Day 4 - Dynamic Frontend Components

Media Mart Functional Deliverables

1. Product Listing Page with Dynamic Data

 Description: The product listing page dynamically displays products using a manually exported JSON dataset. Each product includes its image, name, price, and a brief description, presented in a clean and visually appealing layout to enhance user experience.

2. Individual Product Detail Pages with Accurate Routing and Data Rendering

 Description: Each product detail page dynamically renders detailed information for the selected product, including the title, full description, price, and related images. Leveraging accurate dynamic routing ensures users are directed seamlessly to the correct product page based on their selection.

3. Working Category Filters, and Search Bar

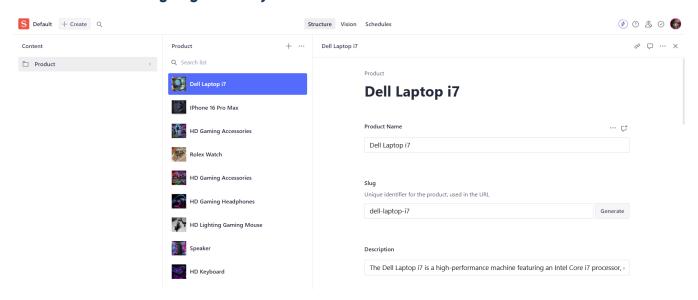
> Screenshots/Recordings:

- Demonstration of category filters.
- Search bar in action.

• Description:

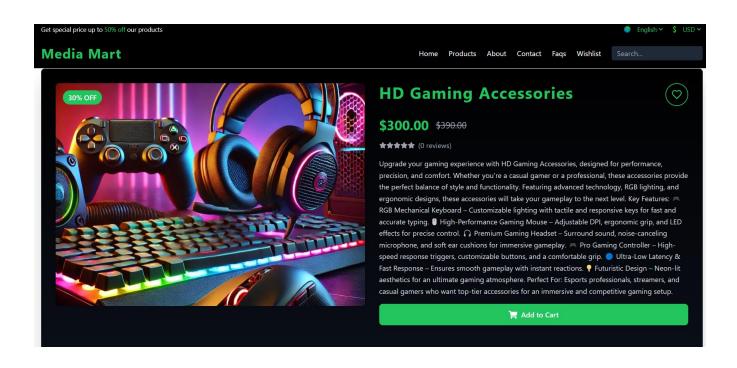
- o Category Filters: Enables filtering of products based on categories like
- "Electronics," "Home Appliances," etc., allowing users to narrow down their preferences.
- Search Bar: Offers real-time search functionality, dynamically
- o filtering products as users type keywords.

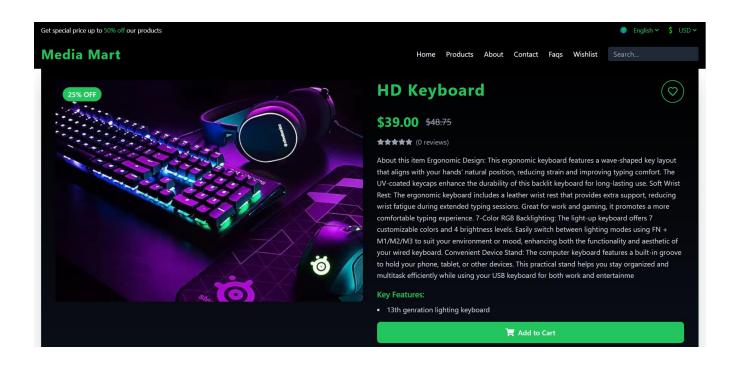
1. Product Listing Page with Dynamic Data



 Description: The product listing page dynamically displays products using a manually exported JSON dataset. Each product includes its image, name, price, and a brief description, presented in an aesthetically pleasing format.

2. Individual Product Detail Pages with Accurate Routing and Data Rendering

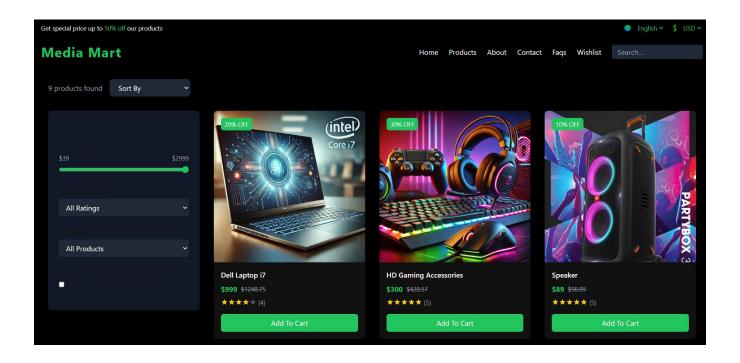


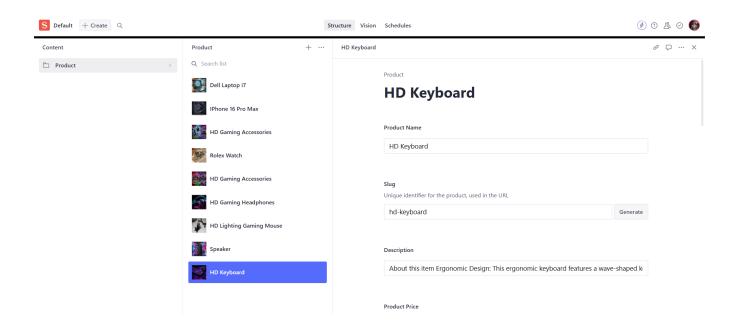


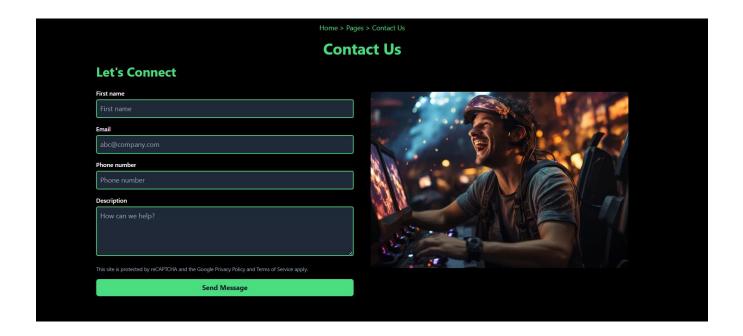


 Description: Each product detail page dynamically renders data for the selected product, including the title, full description, price, and related images. Accurate dynamic routing ensures users are directed to the correct product page based on their selection.

3. Working Category Filters, Search Bar, and Pagination







• Screenshots/Recordings:

- o Demonstration of category filters.
- Search bar in action.

• Description:

- Category Filters: Filters products based on categories like "Electronics," "Home Appliances," etc.
- Search Bar: Provides real-time search functionality, filtering products dynamically as users type.

Code Deliverables

Key Components

Product Card Component

```
▼ File Edit Selection View Go Run Terminal Help
                                                                                                                                               08 🔲 🖃 🖽
                                                                                                                       83
þ
     EXPLORER
    ∨ MART
Q
Ç.
     🗀 арр
      about
      cart
₹

ProductCard.tsx

Drod

       ProductDetails.tsx
       Products.tsx
                                        product: {
    id: string;
    name: string;
    price: number;
    images: ProductImage[];
    description: string;
    slug: {
        current: string;
    }
}
      ■ faqs
                                ☑ node + ∨ Ⅲ 🛍 ··· ^ ×
      > OUTLINE
    > LIVE SHARE
Ln 24, Col 27 (14 selected) Spaces: 2 UTF-8 CRLF () TypeScript JSX 🖗 Go Live 🔠 Codeium: () 🗘 🗘 tabnine basic
```

Review Component

```
📢 File Edit Selection View Go Run Terminal Help
                                                                                                                                                                                                                                                                 æ,
                                                                                                                                                                                                                                                                                                                     08 🔲 🖃 🖽
                                                                                                                                                                                                                                                                 ☐ snipcart.d.ts U   ● ProductCard.tsx M   ● Pop 🖏 🔲 · ·
ф
                                         T; O = ... src > app > components > % ProductDetailstsx > (@) ProductDetails
44 const ProductDetail3: React.Fc@Props> = ({ product })) => {
139 const handlesUbmit = async (e: React.FormEvent) => {
186 setisSubmitting(false);
         ∨ MART
         m public
 9
0
38
              cart
                                                                                           setName(review.name);
<del>L</del>
                                                                                        setRating(review.comment);
setRating(review.rating);
setEditingReview(review.id);
};
                 # Header.tsx
                                                                                         const handleDelete = (reviewId: string) => {
    setReviews(prev => prev.filter(review => review.id !== reviewId));
8

    Popular.tsx

                                                                                         const renderStars = (rating: number) => {
  const stars = [];
  const fullStars = Math.floor(rating);
  const hasHalfStar = rating % 1 !== 0;
                 Products.tsx
                Special.tsx
              contact
                                                                                           for (let i = 0; i < fullStars; i++) {
| stars.push(<FaStar key=[`star-${i}`} className="□text-yellow-500" />);
              context
              products
                                                                                                                                                                                                                                                                                                                       ☑ node + ~ 🏻 🛍 ··· ^ ×
                                                                       GET /favicon.ico 200 in 24ms
GET /favicon.ico 200 in 36ms
GET /favicon.ico 200 in 28ms
O compiling /studio/[[...tool.]] ...
/ Compiled /studio/[[...tool.]] in 24.7s (7815 modules)
GET /studio/vision 200 in 37566ms
O compiling /contact ...
/ Compiled /contact in 14.5s (7788 modules)
              studio
         OUTLINE
         > TIMELINE
          > LIVE SHARE
```

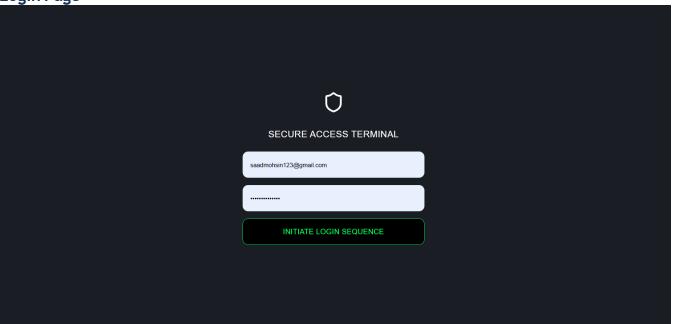
Whish List Component

```
Ф
              ∨ MART
                  🗀 арр
                                                                                                                    import { useWishlist } from '../context/WishlistContext';
import { useEffect, useState } from 'react';
import { client } from '@/sanity/lib/client';
import Import Impage from 'next/image';
import { urlFor } from '@/sanity/lib/image';
import Link from 'next/link';
import { useCart } from '../context/CartContext';
import { motion } from 'framer-motion';
import { Motion } from 'reach from 'lucide-react';
// import Header from '../components/Header';
interface BrowductTumage.
  9
0
38
                      Special.tsx
                    contact
                      context
                    products
                      [slug]
                                                                                                                       interface ProductImage {
    _type: string;
                         page.tsx
                                                                                                                           asset: {
    _ref: string;
                     studio
                                                                                                                                   type: string;
                                                                                                                       interface Product {
    id: string;
                      ★ favicon.ico
♥ globals.css
                                                                                                                           name: string;
price: number;
image: ProductImage[];
                                                                                                                                                                                                                                                                                                                                                                                                                                                 ☑ node + ~ Ⅲ 🛍 ··· ^ :
                                                                                                   GET /favicon.ico 200 in 24ms
GET /products 200 in 636ms
GET /favicon.ico 200 in 78ms
o Compiling /studio/[[...tool]] ...
/ Compiled /studio/[[...tool]] in 24.7s (7815 modules)
GET /studio/vision 200 in 37566ms
                       ■ live.ts
                      schemas
                                                                                                     GET /studio/vision 200 in 37566ms
o Compiling /contact ...
✓ Compiled /contact in 14.5s (7788 modules)
             > OUTLINE
```

Admin Dashboard Component



Login Page



Documentation:

Steps Taken to Build and Integrate Components

1. Data Preparation:

 Created a JSON dataset with fields such as id, name, price, image, and category.

2. Component Development:

- Built reusable components, including Product Card, Product List, and Search Bar.
- Developed additional components like cart, add-to-cart, and user profile sections.

3. **Dynamic Routing:**

o Implemented Next. js dynamic routing for product detail pages.

4. UI/UX Enhancements:

- Added category filters, search functionality, and pagination.
- Styled components using Tailwind CSS for responsiveness and consistency.

Challenges Faced and Solutions Implemented

1. Efficiently Rendering Large Datasets:

 Solution: Implemented pagination to improve performance and load times.

2. Maintaining Consistent Routing and Data Handling:

 Solution: Utilized Next.js dynamic routing and centralized data in a JSON file.

3. Creating Responsive and User-Friendly Components:

 Solution: Applied Tailwind CSS for styling and ensured a responsive design for various devices.

Best Practices Followed

- Adopted a modular component structure for code reusability.
- Used TypeScript for enhanced type safety and error prevention.
- Ensured responsive designs for optimal user experience on both mobile and desktop.

Components Developed for "Media Mart"

- Accordion
- Product Card
- Product List
- o Search Bar
- o Image Gallery
- Add-to-Cart
- Wish List
- o Pagination
- Header
- Footer
- o Admin Dashboard
- o Login
- Payment Method(Snip Cart)

Conclusion

The Media Mart e-commerce platform demonstrates the power of dynamic frontend components in creating an engaging and functional user experience. By utilizing reusable components, efficient data handling, and responsive design principles, the platform successfully combines usability and performance. Key features such as dynamic routing, category filters, search functionality, and pagination ensure smooth navigation and interaction for users. The challenges encountered during development were effectively addressed through best practices, making the platform robust and scalable. Media Mart is a testament to the effectiveness of modern front-end technologies and thoughtful design in crafting user-centric digital solutions.

Best Regards

Web G Media Agency teams