

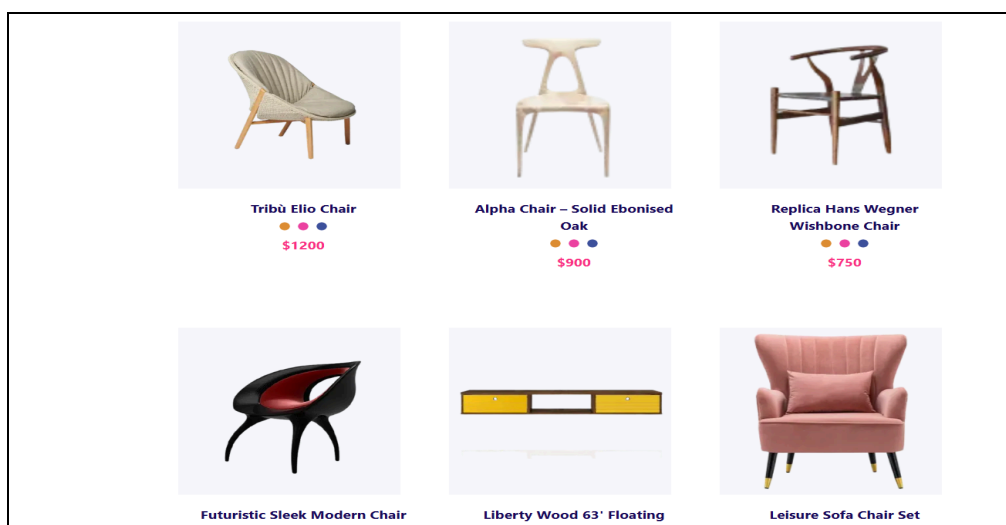
## Day 4 - Dynamic Frontend Components Ecommerce Website ( Hekto )

Today was an exciting and productive day in my journey of creating a marketplace. I focused on building a **dynamic frontend** that brings the platform to life and provides users with an interactive shopping experience. Here's a breakdown of what I achieved:

### Product Listing Component:-

The **Product Listing Component** dynamically showcasing all available products in a clean and organized layout. It fetches product data from a backend API or a static file and renders each item within responsive and visually appealing cards.

- **Dynamic Data Rendering:** Displays product details like name, image, description, price, and category fetched from the backend using groq query.
- **Responsive Design:** Ensures the grid adjusts seamlessly across devices for a consistent user experience.
- **Interactive Cards:** Each product card is styled to highlight essential information, with hover effects and click actions for deeper interaction, such as navigating to a single product page.



## Product Details Component:-

The component provides a comprehensive view of an individual product, allowing users to explore its features and details in depth. This component is designed to dynamically display data passed from the Product Listing Page through the URL using search parameters.

- **Dynamic Data Retrieval:** Retrieves product details such as name, price, description, image, and other attributes from the URL using `useSearchParams` in Next.js. This approach ensures seamless navigation and accurate rendering of the selected product.
- **Interactive Elements:** Users can interact with buttons for adding the product to the cart or initiating the checkout process directly from this page.

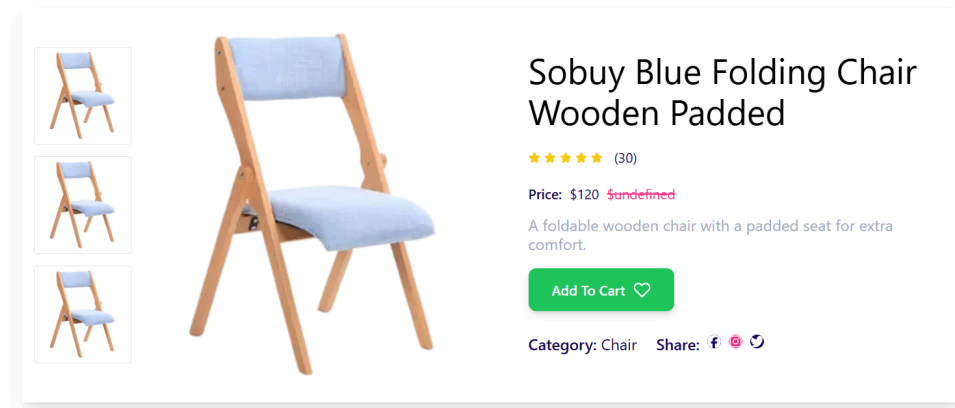
### Some Code Snippets:

```
<div className="p-4 md:text-center">
  <Link href={`/${products.id?name=${product.name}&description=${product.description}&price=${product.price}&imagePath=${
    product.imagePath
  }&category=${product.category}&discountPercentage=${product.discountPercentage}&stockLevel=${product.stockLevel}&
    isFeaturedProduct=${product.isFeaturedProduct}`}>
    <h3 className="text-sm lg:text-base font-bold text-headingsText">
      {product.name}
    </h3>
  </Link>
</div>
```

```
interface IProduct {
  id : number,
  name : string,
  price : number,
  discountPercentage: number,
  imagePath : string,
  category : string,
  description:string,
  stockLevel:number,
  isFeaturedProduct:boolean
}

async function singleProductPage({searchParams} : {searchParams: Promise<{}>}) {
  const {relatedProducts} = await allProducts()
  const {name,price,description,imagePath,category,discountPercentage,stockLevel} = await searchParams
```

## Product Detail Page Screenshot:



## Cart Component:-

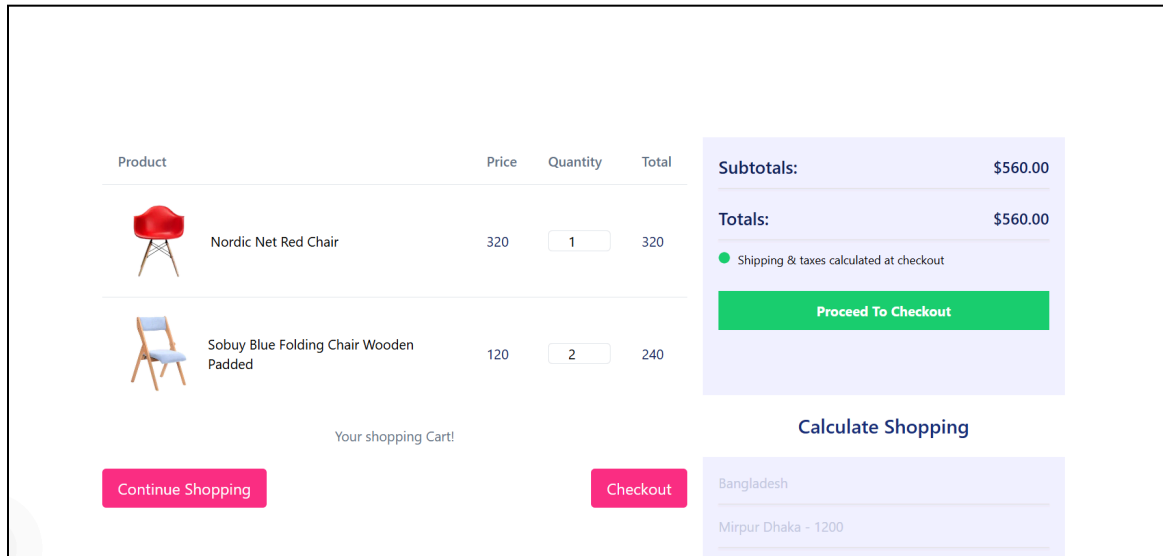
The Cart Component serves as the heart of the shopping experience. It dynamically updates as users add or remove items, ensuring a seamless and interactive experience.

- Displays all products added to the cart with their details such as name, quantity, and price.
- Provides options to update the quantity or remove items from the cart.
- Automatically calculates and updates the total price as changes are made.

### Code Snippet:

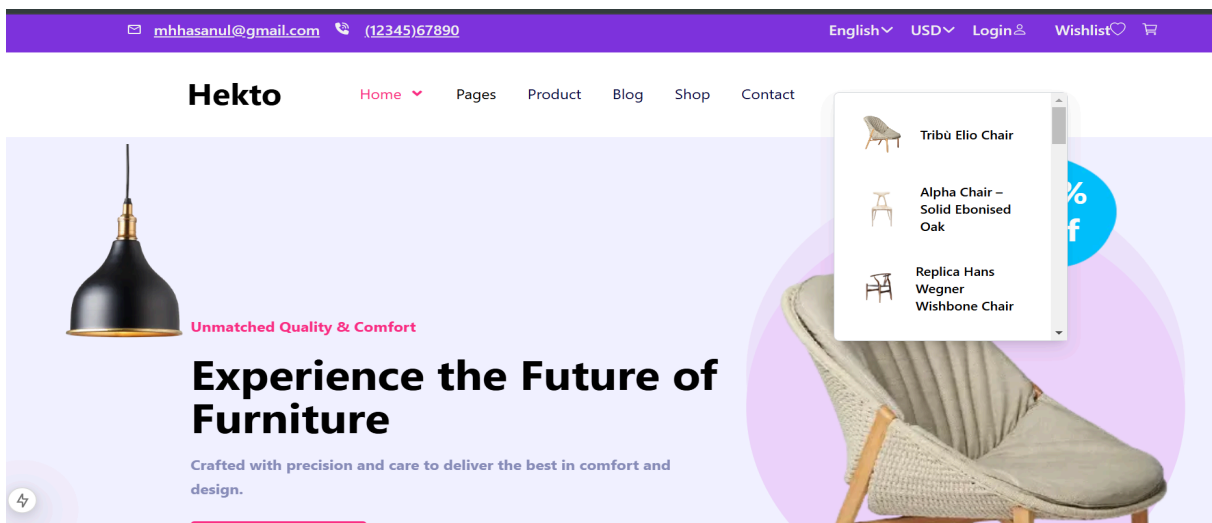
```
function ShoppingCart() {  
  const searchParam = useSearchParams();  
  const router = useRouter();  
  const [cart, setCart] = useState<IProduct[]>([]);  
  const [qty, setQty] = useState(1)  
  
  useEffect(() => {  
    }, [searchParam, router]);  
  
  const removeItem = (id: string) => {  
    };  
  
  function handleQuantity(index: number, num: string) {  
    }  
  
  const getTotalPrice = () => {  
    };  
}
```

## Cart Component Screenshot:



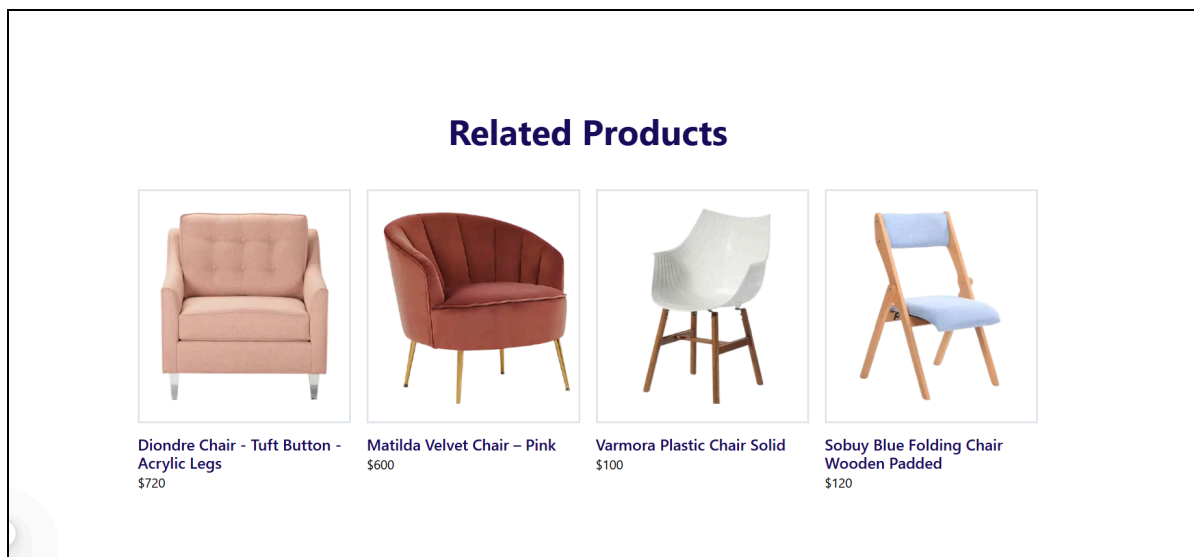
## Search Bar Component:-

The search bar component is designed to enhance user experience by providing an intuitive way to search for products. It fetches matching results dynamically and displays them in a dropdown menu, allowing users to quickly navigate to desired products.



## Related Product Components:-

This component can be used to display products that are similar or complementary to the currently viewed product. It includes responsive styling and dynamic rendering of related products:



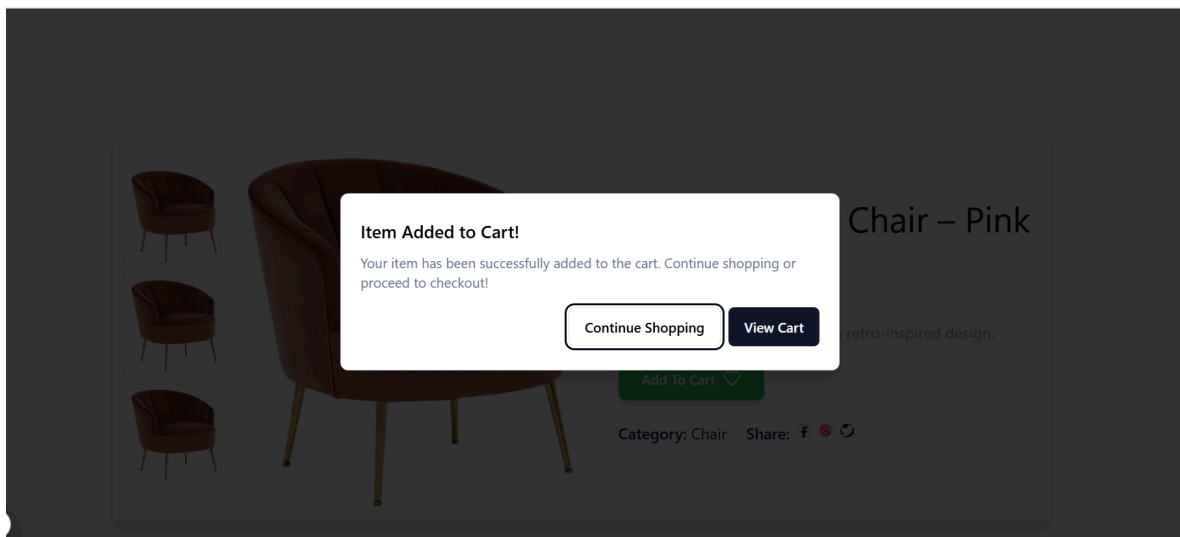
## Header and Footer Components:-

The **Header** and **Footer** components provide navigation, branding, and essential links while ensuring a responsive design for seamless accessibility across devices.

- **Navigation Links:** Includes links to essential pages like Home, Products, Cart, and Contact Us.
- **Responsive Design:** The navigation menu adjusts gracefully for mobile, tablet, and desktop views, with a hamburger menu for smaller screens.
- **Quick Links:** Provides links to important pages like Privacy Policy, Terms of Service, and Help Center.

## Notification Components:-

The **Notification Component** is designed to notify users of successful actions or events, such as when a product is successfully added to the cart or when a project is added. For this, I used the **ShadCN library** and specifically the **AlertDialog** component to display notifications in a clean and accessible manner.



## FAQ Component:-

The **FAQ (Frequently Asked Questions) Component** is designed to provide users with quick answers to common queries about the marketplace. This component improves the user experience by reducing the need for direct support interactions and making information easily accessible. I have used Shadcn Accordion Component to make it.

How can I contact customer support? ^

You can reach out to our customer support team via the Contact Us page. Simply fill out the form, and we'll get back to you as soon as possible.

What if I receive a damaged or incorrect product? v

How do I care for the products I purchase? v

Do I need to create an account to place an order? v

Ask a Question

Your Name

Subject

Type Yor Message