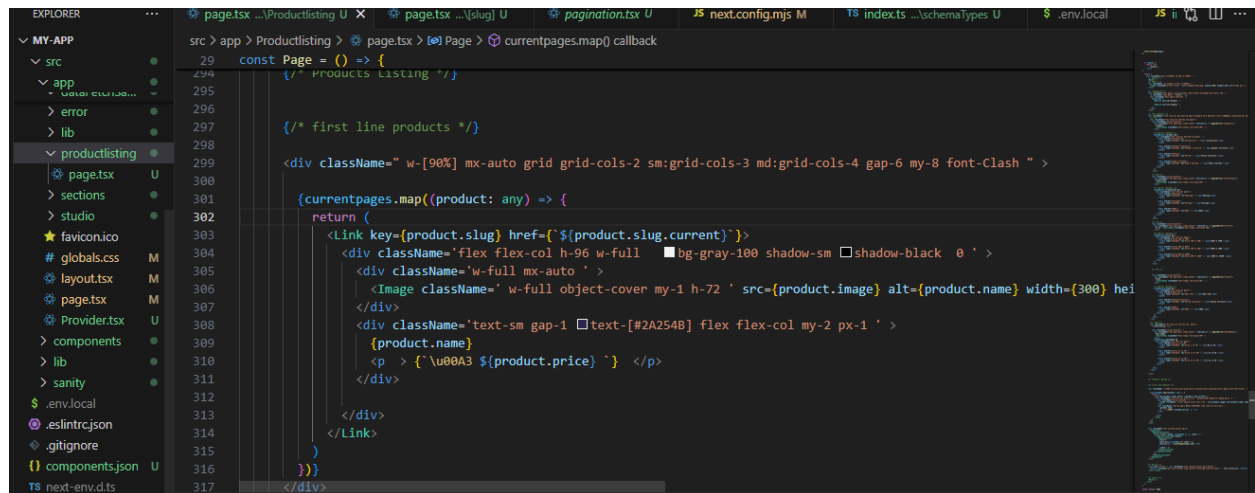


Product Listing :

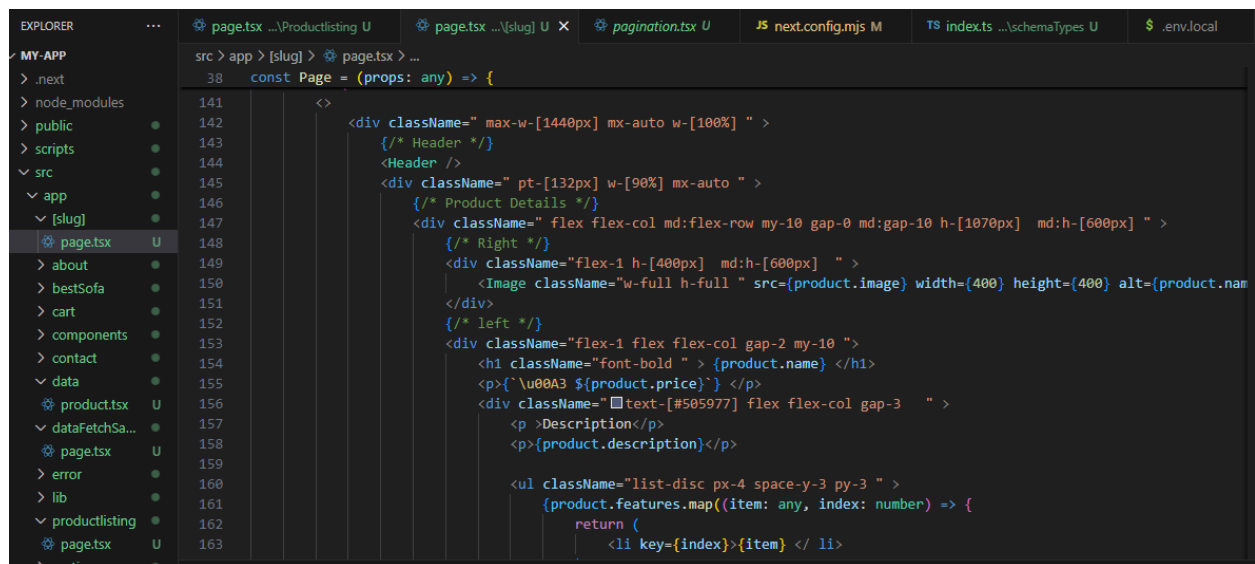


The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'app', 'error', 'lib', 'productlisting', 'sections', 'studio', 'favicon.ico', 'globals.css', 'layout.tsx', 'page.tsx', 'Provider.tsx', 'components', 'lib', 'sanity', '.env.local', '.eslintrc.json', '.gitignore', 'components.json', and 'next-env.d.ts'. The code editor shows the 'page.tsx' file in the 'productlisting' folder. The code defines a 'Page' component that takes 'currentpages.map()' as a prop and returns a JSX element. The JSX element is a grid container with a header and a list of products. Each product is displayed with a link, an image, and a price.

```
const Page = () => {  
  /* Products Listing */  
  
  /* first line products */  
  
  <div className="w-[90%] mx-auto grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 gap-6 my-8 font-Clash ">  
    {currentpages.map((product: any) => {  
      return (  
        <Link key={product.slug} href={`/${product.slug.current}`}>  
          <div className="flex flex-col h-96 w-full bg-gray-100 shadow-sm shadow-black 0 ">  
            <div className="w-full mx-auto ">  
              <Image className="w-full object-cover my-1 h-72 " src={product.image} alt={product.name} width={300} height={300} />  
            </div>  
            <div className="text-sm gap-1 text-[#2A254B] flex flex-col my-2 px-1 ">  
              {product.name}  
              <p> {`\u00A3 ${product.price}`} </p>  
            </div>  
          </div>  
        </Link>  
      )  
    })  
  }  
  </div>  
}
```

I create a component then I list all products which I fetch from sanity through the client.fetch ..

Dynamic Data :



The screenshot shows a VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'app', 'error', 'lib', 'productlisting', 'sections', 'studio', 'favicon.ico', 'globals.css', 'layout.tsx', 'page.tsx', 'Provider.tsx', 'components', 'lib', 'sanity', '.env.local', '.eslintrc.json', '.gitignore', 'components.json', and 'next-env.d.ts'. The code editor shows the 'page.tsx' file in the 'productlisting' folder. The code defines a 'Page' component that takes 'props' as a prop and returns a JSX element. The JSX element is a grid container with a header and a list of products. Each product is displayed with a link, an image, and a price.

```
const Page = (props: any) => {  
  <div className="max-w-[1440px] mx-auto w-[100%] ">  
    /* Header */  
    <Header />  
    <div className="pt-[132px] w-[90%] mx-auto ">  
      /* Product Details */  
      <div className="flex flex-col md:flex-row my-10 gap-0 md:gap-10 h-[1070px] md:h-[600px] ">  
        /* Right */  
        <div className="flex-1 h-[400px] md:h-[600px] ">  
          <Image className="w-full h-full " src={product.image} width={400} height={400} alt={product.name} />  
        </div>  
        /* left */  
        <div className="flex-1 flex flex-col gap-2 my-10 ">  
          <h1 className="font-bold "> {product.name} </h1>  
          <p> {`\u00A3 ${product.price}`} </p>  
          <div className="text-[#505977] flex flex-col gap-3 ">  
            <p>Description</p>  
            <p>{product.description}</p>  
            <ul className="list-disc px-4 space-y-3 py-3 ">  
              {product.features.map((item: any, index: number) => {  
                return (  
                  <li key={index}>{item}</li>  
                )  
              })  
            }  
          </ul>  
        </div>  
      </div>  
    </div>  
  }  
}
```

I created Dynamic routing through the product slug which I generated from products,name and I also define all details in of product in dynamic page ..

Relative Products :

```
src > app > [slug] > page.tsx > ...
38  const Page = (props: any) => {
223
224    <h1 className="font-Clash lg:text-[32px] mt-28 " >You might also love these</h1>
225    <div className="  grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 gap-6 my-5 font-Clash " >
226
227      {Products.slice(0, 4).map((product: any) => {
228        return (
229
230          <Link key={product.slug.current} href={`/${product.slug.current}`}>
231
232            <div className='flex flex-col h-96 w-full  bg-gray-100 shadow-sm shadow-black'
233
234              <div className='w-full mx-auto ' >
235                <Image className=' w-full object-cover my-1 h-72 ' src={product.image} alt=
236              </div>
237
238              <div className='text-sm gap-1 text-[#2A254B] flex flex-col my-2 px-1 ' >
239                {product.name}
240                <p > {product.price} </p>
241              </div>
242            </div>
243          </Link>
244        )
245      })}
246    </div>
247  }
248}
```

I also define You may like products through the slice function in my dynamic page ...

Search Bar:

```
src > app > components > header.tsx > Header
18  function Header() {
114
115    return (
116
117      <div className="max-w-[1440px] mx-auto fixed top-0 w-full  bg-white z-50 " >
118
119        <div className="flex h-[51px]  text-[#22202E] items-center py-4 md:py-0 justify-between px-6 mt-3 text-xl " >
120
121          <div className="flex gap-1 items-center w-1/3 order-2 md:order-1 absolute z-50 right-32 md:static font-Clash " >
122            <div className="" >
123              <input
124                className="w-[150px] text-[14px] rounded-md px-1 border-gray-400 border-[1px]"
125                type="text"
126                placeholder="Search..."
127                value={searchingData}
128                onChange={(e) => setSearchingData(e.target.value)} />
129              <button type="submit" >Search</button>
130            </div>
131          </div>
132        </div>
133      </div>
134    )
135  }
136  <div className="order-2 text-2xl cursor-pointer md:order-1 absolute z-50 block rig
```

I created a search bar functionality using filter product :

Pagination:

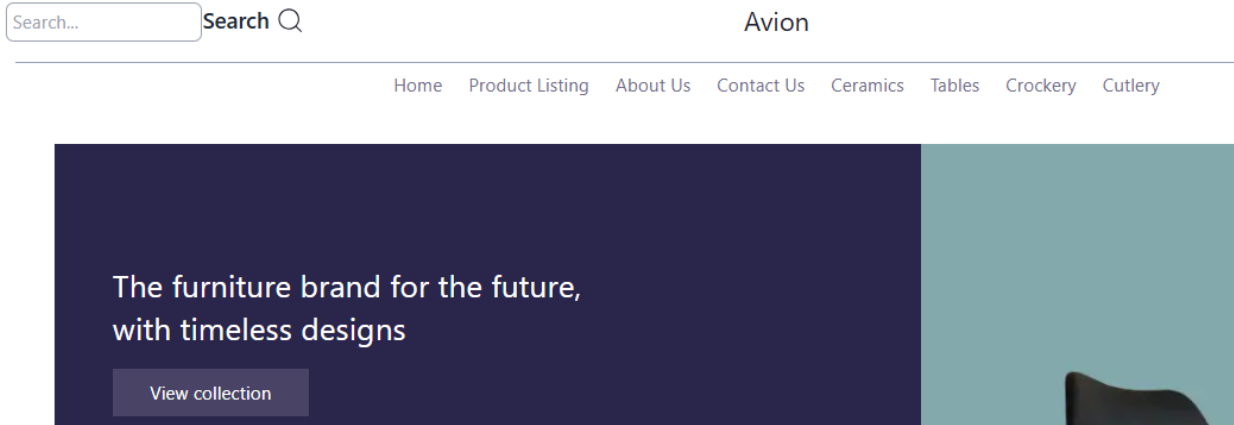
```
src > app > productlisting > page.tsx > Page > currentpages.map() callback
29  const Page = () => {
104    const [showlinks, setShowlinks] = useState(false)
105    const toggle = () => {
106      setShowlinks(!showlinks)
107    }
108
109    // Paggination
110    const ITEMS_PER_PAGE = 12;
111
112    const [currentPage, setCurrentPage] = useState(1);
113    const totalpages = Math.ceil(Data.length / ITEMS_PER_PAGE)
114
115    const currentpages = Data.slice(
116      (currentPage - 1) * ITEMS_PER_PAGE,
117      currentPage * ITEMS_PER_PAGE
118    );
119
120    const handlePageChange = (page: number) => {
121      setCurrentPage(page);
122    };
123
124
125
126
```

I created a extra feture with the help of shadcn ui library . I create a pagination button .. I create a 1 page === 12 products ..and if change a button then new data will render ..

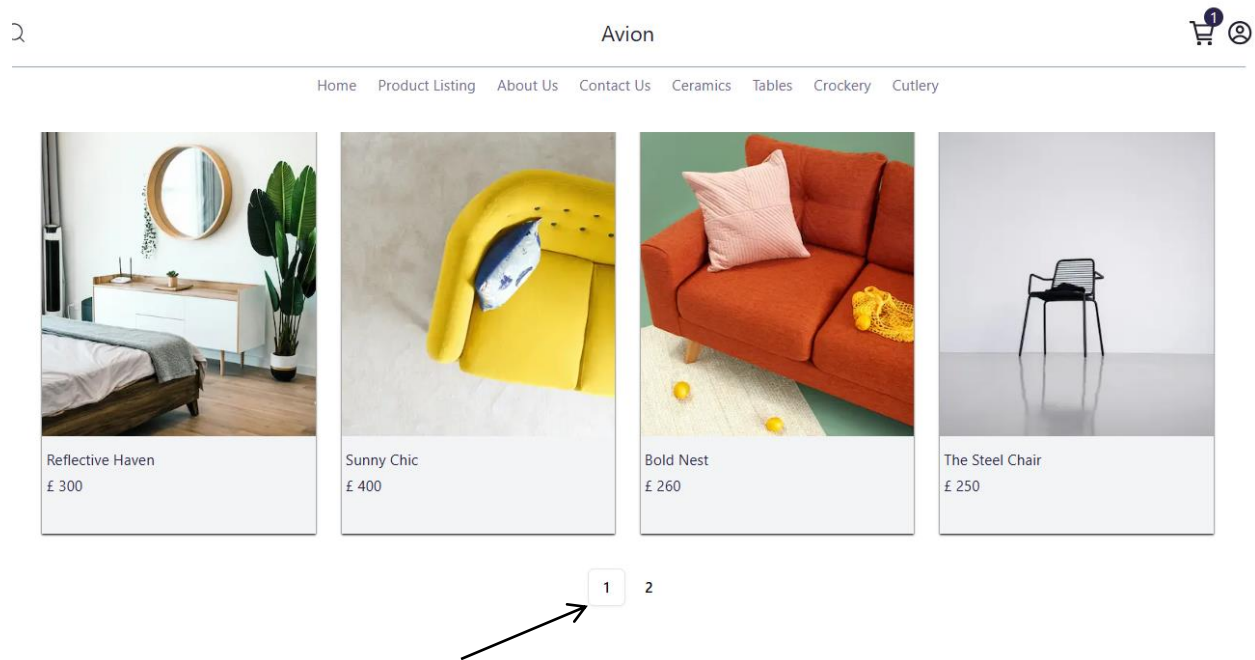
```
src > app > productlisting > page.tsx > Page > currentpages.map() callback
29  const Page = () => {
317    </div>
318
319    <div className="flex justify-center mb-8">
320      <Pagination>
321        <PaginationContent>
322          {Array.from({ length: totalpages }, (_, index) => (
323            <PaginationItem key={index}>
324              <PaginationLink
325                href="#"
326                isActive={currentPage === index + 1}
327                onClick={() => handlePageChange(index + 1)}
328              >
329                {index + 1}
330              </PaginationLink>
331            </PaginationItem>
332          )}}
333        </PaginationContent>
334      </Pagination>
335    </div>
336
337
338
```

Front-end

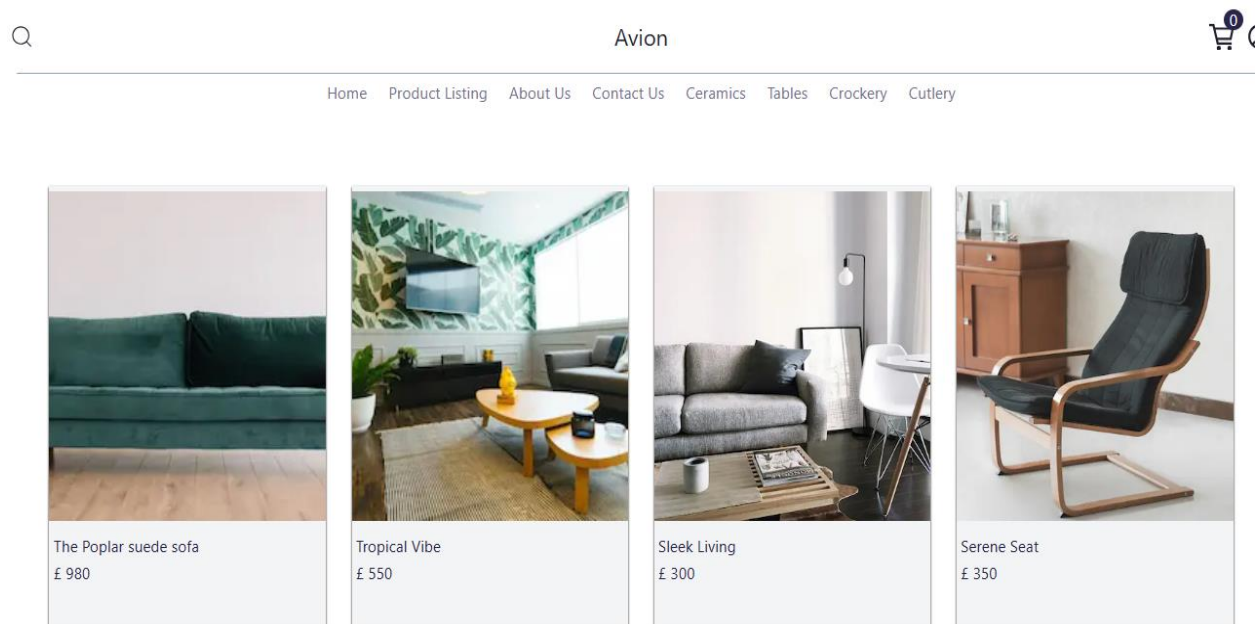
Search Bar :



Pagination:

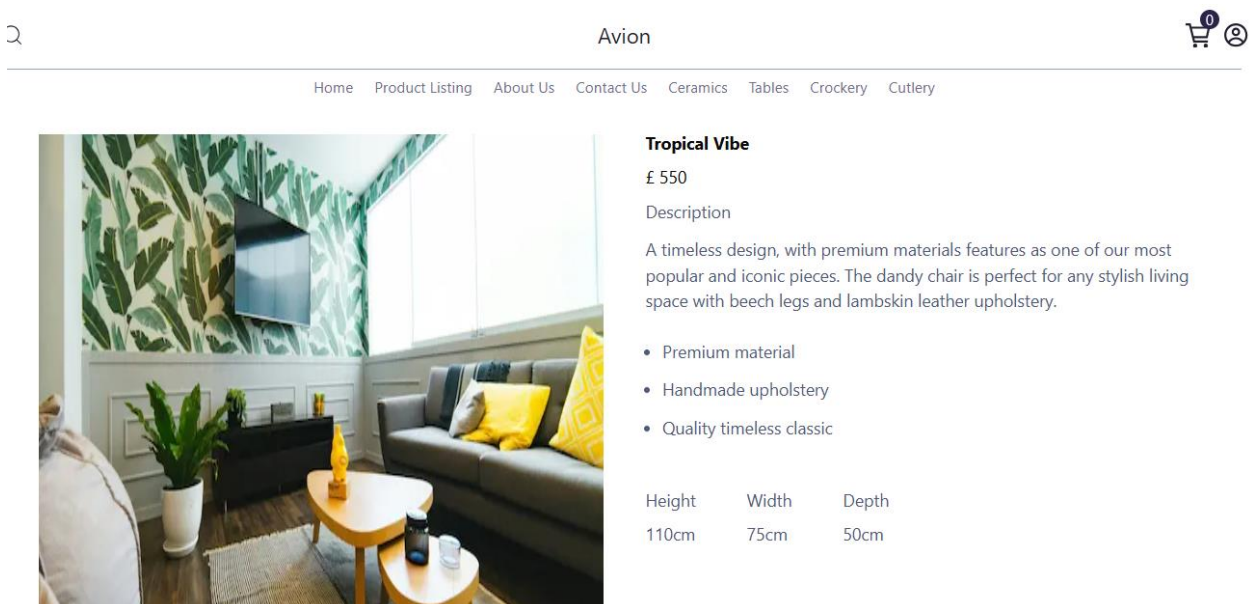


Product Listing:



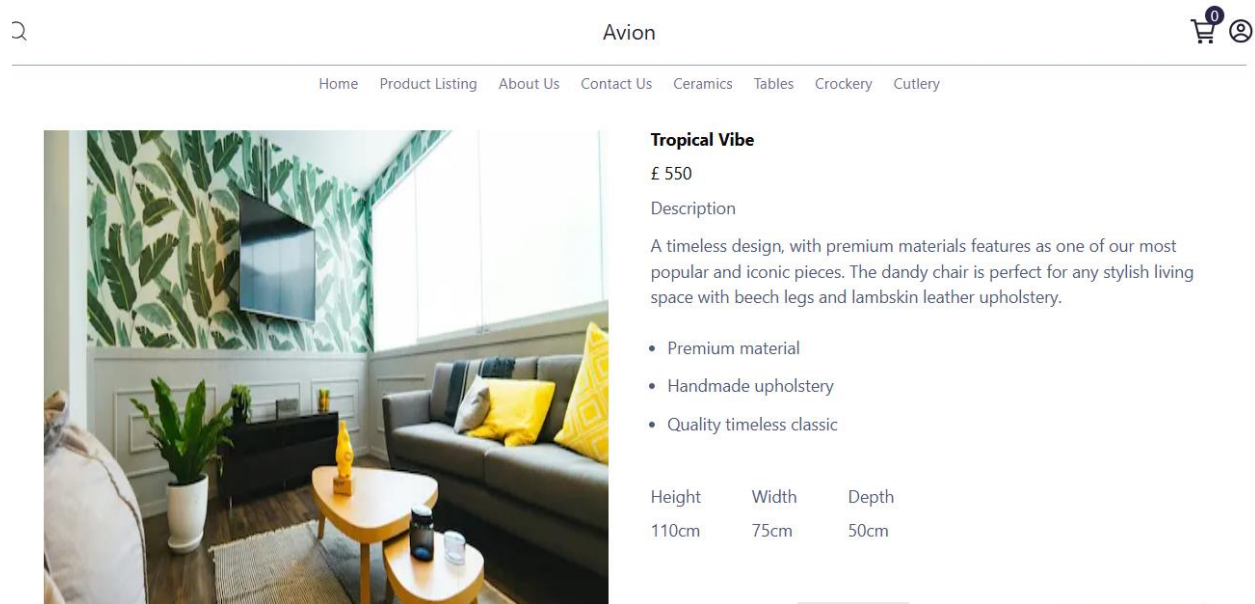
I fetched data from sanity then I list all product in productlisting component through the map function ..

Detail Page of Product:

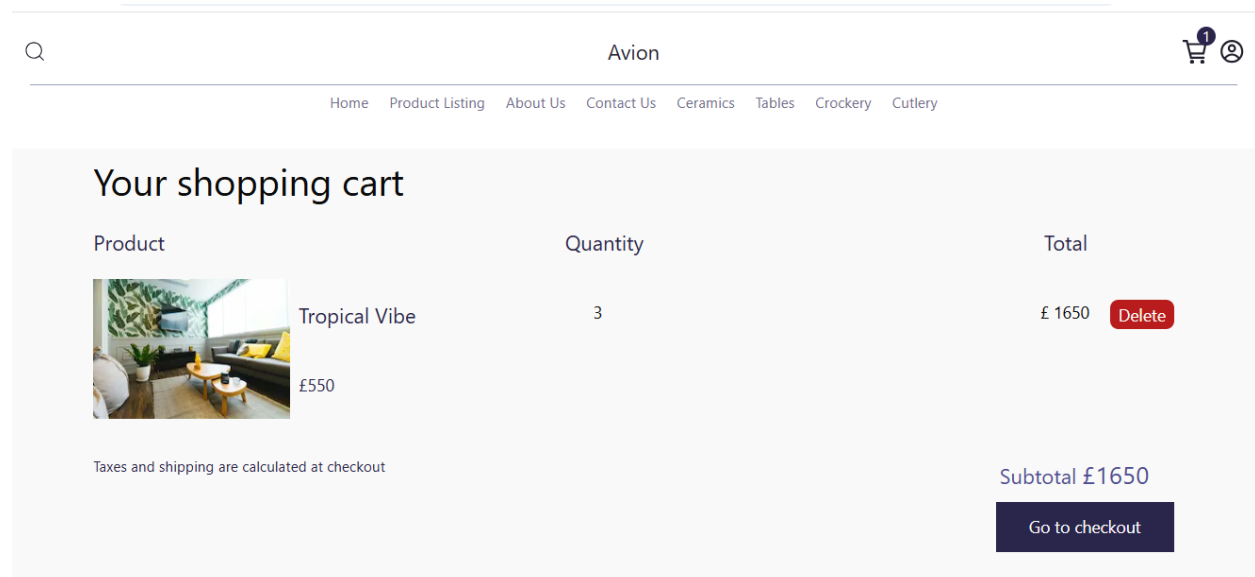


Created detail page of product through the dynamic routing ..

Add To Cart:

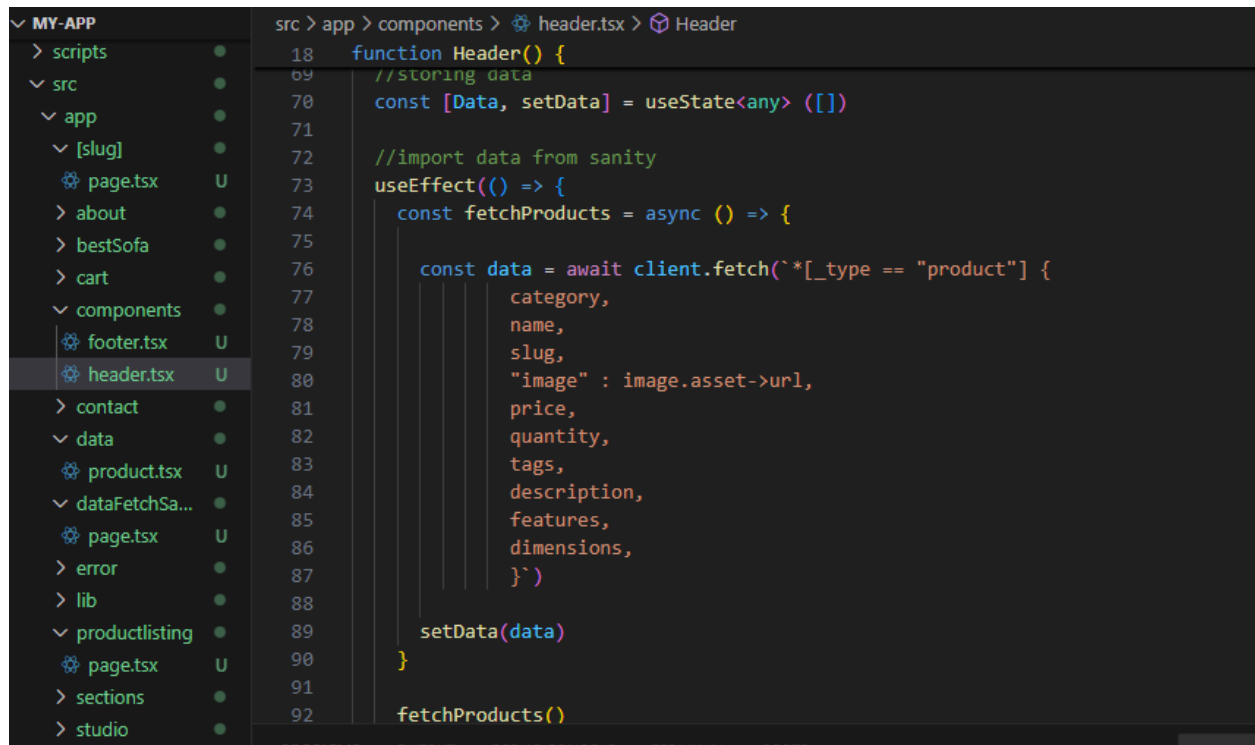


Created add to cart functionality .. I create add to cart functionality through the context api .. If I clicked add to cart button product will be added in cart component and also will the cart length in head of cart icon ..



Added successfully in cart page .. and I also define the delete product ..quantity and Subtotal .. all are automatic ..

Logic Api :



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure for 'MY-APP' with various folders and files. The code editor shows the 'Header' component in 'header.tsx'. The code defines a 'Header' function that uses 'useState' to manage data, 'useEffect' to fetch data from Sanity, and 'fetchProducts' to handle the data fetching logic. The fetched data is then set to the state and the 'fetchProducts' function is called.

```
src > app > components > header.tsx > Header

18  function Header() {
69    //storing data
70    const [Data, setData] = useState<any> ([])
71
72    //import data from sanity
73    useEffect(() => {
74      const fetchProducts = async () => {
75
76        const data = await client.fetch(`*[_type == "product"] {
77          category,
78          name,
79          slug,
80          "image" : image.asset->url,
81          price,
82          quantity,
83          tags,
84          description,
85          features,
86          dimensions,
87        }`)
88
89        setData(data)
90      }
91
92      fetchProducts()
```

Fetches data successfully from sanity .. fetched through the grooq query :