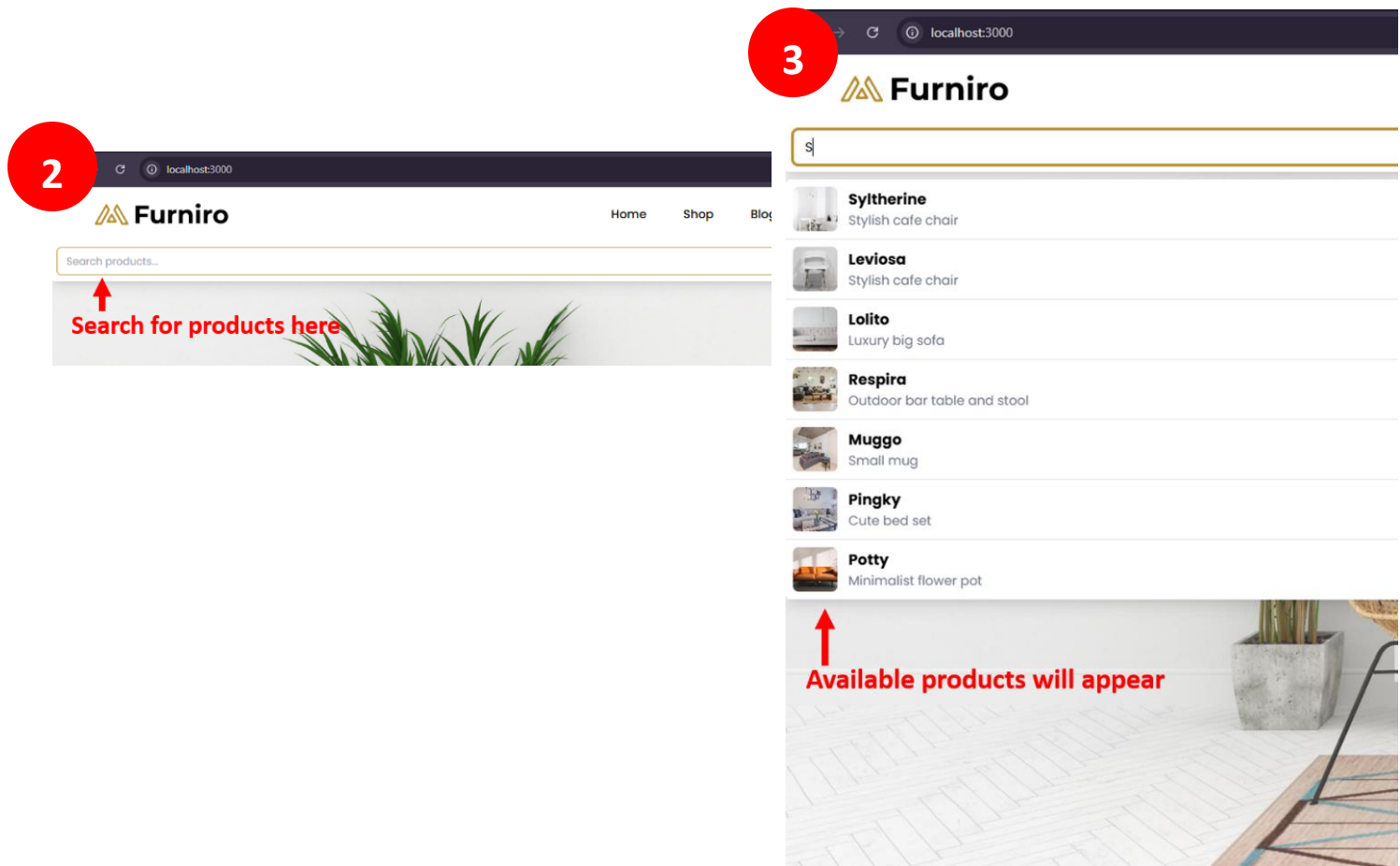
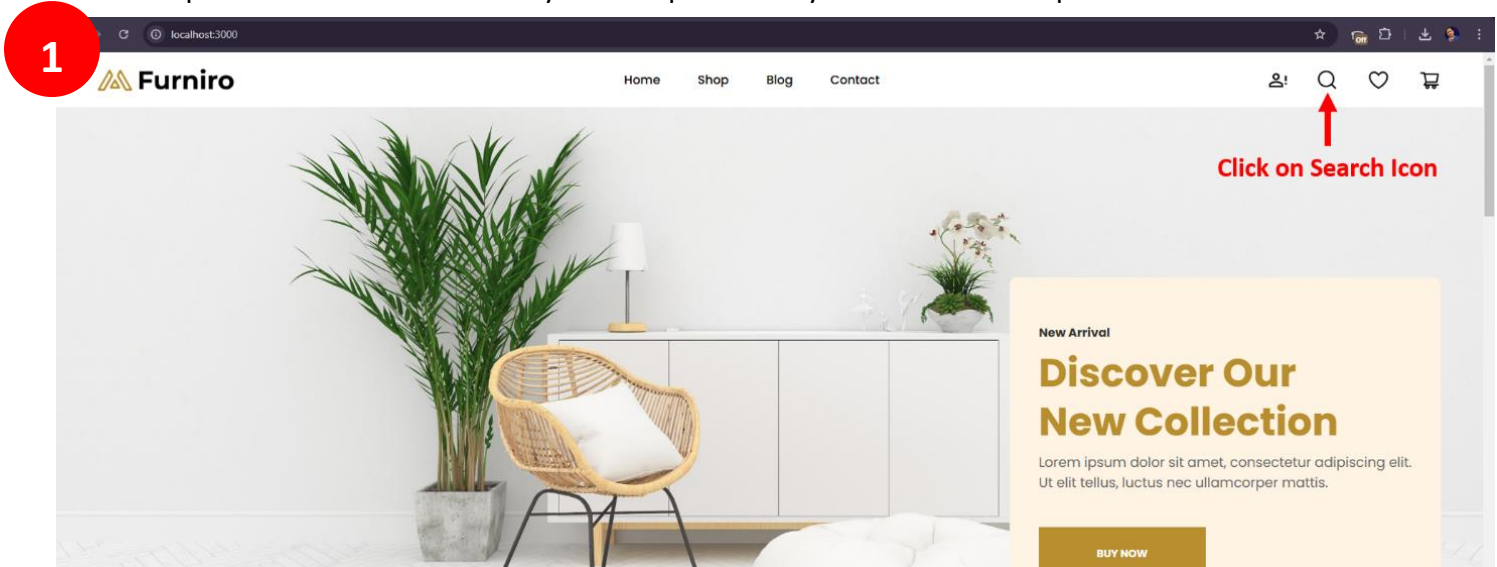


DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE

1. Search Bar:

- ✓ Implement search functionality to filter products by names and description.



Prepared by: MUHAMMAD SALMAN HUSSAIN

Roll #: 00319106

SearchBar.tsx:

```
File Edit Selection View Go Run Terminal Help
UR-UK-Hackathon-NextJS-Design-Jan-2024

EXPLORER
  UR-UK-HACKATHON-NEXTJS-DESIGN-JAN-2024
    > next
    > Day 3 - Report
    > Documentation
    > node_modules
    > notes
    > public
    > screenshots
    > src
      > app
        > about
        > api
        > checkout
        > contact
        > shop
        > stripe
        > studio
        > faviconico
        > globals.css
        > layout.tsx
        > page.tsx
        > assets
        > components
          > cards
          > common
            > Hero.tsx
            > LoadingIndicator.tsx
            > MainButton.tsx
            > NavBar.tsx
            > SearchBar.tsx
            > contact
            > forms
            > sections
            > ui
          > lib
            > actions
            > database
            > json
            > service
            > constants.ts
            > utils.ts
            > validations.ts
            > sanity
            > ...
      > TIMELINE

SearchBar.tsx
src > components > common > SearchBar.tsx > SearchBar > filteredProducts.map() callback
1 import { useState } from "react";
2 import { PRODUCTS, Product } from "@lib/constants";
3
4 const SearchBar = () => {
5   const [searchQuery, setSearchQuery] = useState("");
6   const [filteredProducts, setFilteredProducts] = useState<Product[]>(PRODUCTS);
7
8   const handleSearchChange = (e: React.ChangeEvent<HTMLInputElement>) => {
9     const query = e.target.value;
10    setSearchQuery(query);
11
12    const filtered = PRODUCTS.filter(
13      (product) =>
14        product.title.toLowerCase().includes(query.toLowerCase()) ||
15        product.description.toLowerCase().includes(query.toLowerCase())
16    );
17    setFilteredProducts(filtered);
18  };
19
20   const handleSearchSubmit = (e: React.FormEvent) => {
21     e.preventDefault();
22     console.log("Searching for:", searchQuery);
23   };
24
25   return (
26     <div className="relative w-full max-w-[100%]">
27       <form
28         onSubmit={handleSearchSubmit}
29         className="relative flex items-center w-full bg-white p-2 shadow-lg rounded-md"
30       >
31         <input
32           type="text"
33           placeholder="Search products..."
34           value={searchQuery}
35           onChange={handleSearchChange}
36           className="w-full p-2 pl-3 pr-10 rounded-md border border-[#888e2f] focus:outline-none focus:ring-2 focus:ring-[#888e2f] text-sm"
37         />
38         <button
39           type="submit"
40           className="absolute right-4 top-1/2 transform -translate-y-1/2 p-1 rounded-full bg-[#888e2f] hover:bg-[#a07d24] transition duration-300"
41         >
42           <svg
43             xmlns="http://www.w3.org/2000/svg"
44             className="h-5 w-5 text-white"
45             fill="none"
46             viewBox="0 0 24 24"
47             stroke="currentColor"
48           >
49             <path
50               strokeLinecap="round"
51               stroke="currentColor"
52             />
53           </svg>
54         </button>
55       </form>
56     </div>
57   );
58 }
```