable components that can be used across different parts of the application.

theme/: Components related to theming and styling.

interfaces/: Defines TypeScript interfaces and types used throughout the application.

lib/: Contains small, reusable handlers and utility functions.

model/: Contains data models and stores for managing application data.

store/: Stores and state management logic using libraries like Redux.

modules/: Organizes components and logic into modules based on functionality.

Configuration Files:

package.json: Manages project dependencies and scripts.
next.config.js: Configuration file for Next.js.
tailwindcss.config.ts: Configuration file for Tailwind CSS.

Have any questions? Email me at shahzain30196@gmail.com