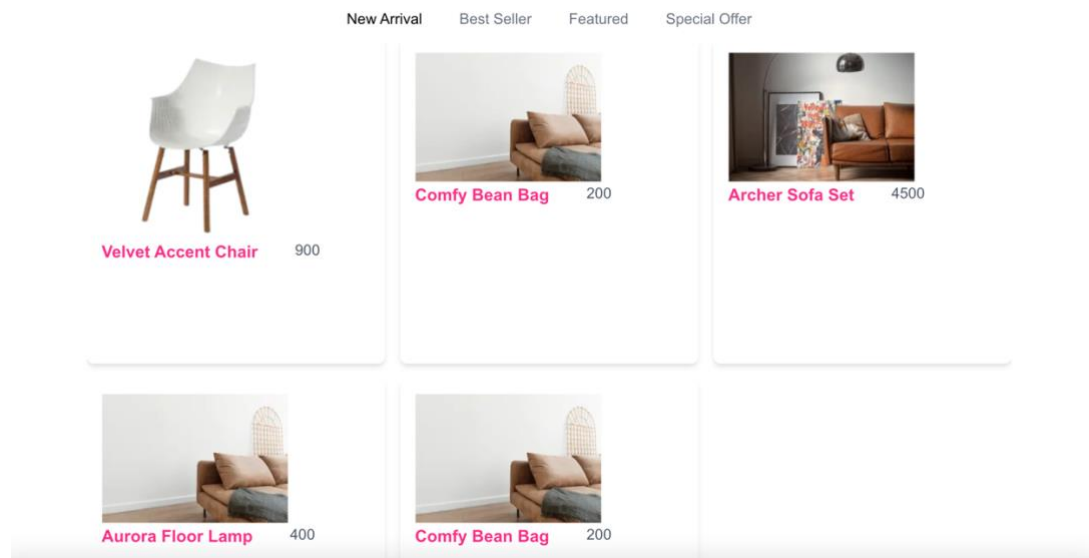


## Day4 building dynamic component for your market place – furniture

- 1- Product listing components
  - a. Product on main page (with tags )
    - i. Product display with tags



Source code

"use client"

```
import Link from "next/link";
import { client } from "@sanity/lib/client";
import { urlFor } from "@sanity/lib/image";
import { useEffect, useState } from "react";
import React from "react";
import Slider from "react-slick"; // Import react-slick
import "slick-carousel/slick/slick.css"; // Import Slick CSS
import "slick-carousel/slick/slick-theme.css"; // Import Slick Theme
import Image from "next/image";
import P1 from '../public/image 1.png';
import CART from '../public/website/fluent_cart-24-regular.png';
import Heart from '../public/website/ui_heart-alt.png';
import magnify from '../public/website/ui_search-plus (1).png';
```

```
function Products() {
  const [products, setProduct] = useState<any[]>([]);

  useEffect(() => {
    async function fetchData() {
      const query = `
        *[_type == "product" && isFeatured == true] {
          _id,
          name,
```

```

        price,
        description,
        discountPercentage,
        stockLevel,
        "categoryName": category->title, // Fetch the title of the referenced category
        rating,
        color[],
        additionalInfo,
        image
    }
    `;

    const data = await client.fetch(query);
    setProduct(data);
}

    fetchData();
}, []);
const sliderSettings = {
    dots: true, // Pagination dots
    infinite: true, // Infinite scrolling
    speed: 500, // Transition speed
    slidesToShow: 4, // Default slides visible
    slidesToScroll: 1,
    // Scroll one slide at a time
    responsive: [
        {
            breakpoint: 1024, // lg and below
            settings: {
                slidesToShow: 1, // Show 1 slide
                vertical: true, // Enable vertical scrolling
            },
        },
    ],
};

return (
    <section className="space-y-4 lg:pr-56 lg:pl-56 pt-20 overflow-x-hidden">
        <div className="text-center">
            <h1 className="text-3xl font-semibold">Featured Products</h1>
        </div>
        { /* Slick Carousel */ }
        <div className="lg:block pt-10">
            <Slider {...sliderSettings}>
                {products.map((product) => (
                    <div key={product.id}>

```

```

    <div className="bg-white shadow-md rounded-lg transition-shadow duration-300
group relative p-4 mx-auto">
    <div className="hidden group-hover:block h-[29px] w-[90px]">
      <div className="flex">
        <Image src={Heart} alt="c"width={15} height={15}/>
        <Image src={CART} alt="c"width={15} height={15}/>
        <Image src={magnify} alt="c"width={15} height={15}/>

      </div>
    </div>
    <div className="w-[270px] h-[236px] bg-gray-300 p-4">
      {product.image ? (
        <Image
          src={urlFor(product.image).url()} // Use category.image
          alt={product.name || "Category Image"}
          width={200}
          height={200}
        />
      ) : (
        <p>No image available</p> // Fallback for missing images
      )}

      <div className="hidden group-hover:block h-[29px] w-[94px] bg-green pl-1 pr-1
text-xs p-2 mt-3 ml-3">
        View Details
      </div>
      <div>
        <div className="hover:bg-blue hover:text-white">
          <h3 className="text-lg font-semibold mb-2 text-pink text-
center">{product.name}</h3>
          <p className="text-gray-600 mb-2 text-center">Code:{product.code}</p>
          <p className="text-gray-600 mb-2 text-center font-bold">{product.price}</p>
          <p className="text-black">{product.category}</p>
        </div>
      </div>
    </div>
  )}
</Slider>

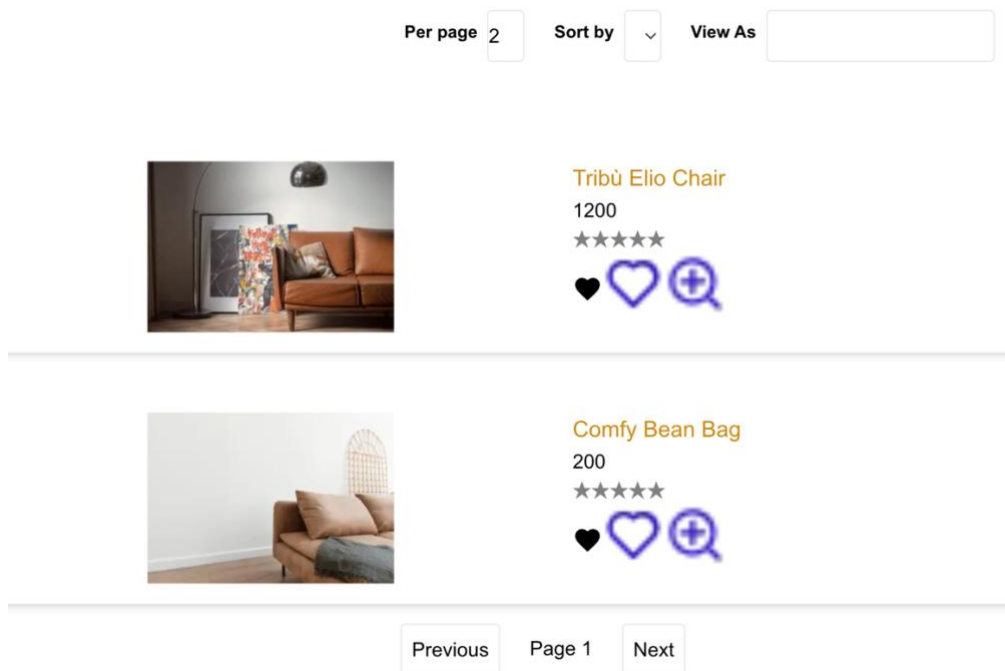
  { /* Decorative Ellipsis Below */ }
</div>
</section>
);
};

```

```

export default Products;
2- Product detail page

```



Source code

"use client"

import StarRating from "@components/StarRating";

import Link from "next/link";

import { urlFor } from "../../sanity/lib/image";

import { useWishlist } from "../../context/wishContext";

import Image from "next/image";

import Wish from '../../public/website/Group 28.png';

import { useEffect, useState } from "react";

import Heart from '../../public/website/ui\_heart-alt.png';

import Magnify from '../../public/website/ui\_search-plus.png';

import { client } from "../../sanity/lib/client"; // Your Sanity client setup

async function getProductsByCategory(id: string, page: number, perPage:

number, sortOrder: string) {

const query = `\*\_type == "product" && category->\_id == \$id | order(price  
\${sortOrder}) [\${(page - 1) \* perPage}...\${page \* perPage}] {`

\_id,

name,

price,

image,

category-> { \_id, name }

};

const data = await client.fetch(query, { id });

return data;

```
}
```

```
export default function CategoryPage({ params }: { params: { id: string } }) {
  const [products, setProducts] = useState<any[]>([]);
  const [loading, setLoading] = useState(true);
  const [isOpen, setIsOpen] = useState(false);
  const [page, setPage] = useState(1);
  const [perPage, setPerPage] = useState(2); // Show 2 products per page
  const [sortOrder, setSortOrder] = useState("asc"); // Default sorting: ascending
  const { addToWishlist } = useWishlist();
  const handlewishlist = (product:any) => {
    addToWishlist({ ...product});
  };
  // Fetch products when the component mounts
  useEffect(() => {
    async function fetchProducts() {
      const data = await getProductsByCategory(params.id, page, perPage,
sortOrder);
      setProducts(data);
      setLoading(false);
    }

    fetchProducts();
  }, [params.id, page, perPage, sortOrder]); // Dependency array to fetch again if
the category changes
```

```
if (loading) return <div>Loading...</div>;
```

```
return (
  <div className="">
    <div className="mx-auto w-full bg-lightpurple">
      <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4 p-10">
        <nav className="text-sm">
          <h1 className="text-6xl text-black">Shop</h1>
          <ol className="flex space-x-2">
            <li>
              <Link href="/" className="text-blue-500 hover:text-blue-
700">Home</Link>
            </li>
            <li className="text-gray-500">.</li>
            <li>
              <Link href="#" className="text-blue-500 hover:text-blue-
700">Shop</Link>
            </li>
          </ol>
        </nav>
      </div>
    </div>
  )
```

```

    </div>
    <div className="grid grid-cols-1 lg:grid-cols-2 xl:grid-cols-2 2xl:grid-cols-2 gap-4">
      <div className="p-4"><h1 className="text-black font-extrabold">Ecommerce accessories & Fashion Items</h1></div>
      <div className="p-4">
        <div className="flex gap-4">
          <div className="flex">
            <label htmlFor="item1" className="text-sm font-semibold p-2">Per
page</label>
            <input type="number" id="item1" className="border rounded w-[30px]"
value={perPage} onChange={(e) => setPerPage(Number(e.target.value))} />
          </div>
          <div className="flex">
            <label htmlFor="item2" className="text-sm font-semibold p-2">Sort
by</label>
            <select id="item2" className="border p-2 rounded w-[30px]" onChange={(e)
=> setSortOrder(e.target.value)}>
              <option value="asc">Price: Low to High</option>
              <option value="desc">Price: High to Low</option>
            </select>
          </div>
        </div>
        <div className="flex">
          <label htmlFor="item3" className="text-sm font-semibold p-2">View
As</label>
          <input type="text" id="item3" className="border p-2 rounded" />
        </div>
      </div>
    </div>
    </div>
    </div>
    <div className="h-screen grid grid-cols-1 lg:grid-cols-5">
      { /* Left Sidebar (20%) */ }
      <div
        className={`lg:block ${isOpen ? "block" : "hidden"} lg:col-span-1 text-black
p-4 transition-all duration-300 ease-in-out`}
      >
        <button onClick={() => setIsOpen(false)} className="lg:hidden text-white">
Close
        </button>
        <div className="space-y-4">
          <div>
            <h1 className="text-sm">Product Brand</h1>
            <div className="flex items-center space-x-2">
              <input type="checkbox" id="checkbox1" className="form-checkbox text-
blue-500" />
              <label htmlFor="checkbox1" className="text-lg font-semibold">Unique
Furniture Restor</label>

```

```

    </div>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <label htmlFor="checkbox1" className="text-lg font-semibold">Unique
Furniture Restor</label>
    </div>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <label htmlFor="checkbox1" className="text-lg font-semibold">Fusion Dot
High Fashion</label>
    </div>
  </div>
  <br/>
  <div>
    <h1 className="text-sm">Discount Offer</h1>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <label htmlFor="checkbox1" className="text-lg font-semibold">20% cash
back</label>
    </div>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <label htmlFor="checkbox1" className="text-lg font-semibold">5% cash
back offer</label>
    </div>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <label htmlFor="checkbox1" className="text-lg font-semibold">25%
discount offer</label>
    </div>
  </div>
  <div>
    <h1 className="text-sm">Rating Item</h1>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
      <StarRating rating={4}/>
    </div>
    <div className="flex items-center space-x-2">
      <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />

```

```

        <StarRating rating={4}/>

    </div>
    <div className="flex items-center space-x-2">
        <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
        <StarRating rating={4}/>
    </div>
    <br/>
    <div>
        <h1 className="text-sm">Categories</h1>
        <div className="flex items-center space-x-2">
            <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
            <label htmlFor="checkbox1" className="text-lg font-semibold">Electronic</label>
        </div>
        <div className="flex items-center space-x-2">
            <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
            <label htmlFor="checkbox1" className="text-lg font-semibold">Jewellery</label>
        </div>
    </div>
    <br/>
    <div>
        <h1 className="text-sm">Price Filter</h1>
        <div className="flex items-center space-x-2">
            <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
            <label htmlFor="checkbox1" className="text-lg font-semibold">$0-$500</label>
        </div>
        <div className="flex items-center space-x-2">
            <input type="checkbox" id="checkbox1" className="form-checkbox text-blue-500" />
            <label htmlFor="checkbox1" className="text-lg font-semibold">$90-900</label>
        </div>
    </div>
    </div>
    </div>
    </div>
    </div>
    </div>

    { /* Right Content (80%) */ }
    <div className="lg:col-span-4 p-4">
        { /* Toggle Icon for Mobile */ }

```



```

<button
  onClick={() => setIsOpen(true)}
  className="lg:hidden text-xl text-gray-800"
>
  &#9776; { /* Hamburger icon */ }
</button>

<div className="mt-4">
<div>
{products.map((p:any) => (
  <div className="grid grid-cols-1 lg:grid-cols-2 xl:grid-cols-2 2xl:grid-cols-2
gap-4 mt-8 text-black lg:pr-56 lg:pl-56 shadow-md bg-white"key={p.id}>
    <div className="p-4"> {p.image ? (
      <Image
        src={urlFor(p.image).url()} // Use category.image
        alt={p.name || "Category Image"}
        width={200}
        height={200}
      />
    ) : (
      <p>No image available</p>
    )}</div>
    <div className="p-4"><h1 className="text-lg"><Link
      href={` /product/${p._id}`}
      className="text-yellow-600"
    >{p.name}</Link></h1>
    <div className="text-black"><p>{p.price}<span className="hover:text-
pink text-black">{p.discount}</span></p><StarRating rating={p.rating}/>
    <p>{p.description}</p>
    </div>
    <div className="flex">
    <button
      className=" text-white mt-2"
      onClick={handlewishlist}
    >
      <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"
fill="#000000" width="24" height="24">
        <path d="M12 21.35l-1.45-1.32C5.4 15.36 2 12.28 2 8.5 2 5.42 4.42 3 7.5
3c1.74 0 3.41.81 4.5 2.09C13.09 3.81 14.76 3 16.5 3 19.58 3 22 5.42 22 8.5c0
3.78-3.4 6.86-8.55 11.54L12 21.35z"/>
      </svg>
    </button>
    <Image src={Heart} alt="c" height={50} width={50}/>
    <Image src={Magnify} alt="c" height={50} width={50}/>
    </div>
  </div>
</div>
</div>

```

```

    ))}
  </div>
  <div className="mt-4 flex justify-center space-x-4">
    <button
      onClick={() => setPage(prev => Math.max(prev - 1, 1))}
      className="border p-2 rounded"
    >
      Previous
    </button>
    <span className="p-2">Page {page}</span>
    <button
      onClick={() => setPage(prev => prev + 1)}
      className="border p-2 rounded"
    >
      Next
    </button>
  </div>
</div>
</div>
</div>
</div>

);
}

```

Related product

"use client"

```

import Link from "next/link";
import { client } from "@sanity/lib/client";
import { urlFor } from "@sanity/lib/image";
import { useEffect, useState } from "react";
import React from "react";
import Slider from "react-slick"; // Import react-slick
import "slick-carousel/slick/slick.css"; // Import Slick CSS
import "slick-carousel/slick/slick-theme.css"; // Import Slick Theme
import Image from "next/image";
import CART from '../public/website/fluent_cart-24-regular.png';
import Heart from '../public/website/ui_heart-alt.png';
import magnify from '../public/website/ui_search-plus (1).png';

interface ProductsProps {
  category: string; // New prop to pass category name
}

function Products({ category }: ProductsProps) {

```

```

const [products, setProduct] = useState<any[]>([]);

useEffect(() => {
  async function fetchData() {
    // Adjust the query to use category's name to fetch matching products
    const query = `
      *[_type == "product" && category -> _id == $category] {
        _id,
        name,
        price,
        description,
        discountPercentage,
        stockLevel, // Fetch category title directly
        rating,
        color[],
        additionalInfo,
        image
      }
    `;
    // Pass the category name as a parameter for querying
    const data = await client.fetch(query, { category });
    setProduct(data);
  }

  fetchData();
}, [category]); // Dependency on category prop to refetch when it changes

const sliderSettings = {
  dots: true, // Pagination dots
  infinite: true, // Infinite scrolling
  speed: 500, // Transition speed
  slidesToShow: 4, // Default slides visible
  slidesToScroll: 1, // Scroll one slide at a time
  responsive: [
    {
      breakpoint: 1024, // lg and below
      settings: {
        slidesToShow: 1, // Show 1 slide
        vertical: true, // Enable vertical scrolling
      },
    },
  ],
};

return (
  <section className="space-y-4 lg:pr-56 lg:pl-56 pt-20 overflow-x-hidden">
    <div className="text-center">

```

```

    <h1 className="text-3xl font-semibold">Related Products </h1>
  </div>
  { /* Slick Carousel */ }
  <div className="lg:block pt-10">
    <Slider {...sliderSettings}>
      {products.map((product) => (
        <div key={product._id}>
          <div className="bg-white shadow-md rounded-lg transition-shadow
duration-300 group relative p-4 mx-auto">
            <div className="hidden group-hover:block h-[29px] w-[90px]">
              <div className="flex">
                <Image src={Heart} alt="c" width={15} height={15} />
                <Image src={CART} alt="c" width={15} height={15} />
                <Image src={magnify} alt="c" width={15} height={15} />
              </div>
            </div>
            <div className="w-[270px] h-[236px] bg-gray-300 p-4">
              {product.image ? (
                <Image
                  src={urlFor(product.image).url()}
                  alt={product.name || "Product Image"}
                  width={200}
                  height={200}
                />
              ) : (
                <p>No image available</p> // Fallback for missing images
              )}
              <div className="hidden group-hover:block h-[29px] w-[94px] bg-green
pl-1 pr-1 text-xs p-2 mt-3 ml-3">
                View Details
              </div>
            </div>
            <div className="hover:bg-blue hover:text-white">
              <h3 className="text-lg font-semibold mb-2 text-pink text-
center">{product.name}</h3>
              <p className="text-gray-600 mb-2 text-center">Code:
{product.code}</p>
              <p className="text-gray-600 mb-2 text-center font-
bold">{product.price}</p>
              { /* Render category name here */ }
              <p className="text-black">{product.category}</p>
            </div>
          </div>
        </div>
      )})}
    </Slider>
  </div>

```

```

    </section>
  );
};

export default Products;

```

### 3- Advanced category filtering

Source code

```

"use client"
import Link from "next/link";
import React from "react";
import { client } from "@/sanity/lib/client";
import { urlFor } from "@/sanity/lib/image";
import { useState, useEffect } from "react";
import Slider from "react-slick"; // Import react-slick
import "slick-carousel/slick/slick.css"; // Import Slick CSS
import "slick-carousel/slick/slick-theme.css"; // Import Slick Theme
import Image from "next/image";
function Products() {
  const [products1, setProduct] = useState<any[]>([]);

  useEffect(() => {
    async function fetchData() {
      const query = `
        *[_type == "category1"] {
          _id,
          id,
          name,
          image
        }
      `;
      const data = await client.fetch(query);
      setProduct(data);
    }

    fetchData();
  }, []);

  const sliderSettings = {
    dots: true, // Pagination dots
    infinite: true, // Infinite scrolling
    speed: 500, // Transition speed
    slidesToShow: 4, // Default slides visible
    slidesToScroll: 1,
    // Scroll one slide at a time
  };

```

```

responsive: [
  {
    breakpoint:1366,
    settings:{
      slideToShow:2,
      vertical:false
    }
  },
  {
    breakpoint: 1024, // lg and below
    settings: {
      slidesToShow: 2, // Show 1 slide
      vertical: false, // Enable vertical scrolling
    },
  },
  {
    breakpoint: 768, // lg and below
    settings: {
      slidesToShow: 1, // Show 1 slide
      vertical: true, // Enable vertical scrolling
    },
  },
  {
    breakpoint: 640, // md and below
    settings: {
      slidesToShow: 1, // Show 2 slides
      vertical: false, // Horizontal scrolling
    },
  },
],
};
return (
  <section className="space-y-4 lg:pr-32 lg:pl-32 pt-20 sm:pr-20 sm:pl-20
overflow-x-hidden">
    <div className="text-center">
      <h1 className="text-3xl font-semibold">Top Categories</h1>
    </div>
    <div className="lg:block pt-10 slider-container mx-auto p-2 sm:pl-10">
      <Slider {...sliderSettings}>
        {products1.length === 0 ? (
          <p>No categories available</p> // Fallback if no categories found
        ) : (
          products1.map((c) => (
            <div key={c._id} className="flex justify-center" data-testid="product-
category">
              <div className="w-[269px] h-[345px] bg-white group relative mx-auto">
                {c.image ? (

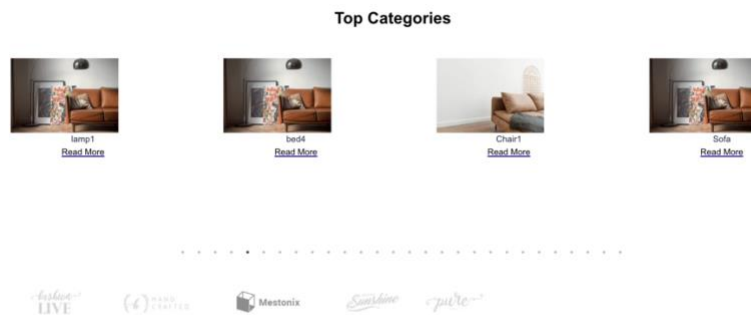
```

```

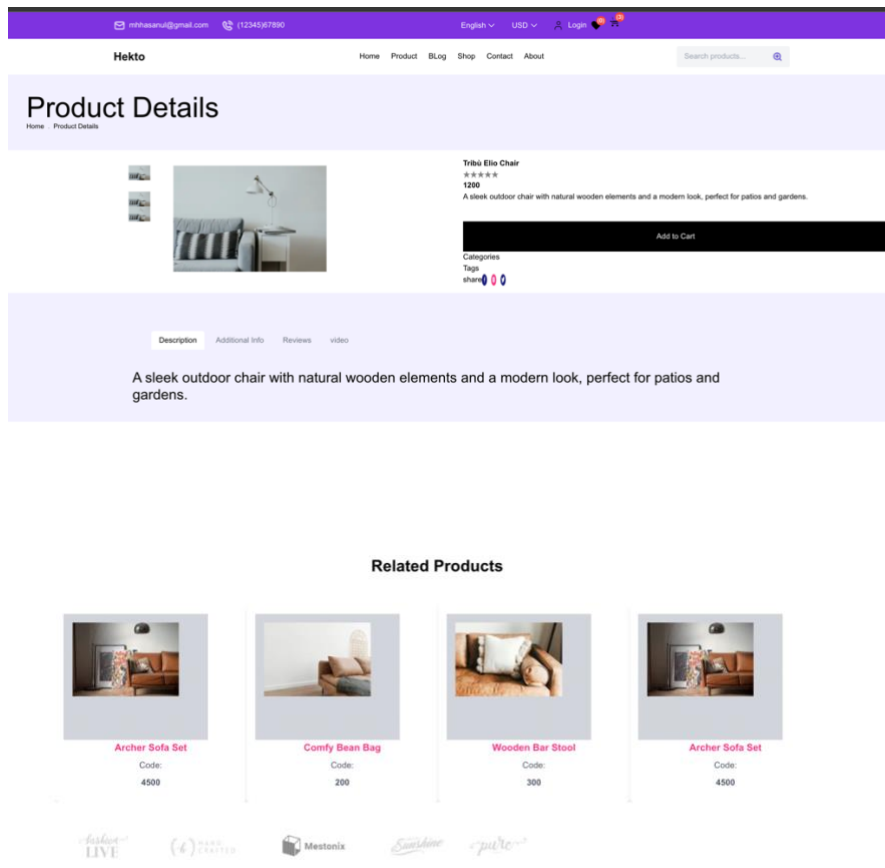
<Image
  src={urlFor(c.image).url()} // Use category.image
  alt={c.name || "Category Image"}
  width={200}
  height={200}
/>
):(
  <p>No image available</p>
)
<div className="h-auto text-center">
  <h3 className="hover:text-pink text-gray-800">{c.name}</h3>
  <Link href={`category/${c._id}`}>
    <p className="underline decoration-2 decoration-blue">
      Read More
    </p>
  </Link>
</div>
</div>
</div>
))
)}
</Slider>
</div>
</section>
);
}

```

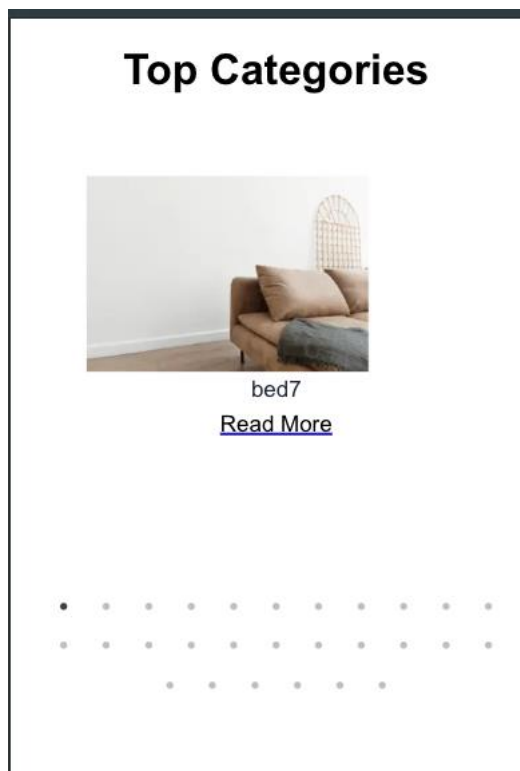
export default Products;



#### 4- Pagination and related product page



- a.
- 5- Components responsive
  - a. Yes it is responsive it just few amendments



b.



- 6- Features included :
- 7- Product listing --- on main page
- 8- Category when clicking on category it directs to product detail page using dynamic routing based on category->[id] and it produces url like  
http:{baseurl}/category/catefory\_id[dynamic]
- 9- When clicking on category href in list product listing page it produces url like  
http{baseurl}/product/productid
- 10- When going to product detail page we have add to cart and wish list option the supported using context api the cart price is display on top
- 11- Search filter when clicking on search button and putting value "chair" it filter in model
- 12- Source code
  - a. <https://github.com/maeyda-masroor/my-appv1/tree/main>
  - b.