

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

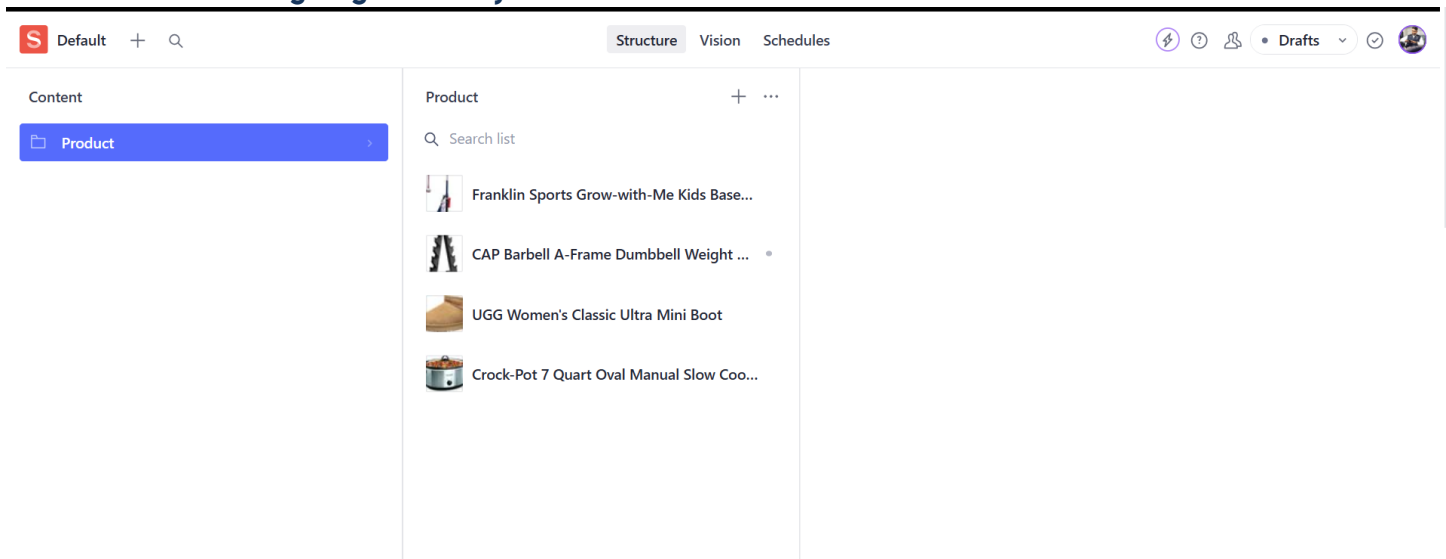
➤ *Screenshots/Recordings:*

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

- **Description:**


- 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 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





Smart watch

\$600.00

★★★★★ (0 reviews)

This sleek and modern smartwatch features a black round dial with a vibrant digital display, showing time, notifications, and health stats. The premium silicone strap ensures maximum comfort, while its sporty yet elegant design makes it an ideal accessory for both fitness enthusiasts and tech-savvy individuals.


 **Add to Cart**



20% OFF



Elegant Home Décor



\$36000.00

\$43200.00

★★★★★ (0 reviews)

Create a warm and stylish atmosphere with this modern home décor setup. Featuring luxurious wall art, decorative vases with fresh flowers, stylish cushions, and a sleek coffee table with scented candles, this setting brings a perfect blend of comfort and sophistication. The neutral color palette is complemented by gold and wooden accents, while the soft rugs, warm lighting, and artistic elements make this space a true reflection of elegance and coziness. Perfect for those who love minimalist yet high-end décor to elevate their home interior.

 Add to Cart

10% OFF



Sleek Modern Laptop on Minimalist Desk



\$405.00

\$445.50

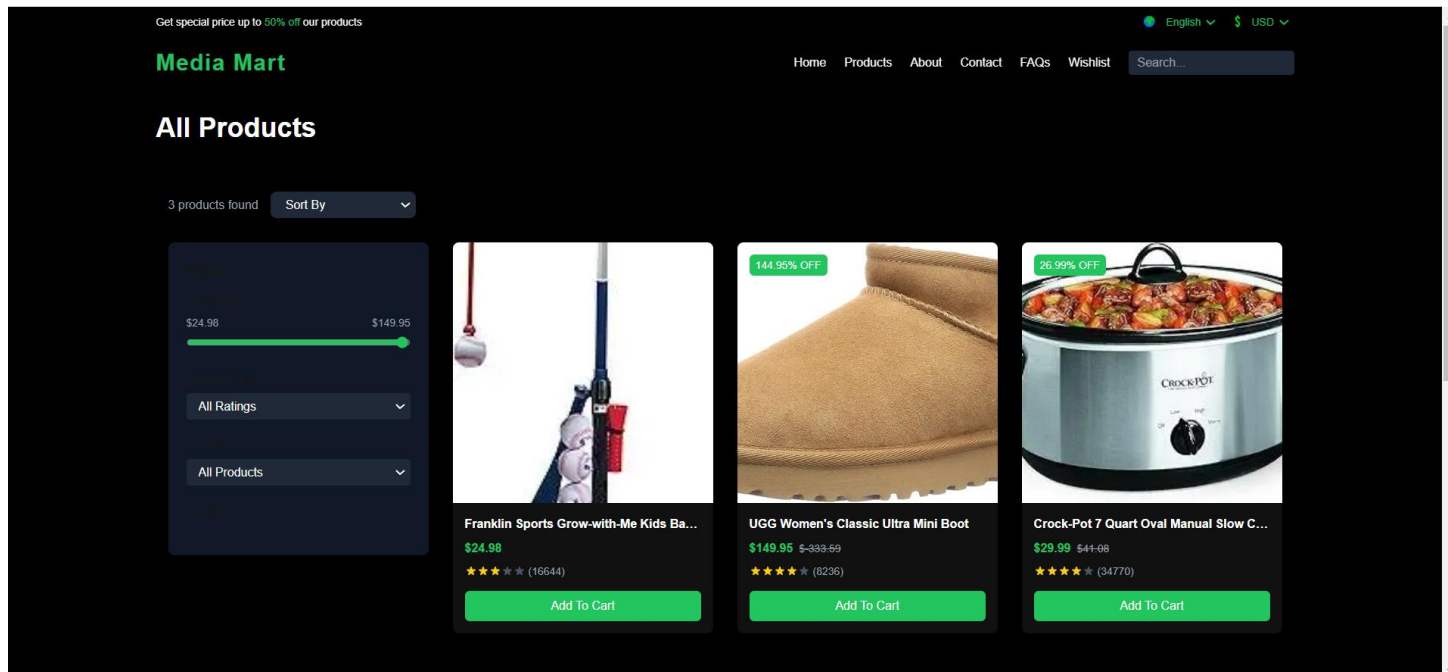
★★★★★ (0 reviews)

This high-definition image showcases a sleek, matte black laptop with a large, high-resolution display, set on a minimalist desk. The clean workspace features a few essential items: a coffee cup, a mouse, and a small plant, creating an environment of focus and productivity. Ideal for work, study, or creative projects, the laptop's design emphasizes modernity and efficiency.

 Add to Cart

- **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



S

Default

+

🔍

StructureVisionSchedules

⚡🔍👤• Drafts

👤🔍

Content


Product


Product


+


...

🔍 Search list

Franklin Sports Grow-with-Me Kids Base...

CAP Barbell A-Frame Dumbbell Weight ...

UGG Women's Classic Ultra Mini Boot

Crock-Pot 7 Quart Oval Manual Slow Co...

Contact Us

Let's Connect

First name

Email

Phone number

Description

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Send Message



- **Screenshots/Recordings:**

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

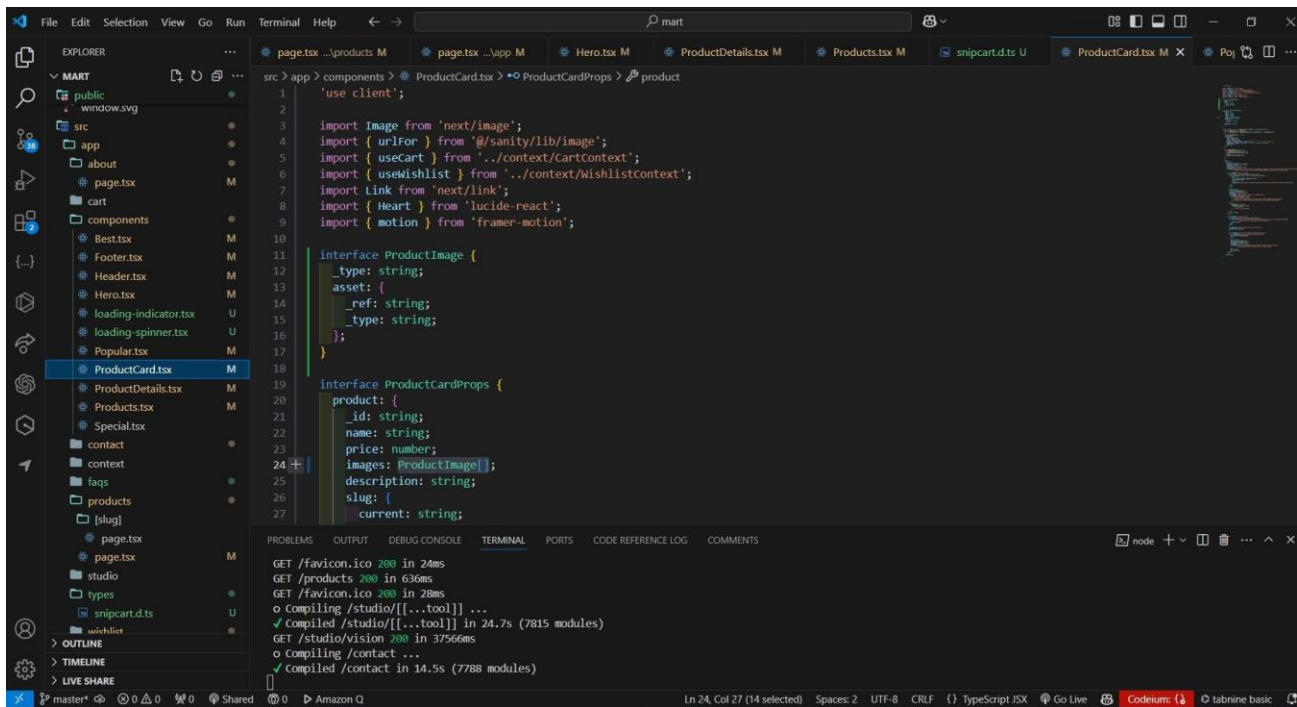
- **Description:**

- **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



The screenshot shows a VS Code editor with the following components:

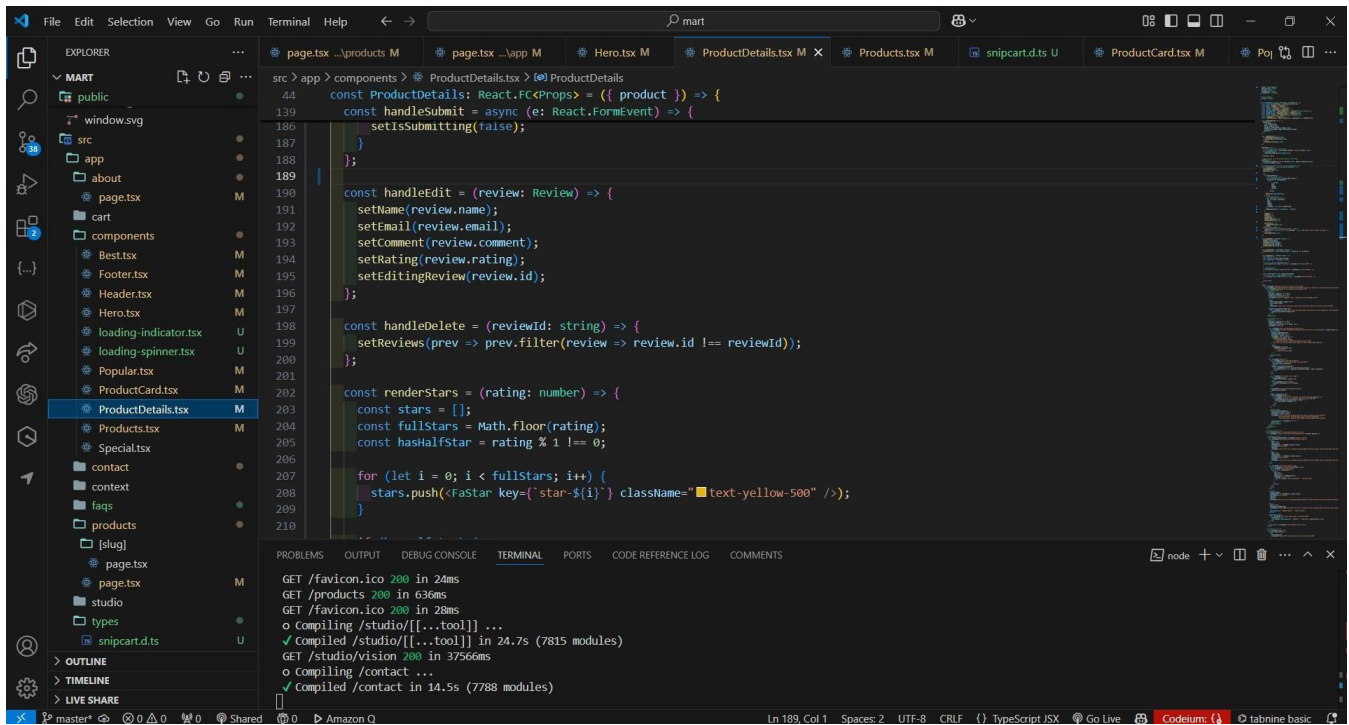
- EXPLORER:** A file tree on the left showing the project structure. The 'components' folder is expanded, and 'ProductCard.tsx' is selected.
- ProductCard.tsx:** The main file being edited, containing the following code:

```
1 'use client';
2
3 import Image from 'next/image';
4 import { urlFor } from '@sanity/lib/image';
5 import { useCart } from '../context/cartcontext';
6 import { useWishlist } from '../context/wishlistcontext';
7 import Link from 'next/link';
8 import { Heart } from 'lucide-react';
9 import { motion } from 'framer-motion';
10
11 interface ProductImage {
12   _type: string;
13   asset: {
14     _ref: string;
15     _type: string;
16   };
17 }
18
19 interface ProductCardProps {
20   product: {
21     _id: string;
22     name: string;
23     price: number;
24     images: ProductImage[];
25     description: string;
26     slug: {
27       current: string;
```

- Terminal:** A terminal window at the bottom showing the output of the build process, including the following logs:

```
GET /favicon.ico 200 in 24ms
GET /products 200 in 636ms
GET /favicon.ico 200 in 28ms
o compiling /studio/[...tool] ...
✓ compiled /studio/[...tool] in 24.7s (7815 modules)
GET /studio/vision 200 in 3756ms
o compiling /contact ...
✓ compiled /contact in 14.5s (7788 modules)
```


Review Component



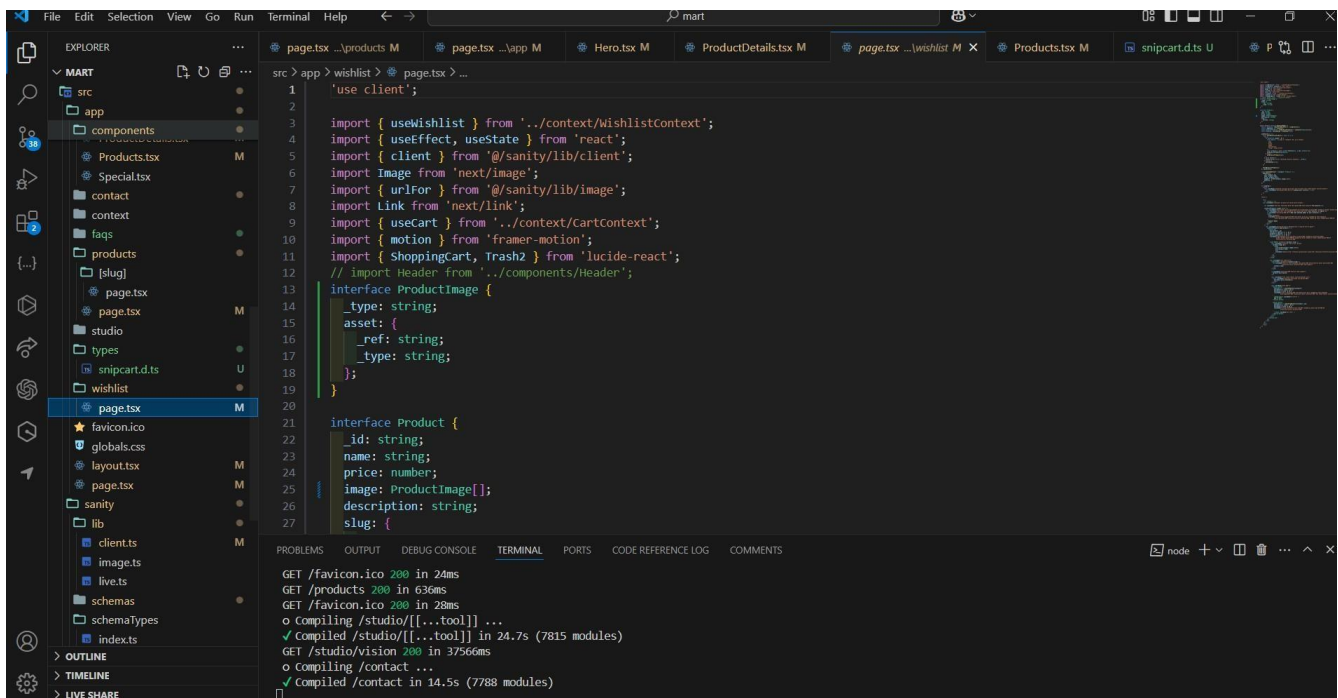
The screenshot shows the VS Code editor with the `ProductDetails.tsx` file open. The file contains the following code:

```
src > app > components > ProductDetails.tsx > ProductDetails
44  const ProductDetails: React.FC<Props> = ({ product }) => {
139  const handleSubmit = async (e: React.FormEvent) => {
186      setIsSubmitting(false);
187  }
188  }
189  }
190  const handleEdit = (review: Review) => {
191      setName(review.name);
192      setEmail(review.email);
193      setComment(review.comment);
194      setRating(review.rating);
195      setEditingReview(review.id);
196  };
197  }
198  const handleDelete = (reviewId: string) => {
199      setReviews(prev => prev.filter(review => review.id !== reviewId));
200  };
201  }
202  const renderStars = (rating: number) => {
203      const stars = [];
204      const fullStars = Math.floor(rating);
205      const hashHalfStar = rating % 1 !== 0;
206  }
207  for (let i = 0; i < fullStars; i++) {
208      stars.push(<FaStar key={`star-${i}`} className="text-yellow-500" />);
209  }
210  }
```

The terminal output shows the following commands and their results:

```
GET /favicon.ico 200 in 24ms
GET /products 200 in 636ms
GET /favicon.ico 200 in 28ms
o Compiling /studio/[...tool] ...
✓ compiled /studio/[...tool] in 24.7s (7815 modules)
GET /studio/vision 200 in 3756ms
o Compiling /contact ...
✓ compiled /contact in 14.5s (7788 modules)
```

Whish List Component



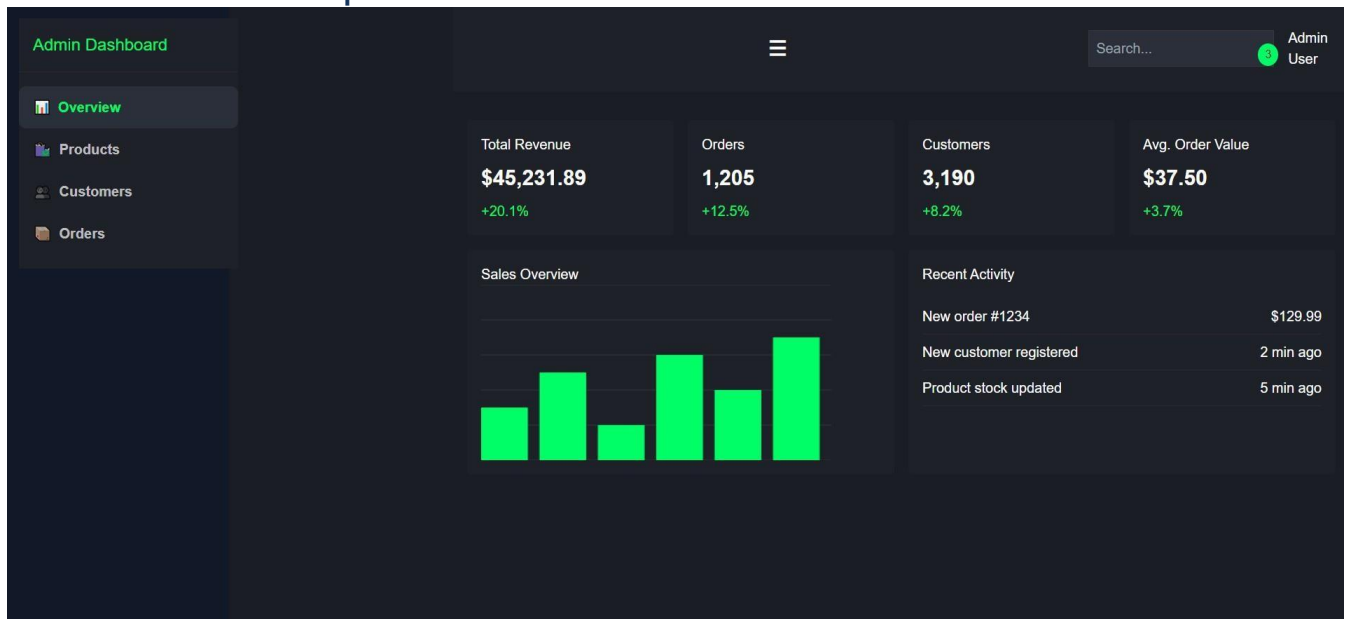
The screenshot shows the VS Code editor with the `page.tsx` file open. The file contains the following code:

```
src > app > wishlist > page.tsx > ...
1  'use client';
2
3  import { useWishlist } from '../context/WishlistContext';
4  import { useEffect, useState } from 'react';
5  import { client } from '@sanity/lib/client';
6  import Image from 'next/image';
7  import { urlFor } from '@sanity/lib/image';
8  import Link from 'next/link';
9  import { useCart } from '../context/CartContext';
10 import { motion } from 'framer-motion';
11 import { ShoppingCart, Trash2 } from 'lucide-react';
12 // import Header from '../components/Header';
13 interface ProductImage {
14   _type: string;
15   asset: {
16     _ref: string;
17     _type: string;
18   };
19 }
20
21 interface Product {
22   _id: string;
23   name: string;
24   price: number;
25   image: ProductImage[];
26   description: string;
27   slug: {
```

The terminal output shows the following commands and their results:

```
GET /favicon.ico 200 in 24ms
GET /products 200 in 636ms
GET /favicon.ico 200 in 28ms
o Compiling /studio/[...tool] ...
✓ compiled /studio/[...tool] in 24.7s (7815 modules)
GET /studio/vision 200 in 3756ms
o Compiling /contact ...
✓ compiled /contact in 14.5s (7788 modules)
```

Admin Dashboard Component



Login Page

A screenshot of a Login Page. The page has a dark background. In the center is a white rounded rectangle containing a shield icon, the text 'SECURE ACCESS TERMINAL', two input fields for 'Email' and 'Password', and a blue button labeled 'INITIATE LOGIN SEQUENCE'.

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:**

- 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
- Search Bar
- Image Gallery
- Add-to-Cart
- Wish List
- Pagination
- Header
- Footer
- Admin Dashboard
- 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