

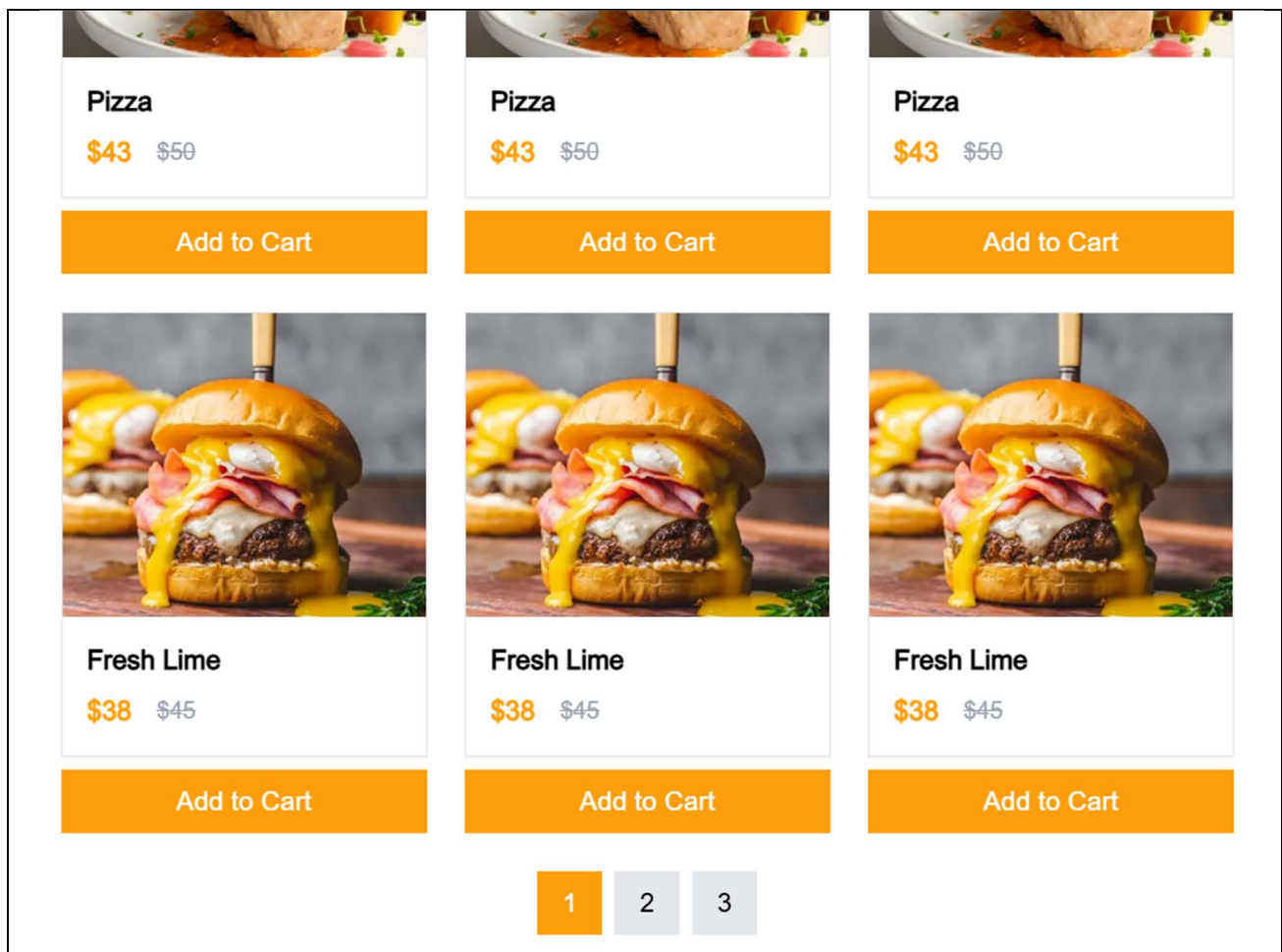
Day 4 Submission Document

Title: Day 4 - Dynamic Frontend Components - [Q-Commerce MarketPlace]

1. Functional Deliverables

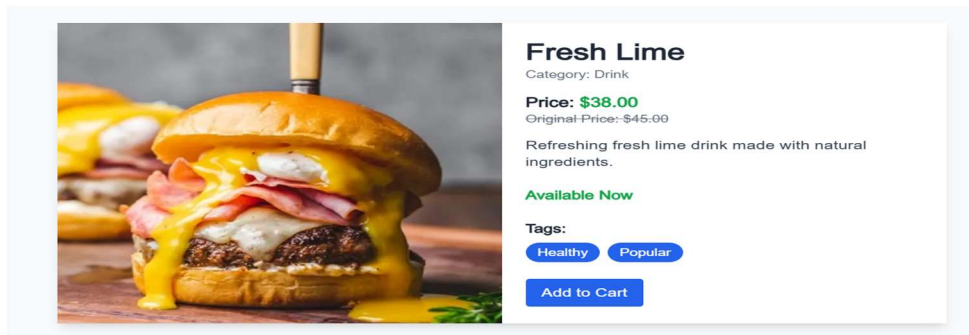
Attach screenshots/screen recordings of your components. Replace [SCREENSHOT] with your images.

Product Listing Page



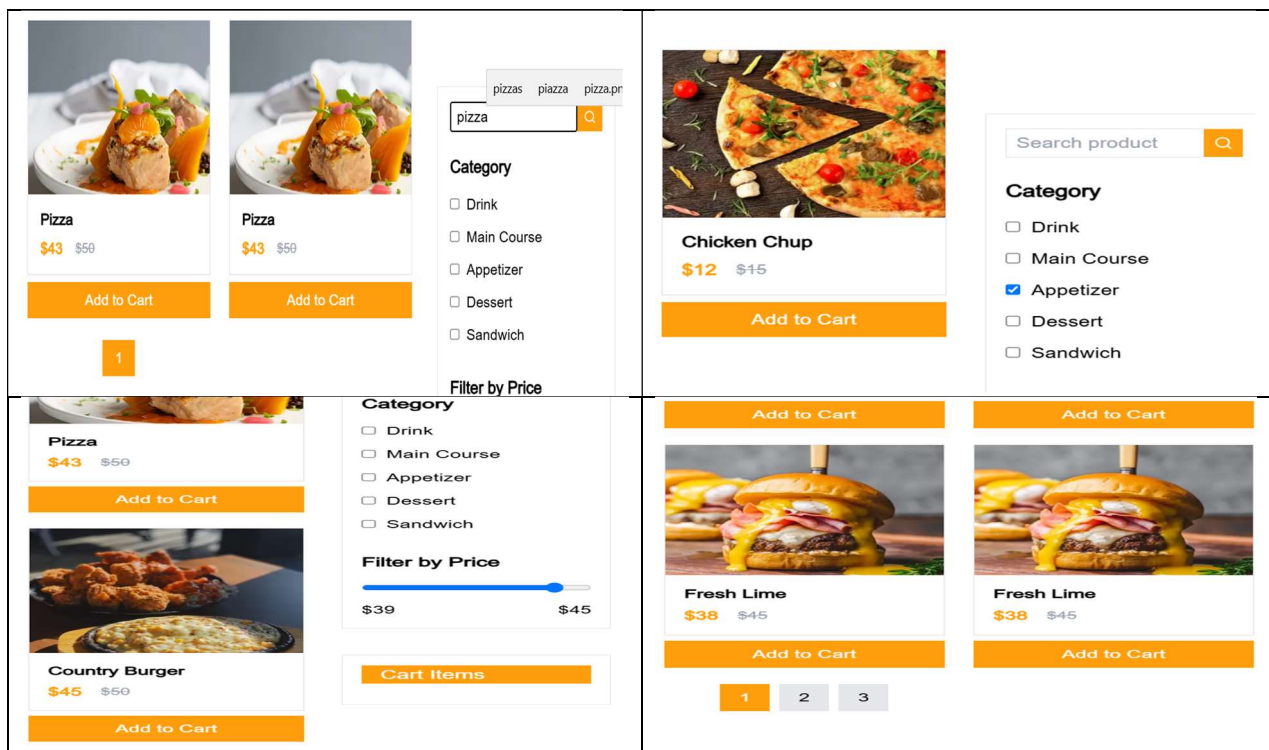
- Fetches product data from Sanity CMS/API.
- Displays products in a responsive grid layout.
- Includes product name, price, image, and stock status.

Product Detail Page








- Uses Next.js dynamic routing (/product/[slug]).
- Shows detailed product information (description, price, sizes/colors).

Search & Filters



- Search functionality filters products by name/tags.
- Category checkboxes and price range slider for filtering.
- Pagination

Cart Component

Shopping Cart				
Product	Price	Quantity	Total	Remove
 Fresh Lime	\$38.00	- 2 +	\$76.00	×
 Chocolate Muffin	\$28.00	- 1 +	\$28.00	×
 Country Burger	\$45.00	- 1 +	\$45.00	×
 Chicken Chup	\$12.00	- 1 +	\$12.00	×
 Burger	\$21.00	- 1 +	\$21.00	×
Total: \$182.00				
Proceed to Checkout				
Clear Cart				

- Displays added items, quantities, and total price.
- Uses React state/context for cart management.

2. Code Deliverables

Include code snippets for critical components.

ProductCard Component

```
const [selectedCategory, setSelectedCategory] = useState<string>('');
const [priceRange, setPriceRange] = useState<number[]>([0, 100]);
const [sortOrder, setSortOrder] = useState<string>('Newest');
const [cart, setCart] = useState<Product[]>([]);
const [currentPage, setCurrentPage] = useState<number>(1);
const productsPerPage = 6;

useEffect(() => {
  async function fetchData() {
    try {
      // Fetch products from the CMS
      const productData = await client.fetch(
        `*[_type == "food"]{
          name,
          price,
          "oldPrice": originalPrice,
          "img": image.asset->url,
          category,
          available,
          slug
        }`
      );
    } catch {
      // Handle error
    }
  }
  fetchData();
}, []);
```

SearchBar Component

```
const handleSearch = (e: React.ChangeEvent<HTMLInputElement>) => {
  setSearchTerm(e.target.value);
};

const handleCategoryChange = (category: string) => {
  setSelectedCategory(category === selectedCategory ? '' : category);
};

const handleSortChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
  setSortOrder(e.target.value);
};

const addToCartBlock = (product: Product) => {
  // Use the functional form of setCart to ensure correct state updates
  setCart((prevCart) => {
    const updatedCart = [...prevCart, product];
    console.log('Cart Updated:', updatedCart); // Log the updated cart for debugging
    return updatedCart;
  });
};
```

3. Documentation

Implementation Steps

1. **Sanity CMS Integration:**
 - Used GROQ queries to fetch product and category data.
 - Created schemas for products in Sanity Studio.
2. **Dynamic Routing:**
 - Implemented use Params for product pages.
3. **State Management:**
 - Utilized React Context API for cart functionality.

Challenges & Solutions

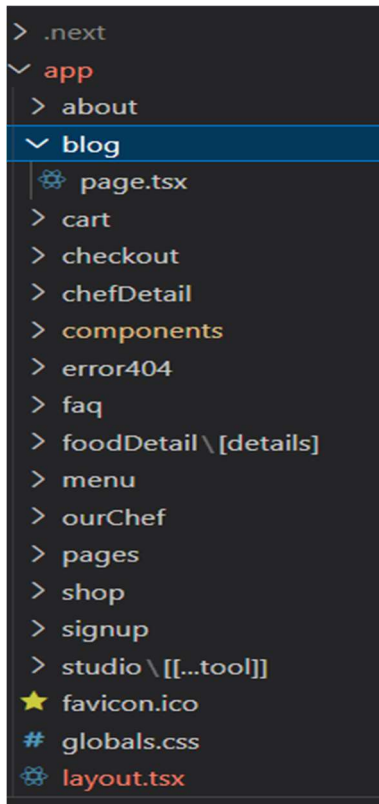
- **Challenge:** Slow API responses.
Solution: Added loading skeletons and pagination.
 - **Challenge:** Cart state persistence.
Solution: Integrated localStorage to save cart items.
-

4. Repository Submission

- **GitHub Link:**

<https://github.com/Muhammadasif1986/UIUX-hackathon---Nextjs-Design-Jam/tree/main/Documents>

- **Folder Structure:**



5. Checklist

- All components are functional.
- Responsive design tested on mobile/desktop.
- Code follows modular architecture.
- Documentation submitted in PDF format.

