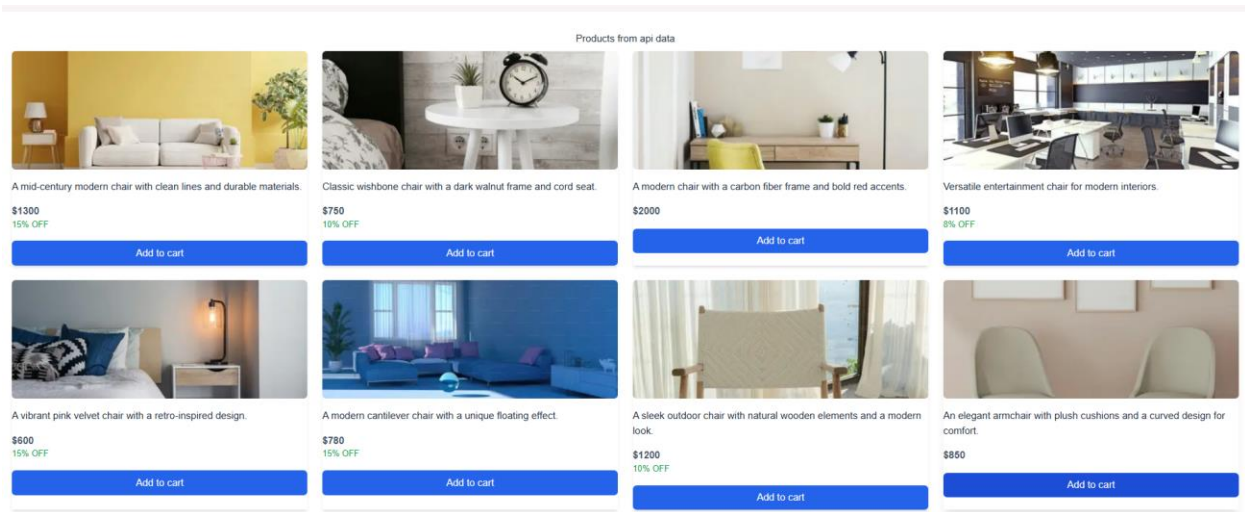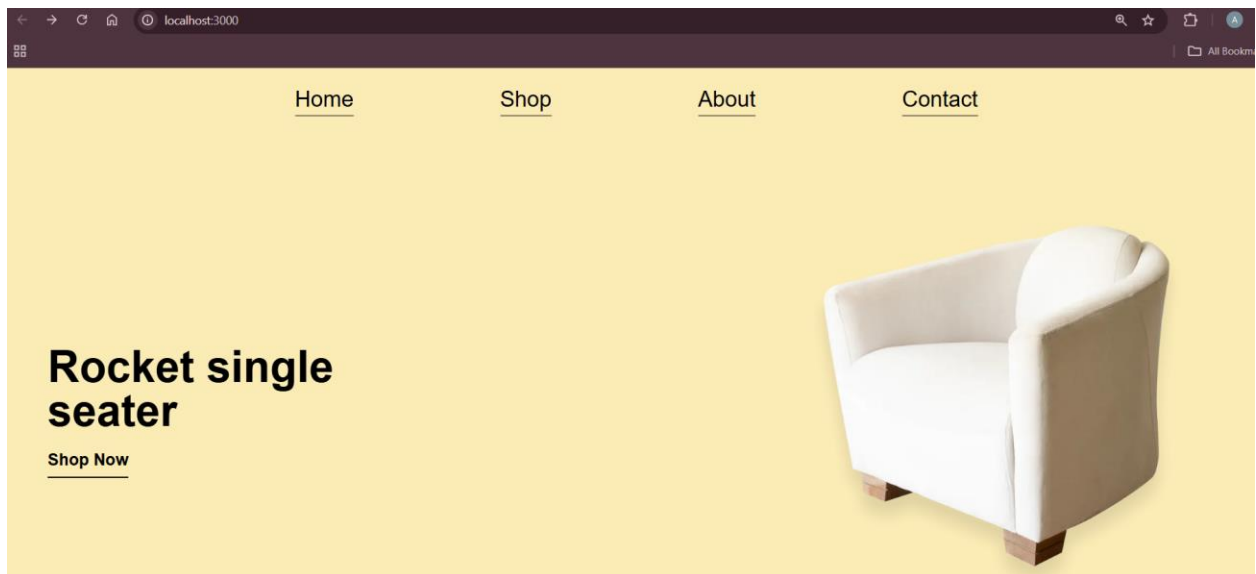# Day 4 - Dynamic Frontend Components - [E-Furniture Store]
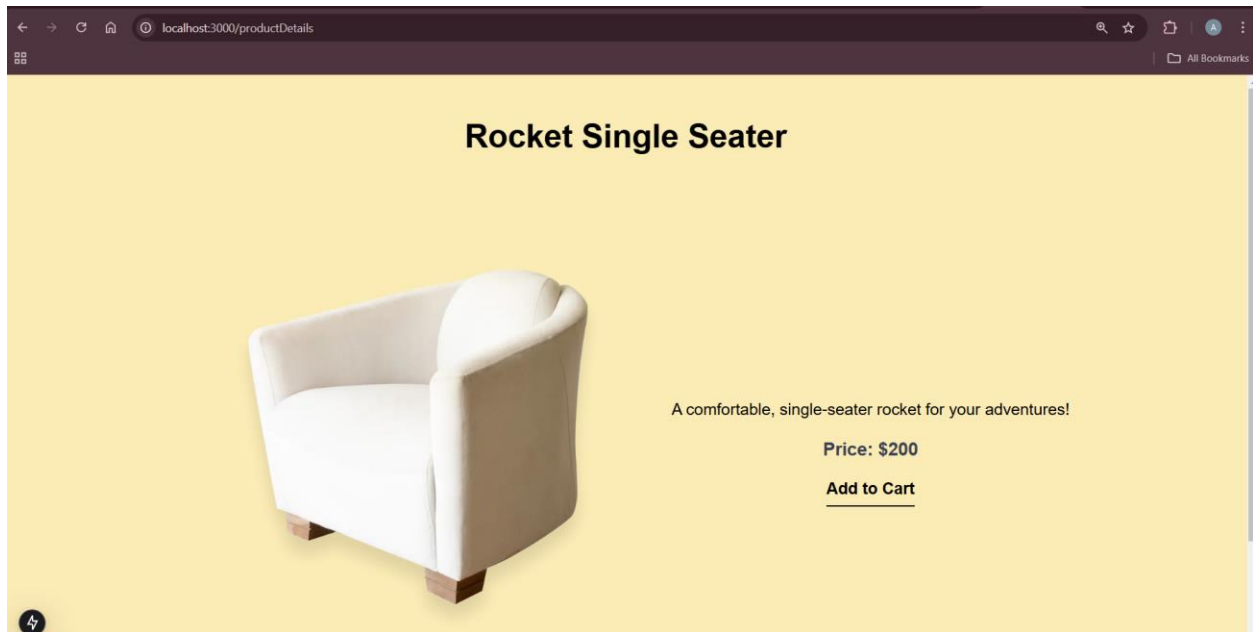
1. Functional Deliverables:
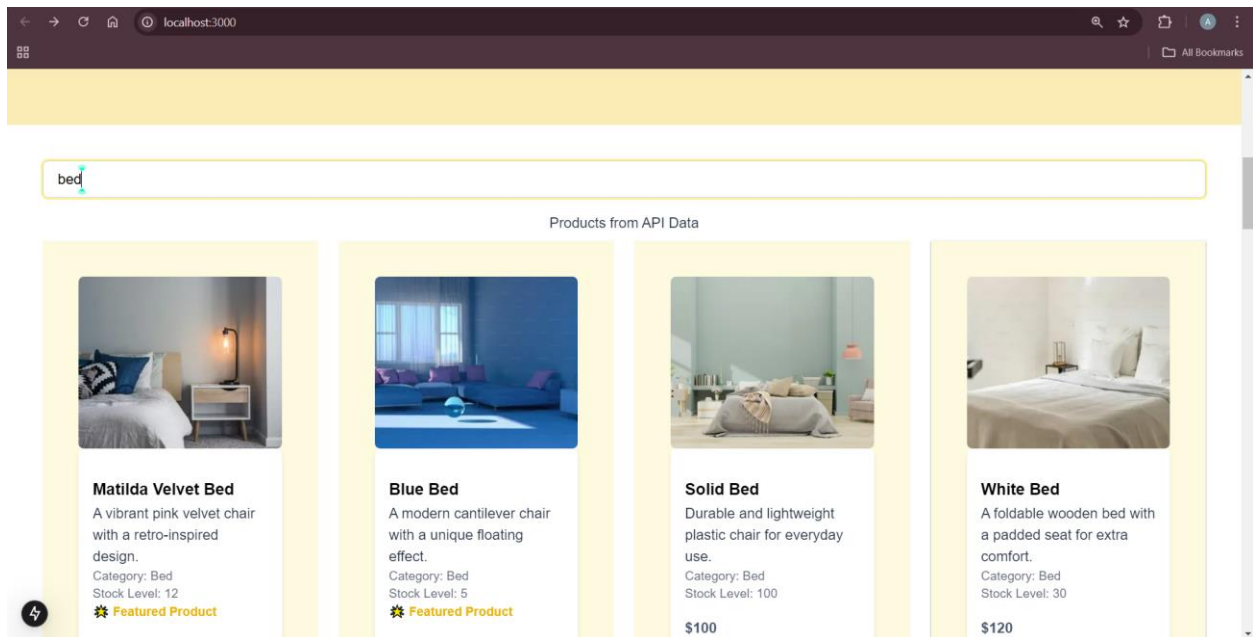
a. Product listing

b. Individual product detail pages with accurate routing and data rendering.

c. Working category filters, search bar, and pagination.

2. Code Deliverables:

a. Search Bar:

```
// SEARCH BAR
const filteredProducts = products.filter((product) =>
  product.name.toLowerCase().includes(searchQuery.toLowerCase()) ||
  product.category.toLowerCase().includes(searchQuery.toLowerCase())
);

useEffect(() => {
  fetchProducts();
}, []);

return (
  <div className="p-10">
    {/*SEARCH BAR */}
    <div className="mb-4">
      <input
        type="text"
        value={searchQuery}
        onChange={(e) => setSearchQuery(e.target.value)}
        placeholder="Search by name or category..."
        className="w-full px-4 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-yellow-200"
      />
    </div>
```

b. Add to Cart:

```
    {/* ADD TO CART */}
    <button
      className="w-full bg-[#a3b18a] text-white py-2 rounded-md hover:bg-[#3a5a40]"
      onClick={() => addToCart(product)}
    >
      Add to Cart
    </button>
    </div>
    </div>
    </div>
  ))}
</div>
```

3. Documentation

a. Steps Taken to Build and Integrate Components:

We started by identifying the key features and designing a modular architecture. Each component was built individually, ensuring proper functionality and usability. Components were then integrated into the application through step-by-step testing and debugging, ensuring seamless communication between them.

b. <u>Challenges faced and solutions implemented:</u>

During development, we encountered issues such as dependency conflicts and layout inconsistencies. Dependency conflicts were resolved by updating or downgrading specific packages to compatible versions. Layout inconsistencies were fixed by standardizing styles and ensuring responsive design principles were followed.