Amaranta React

Descripción

Amaranta, es un proyecto creado con react y estilizado con css modules.
Amaranta es un emprendimiento nuevo enfocado en la venta de productos hechos a mano con estilo crochet.

Getting Started

Para poder empezar el proyecto debes clonarlo desde el repositorio oficial.

```
git clone https://github.com/ourainbows/amaranta-store
```

Una vez el proyecto este clonado, deberá ingresar a la nueva carpeta

```
cd amaranta-store
```

Ahora debemos instalar las dependencias

```
npm install
```

En el directorio del proyecto, puede ejecutar

```
npm start
```

Ejecuta la aplicación en el modo de desarrollo.
Abra http://localhost:3000 para verlo en el navegador.
La página se volverá a cargar cuando realice cambios.
También puede ver errores de pelusa en la consola.

Api Documentation

```
src/
|-- assets/
|   |-- icons/
|   |   |-- cart.svg
|   |   |-- home.svg
|   |   |-- logo.svg
|   |   |-- logoFooter.svg
|   |   |-- shirt.svg
|   |-- images/
|       |-- exportImages/
|       |   |-- index.js
|       |-- bag.jpeg
|       |-- bags.jpeg
|       |-- bgCoat.jpg
|       |-- brassier.jpg
|       |-- bufanda.jpg
|       |-- bufandaInfinita.jpg
|       |-- bustier.jpg
|       |-- clothePet.jpg
|       |-- clothes.jpg
|       |-- conjuntoMujer.jpg
|       |-- dogFlag.jpeg
|       |-- gaban.jpeg
|       |-- grayScarf.jpeg
|       |-- hatPet.jpeg
|       |-- sacoMascota.jpg
|       |-- shirtYellow.jpg
|       |-- wool.jpeg
```

```
|-- components/
|    |-- AddToCart.jsx
|    |-- ButtonProducts.jsx
|    |-- BuyProduct.jsx
|    |-- CartProduct.jsx
|    |-- CartSvg.jsx
|    |-- Catalog.jsx
|    |-- Footer.jsx
|    |-- GridImages.jsx
|    |-- HomeSvg.jsx
|    |-- Navbar.jsx
|    |-- Navigation.jsx
|    |-- Product.jsx
|    |-- ScrollToTop.jsx
|    |-- ShirtSvg.jsx
|    |-- Modal.jsx
|    |-- TrashSvg.jx
|-- context/
|    |-- ProductsProvider
|         |-- ProductsProvider.js
|-- pages/
|    |-- Buy.jsx
|    |-- Cart.jsx
|    |-- Home.jsx
|    |-- NotFound.jsx
|    |-- Product.jsx
|-- routes/
|    |-- routes.jsx
|-- styles/
|    |-- addToCart.module.css
|    |-- buttonProdcts.module.css
|    |-- buy.module.module.css
|    |-- buyProduct.module.css
```

| |-- cart.module.css
| |-- cartProduct.module.css
| |-- catalog.module.css
| |-- footer.module.css
| |-- gridImages.module.css
| |-- home.module.css
| |-- index.module.css
| |-- navbar.module.css
| |-- navigation.module.css
| |-- notFound.module.css
| |-- product.module.css
| |-- products.module.css
|-- App.jsx
|-- data.js
|-- index.js

Assets

Contiene las imágenes e iconografía utilizadas en el proyecto.

ExportImages

El archivo index.js que se encuentra dentro de este directorio exporta todas las imágenes que tenemos en assets/images para posteriormente usarlas en nuestro archivo data.js .

Components

AddToCart

Recibe como propiedad el objeto completo de un producto, su funcionalidad es agregar un producto al carrito, se activa cuando damos click al botón que tiene como elemento nos ejecuta la función onAdd, la cual agrega nuevos valores a el objeto de producto, estos nuevos valores nos servirán posteriormente cuando el usuario esté haciendo las elecciones de tipo talla, cantidad y color. Adicional a eso nos guarda nuestro objeto modificado en un array con los objetos que también hemos ido agregando al carrito.

ButtonProducts
Este componente se renderiza en el home, su única funcionalidad es direccionar a la vista de productos.

BuyProduct
Crea la tarjeta de producto para la vista de completar-compra, esta tarjeta contendrá la imagen del producto, nombre, color, talla y cantidad seleccionados por el usuario y el precio que está dado por el precio base multiplicado por la cantidad de productos seleccionados.

Cart Product
Es el componente con el mayor grado de complejidad de toda el proyecto, renderiza el producto previamente agregado al carrito, en la vista del carrito. Además del producto del carrito tenemos opciones con las que el usuario puede interactuar para escoger los valores que más se adecuen a lo que vaya a comprar. Puede seleccionar cantidad, color y talla, además de poder eliminar el producto en caso de ser requerido.
Aquí tenemos varias funciones que nos ayudan a modificar el estado del componente cada vez que el usuario cambie alguno de los valores mencionados anteriormente.

CartSvg, HomeSvg, ShirtSvg, TrashSvg
Son componentes que exportan un svg, se decidió trabajarlos como componentes para modificar de forma más sencilla su estilo de manera dinámica.

Catalog
Su función es ser un componente contenedor y crear un componente de tipo Product para cada objeto que tenemos en nuestra data, este componente catalog son los productos que podemos ver en la vista de productos.

Footer
Es el componente que renderiza el footer y solo se muestra en la vista principal.

Grid Images
Crea un componente que renderiza 6 imágenes que con ayuda de css mostramos las imágenes con un estilo muy particular en nuestra vista principal.

Navbar
El navbar es el componente que nos acompañará en todas las vista de la aplicación, este navbar va en la parte superior, en la versión de escritorio podemos ver renderizados una lista de elementos que nos servirán para navegar entre las diferentes rutas.

Navigation
Es la barra de navegación para las vistas de móvil y tablet, la componen 3 botones, que redirigen a 3 vistas diferentes de la aplicación, el Inicio, los productos y nuestro carrito de compras. Como dato adicional estamos usando nuestro cartProducts traído desde el context para poder mostrar al usuario en el carrito cuantos productos tiene agregados.

Product
Crear una tarjeta donde se muestra la información del producto que le entra como propiedad. Este componente es llamado por el componente catalog.

ScrollToTop
Cada vez que cambiamos de página nos va a hacer un scroll a el top de la página.

Modal
Componente que nos muestra un modal en la pantalla, tiene como finalidad avisar al usuario que será redireccionado a whatsapp para completar su compra.

Context

Product Provider
Aquí usamos react context para exportar, el estado del carrito de compras para poderlo usar en las diferentes rutas y poder agregarle información en nuestra vista de productos y el precio total del carrito.

Pages

Buy
Esta es la vista final, en ella tenemos un formulario donde el usuario pone su nombre, dirección y algunos comentarios adicionales sobre su compra.
Una breve sección donde nos muestra el resumen de nuestra compra y el botón de comprar. Este botón llama a una función que procesa todo nuestra cartProducts y crea un string con la información procesada para su legibilidad, luego hacemos un encode a este string y finalmente nos redirige a Whatsapp para que el cliente y el vendedor se pongan en contacto.

Cart
Nuestra vista de carrito, recibe todas las propiedades de nuestro contexto y de este mismo contexto renderiza cartProducts, que recordemos es el que trae los objetos que el usuario ha agregado al carrito.

Home
Nos da una breve introducción a nuestro sitio,renderizamos los componentes de gridImages, buttonProducts y el footer.

NotFound
Una página en caso de que tengamos un error 404.

Products
Es la vista donde se mostrará todo nuestro catálogo, tenemos tres botones que nos filtran dependiendo de la categoría seleccionada.

Routes

Routes
Este componente es el encargado de manejar las 5 vistas que tiene nuestra aplicación (También estamos contando la vista de error 404).

Styles
Son los estilos para todos nuestros componentes y páginas.

App
Recibe nuestro componente de rutas.

Data
Es un array de objetos de los productos que tenemos en nuestra tienda, cada objeto de producto tiene: id, nombre, color, tamaño, precio, categoría y una imagen que estamos consumiendo del exportImages.

Index.js
Nuestro archivo padre que renderiza la aplicación.