

DAY 4 IMPLEMENTATION REPORT

Introduction:

To create a user-friendly Ecommerce Furniture website where retailer can easily generate payments and manage their sales according to their needs. Using this Ecommerce Platform, customers can purchase and explore a big range of Unique and Trendy equipment at competitive and affordable prices, Ensuring a smooth purchasing experience.

Products list we are offering:

Bedroom beds,
Sofas,
Chairs sets,
Dining tables,
Lamps,
Room Wardrobes etc.

Taken to Build and Integrate Components:

Frontend Development:

(HomePage, collection, contact , about, cart, signUp / signin)

Frontend : Nextjs

Backend : sanity

Authentication : Auth.js.

Responsive : Tailwind css, Css.

Payments Gateway : (Stripe , jazzCash , Easypaisa) After the hackathon.

User Profile : Google , apple , manual signup system. After the hackathon.

Conclusion :

This Ecommerce Furniture website aims to provide a seamless shopping experience for both retailers and customers. With a user-friendly interface, secure authentication, and responsive design, users can explore and purchase a variety of trendy furniture at competitive prices. Built with Next.js, Sanity, and Tailwind CSS, the platform ensures efficiency and smooth navigation. Future integrations, including multiple payment gateways and expanded user profile options, will further enhance the platform's functionality.

1. Product listing



The Poplar suede sofa
980



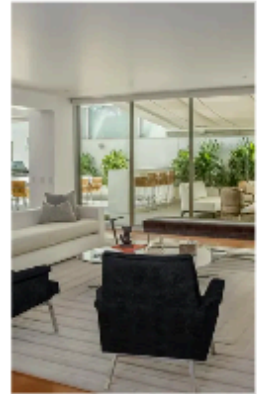
Tropical Vibe
550



Sleek Living
300



Serene Seat
350



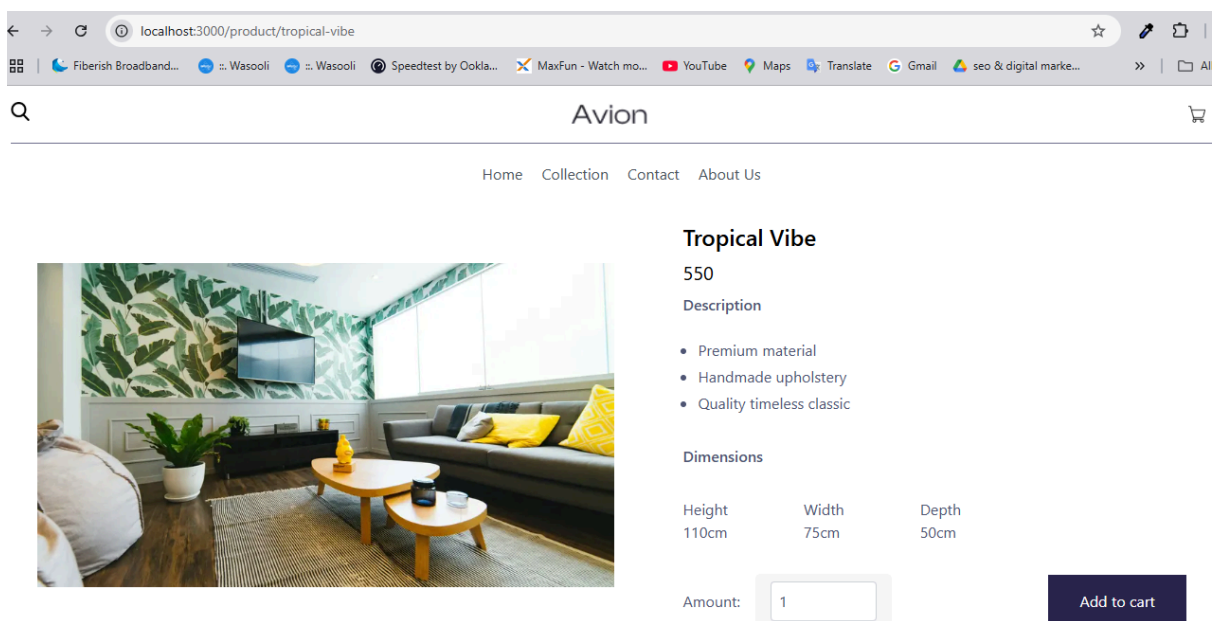
- Product page code

```

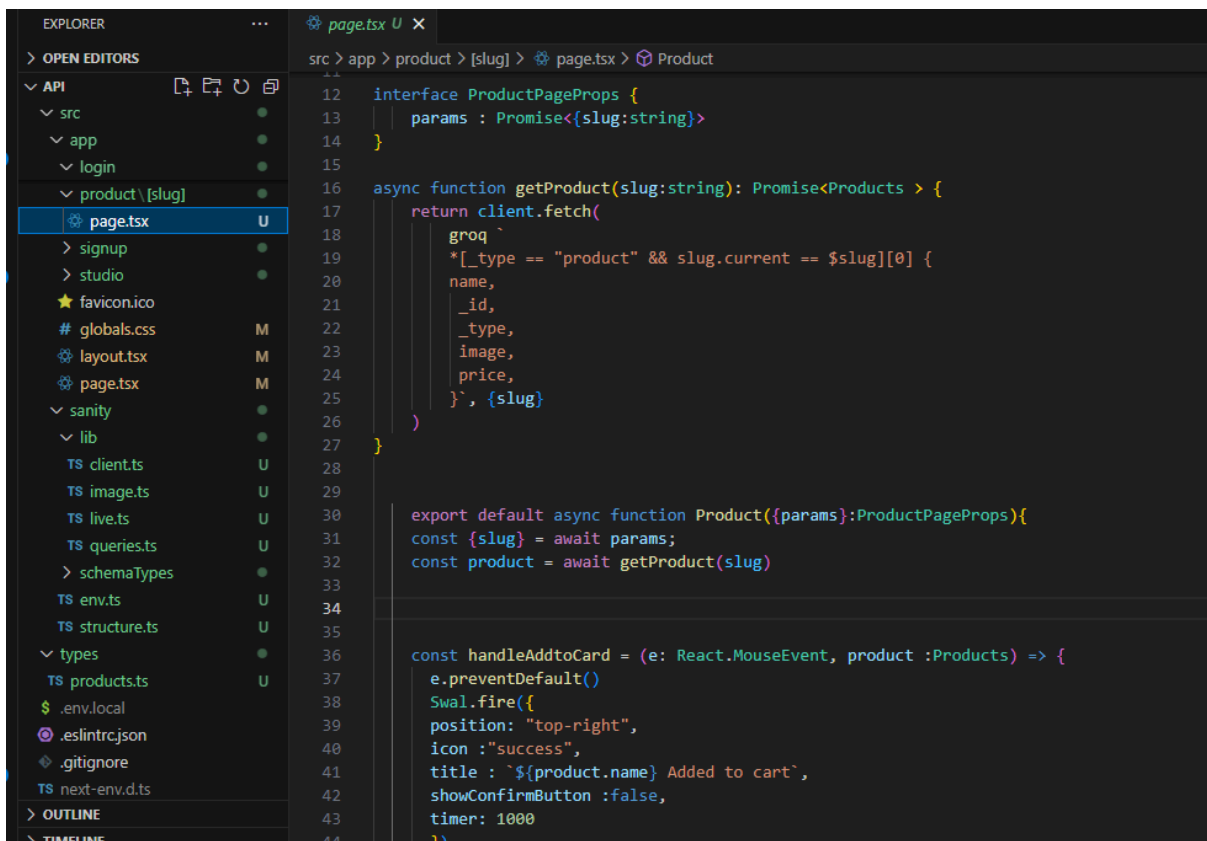
OPEN EDITORS
src > app > allcard > page.tsx > AllCard > filteredProducts.map() callback
12
13
14 const AllCard = () => {
15   const [products, setProducts] = useState<Products[]>([]);
16   const [searchQuery, setSearchQuery] = useState("");
17   const [showSearch, setShowSearch] = useState(false);
18
19   useEffect(() => {
20     async function fetchProducts() {
21       const fetchedProducts: Products[] = await client.fetch(allProducts);
22       setProducts(fetchedProducts);
23     }
24     fetchProducts();
25   }, []);
26
27   // Filtered Products Based on Search Query
28   const filteredProducts = products.filter((product) =>
29     product.name.toLowerCase().includes(searchQuery.toLowerCase())
30   );
31
32   return (
33     <div className="py-[20px]">
34       { /* All Products Banner */ }
35       <Image className="all-product" src={allproduct} alt="all product banner" />
36
37       { /* Search & Filter Section */ }
38       <div className="w-full h-auto bg-pink flex flex-col px-4 py-3 sm:flex-row sm:h-[10vh] sm:justify-between sm:items-center">
39         { /* Filters */ }
40         <ul className="flex flex-wrap w-full sm:w-2/5 justify-around items-center space-y-2 sm:space-y-0 sm:space-x-4">

```

2. Product Detail with Dynamic routing



Dynamic routing page code



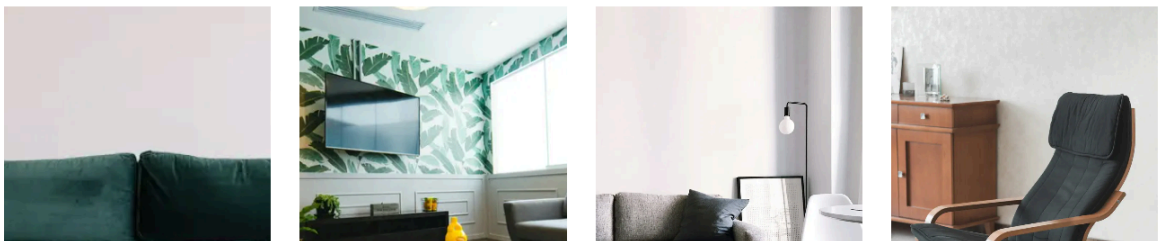
```
12 interface ProductPageProps {
13   params : Promise<{slug:string}>
14 }
15
16 async function getProduct(slug:string): Promise<Products > {
17   return client.fetch(
18     groq `
19     *[_type == "product" && slug.current == $slug][0] {
20       name,
21       _id,
22       _type,
23       image,
24       price,
25     }, {slug}
26   )
27 }
28
29 export default async function Product({params}:ProductPageProps){
30   const {slug} = await params;
31   const product = await getProduct(slug)
32
33
34
35
36   const handleAddtoCard = (e: React.MouseEvent, product :Products) => {
37     e.preventDefault()
38     Swal.fire({
39       position: "top-right",
40       icon : "success",
41       title : `${product.name} Added to cart`,
42       showConfirmButton : false,
43       timer: 1000
44     })
45   }
```

3. Product Search Bar

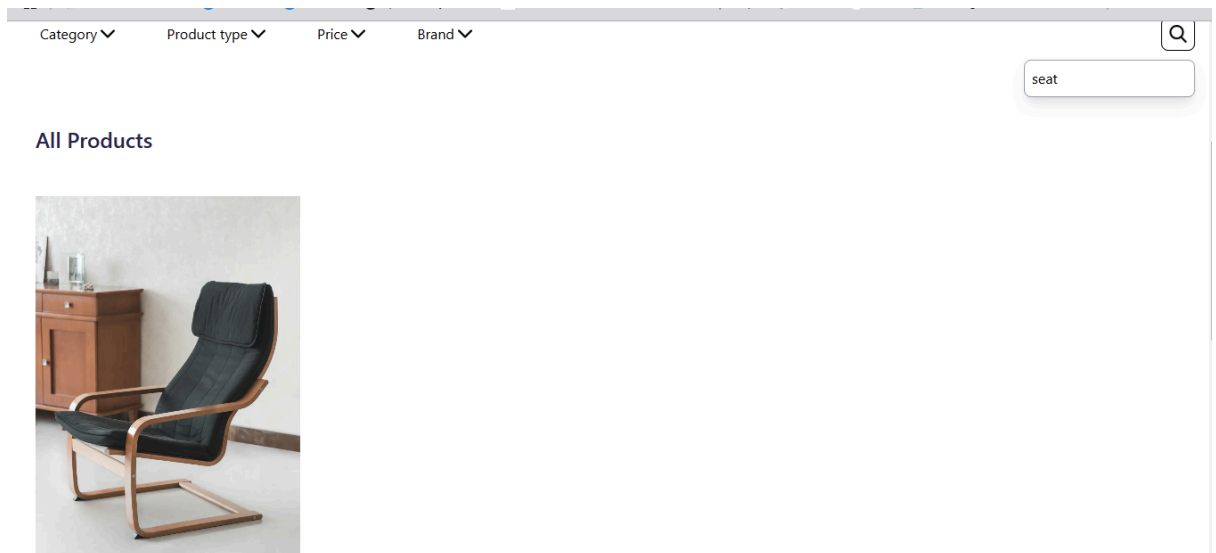
- Before



All Products



- After




Search bar code

```
OPEN EDITORS
src > app > allcard > page.tsx > AllCard > filteredProducts.map() callback
13  const AllCard = () => {
58    /* Search Icon & Bar */
59    <div className="w-full mt-3 sm:mt-0 sm:w-1/5 flex justify-center sm:justify-end relative">
60      /* Search Icon */
61      <button
62        onClick={() => setShowSearch(!showSearch)}
63        className="p-2 border border-black rounded-lg flex items-center space-x-2"
64      >
65        <Image src={searchIcon} alt="search icon" className="w-5 h-5" />
66      </button>
67
68      /* Search Input (Show/Hide on Click) */
69      {showSearch && (
70        <input
71          type="text"
72          placeholder="Search products..."
73          value={searchQuery}
74          onChange={(e) => setSearchQuery(e.target.value)}
75          className="absolute right-0 mt-12 w-48 px-2 py-2 border border-gray-400 rounded-lg focus:outline-non
76            shadow-lg bg-white"
77        />
78      )}
79    </div>
80
81    /* Products Section */
82    <div className="px-4 md:px-8 py-12 text-[#2A254B] mt-6">
83      <h1 className="text-2xl font-semibold">All Products</h1>
84      <div className="grid grid-cols-2 md:grid-cols-4 gap-8 mt-12">
85        {filteredProducts.length > 0 ? (
86          filteredProducts.map((product) => (
87            <div className="w-full h-auto key={product._id}>
88              <Link href={`/product/${product.slug.current}`>
```

4. Cart : add to cart


Shopping Cart



Tropical Vibe
\$550.00

- 1 +


Remove



Serene Seat
\$350.00

- 1 +


Remove



Sleek Living
\$300.00

- 1 +

Remove



The Poplar suede sofa
\$980.00

- 1 +

Remove

Order Summary

Full Name

Enter your full name

Email

Enter your email

Phone No.

Enter your phone number

Address

Enter your address

Subtotal

\$2180.00

Shipping

\$2.00

Tax

\$4.00

Total

\$2186.00

Add to cart Code

5. Notification

- Add to cart notification is working successfully..
- Remove to cart notification is working successfully.
- Confirm order notification is working successfully.

```
const handleCheckout = () => {
  console.log("Order Details: ", userDetails);
  Swal.fire("Success!", "Your order has been placed successfully.", "success");
};
```

```
const handleRemoveCart = (id: string) => {
  Swal.fire({
    title: "Are you sure?",
    text: "You will not be able to recover this item",
    icon: "warning",
    showCancelButton: true,
    confirmButtonColor: "#3085d6",
    cancelButtonColor: "#d33",
    confirmButtonText: "Yes, remove it!",
  }).then((result) => {
    if (result.isConfirmed) {
      removeFromCart(id);
      setCartItems(getCartItems());
      Swal.fire("Removed!", "Item has been removed.", "success");
    }
  });
};
```

6. Order overview

Order Summary

Full Name

muhammad

Email

@gmail.com

Phone No.

00000000000

Address

house 420

Subtotal

\$2180.00

Shipping

\$2.00

Tax

\$4.00

Total

\$2186.00

Checkout

Code

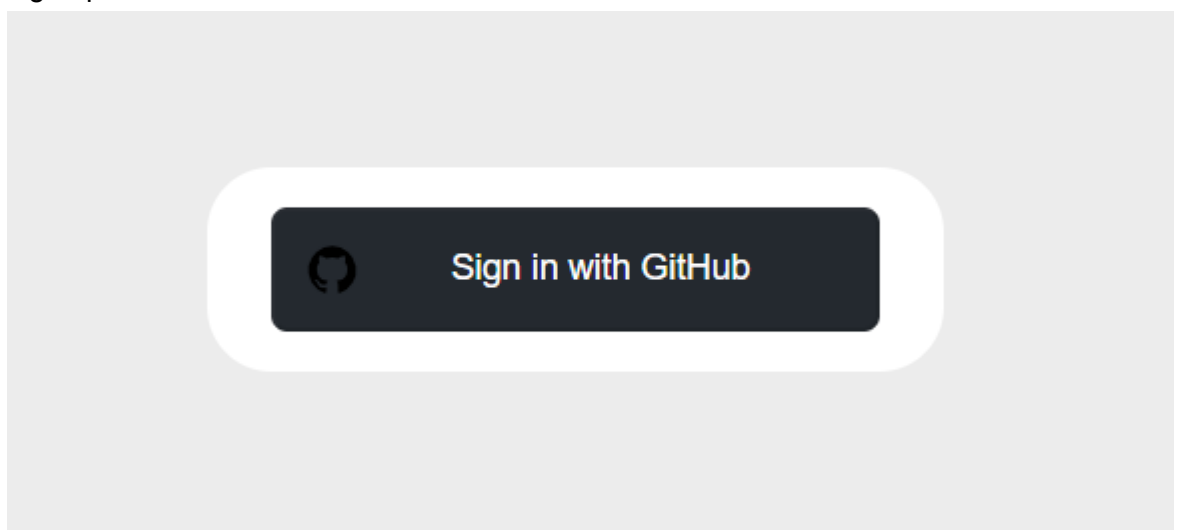
```
value={userDetails.address}
onChange={handleChange}
className="w-full mt-1 px-3 py-2 border rounded-md focus:ring-blue-500 focus:b
placeholder="Enter your address"
rows={3}
></textarea>
</div>
</form>

<div className="mt-6">
  <div className="flex justify-between text-gray-700">
    <p>Subtotal</p>
    <p>${calculatedTotal().toFixed(2)}</p>
  </div>
  <div className="flex justify-between text-gray-700">
    <p>Shipping</p>
    <p>$2.00</p>
  </div>
  <div className="flex justify-between text-gray-700">
    <p>Tax</p>
    <p>$4.00</p>
  </div>
  <div className="flex justify-between font-bold text-gray-900 mt-4">
    <p>Total</p>
    <p>${(calculatedTotal() + 2 + 4).toFixed(2)}</p>
  </div>
</div>

<button
  type="button"
  onClick={handleCheckout}
  className="w-full mt-6 py-2 px-3 border rounded-md text-white bg-gray-800">
```

7. Github authentication

Sign up



Setup for Implementation

Setup

- Sanity all data perfect working on this project.

Component Building

- All components are perfectly responsive and reusable.

State Management

- useState & useEffect using for global and local state.

Task 4 complete