Day 4 - Building Dynamic Frontend Components for Car Rental Marketplace

On Day 4, I focused on designing and developing dynamic frontend components to display marketplace data fetched from Sanity CMS or APIs. The goal was to build modular and reusable components while ensuring responsiveness and scalability. Here's what I accomplished step by step:

1. Setup

- Connected the Next.js project to Sanity CMS and ensured data was fetching correctly.

- Tested API calls to verify the availability of marketplace data.

2. Implemented the Product Listing Component

- Created a dynamic grid layout to render product data.

- Displayed essential fields:

- Product Name

- Price

- Image

- Ensured responsiveness for different screen sizes.

3. Built the Product Detail Component

- Implemented dynamic routing in Next.js for individual product pages.

- Displayed detailed information:

- Product Description

- Price

- Available Sizes or Colors

4. Developed the Category Component

- Fetched categories dynamically from the database.

- Enabled filtering of products based on selected categories.

5. Added Search Functionality

- Implemented a search bar to filter products by name or tags.

6. Created the Wishlist Component

- Since our car rental marketplace doesn’t have an 'Add to Cart' feature, I built an 'Add to Wishlist' functionality.

- Used local storage or a global state management tool to persist wishlist data.

```
"use client";

import { useEffect, useState } from "react";
import Image from "next/image";
import Link from "next/link";

Codeium: Refactor | Explain | Generate JSDoc | ×
export default function Wishlist() {
  const [wishlist, setWishlist] = useState<any[]>([]);

  useEffect(() => {
    const storedWishlist = JSON.parse(localStorage.getItem("wishlist") || "[]");
    setWishlist(storedWishlist);
  }, []);

  Codeium: Refactor | Explain | Generate JSDoc | ×
  const handleRemoveFromWishlist = (item: any) => {
    const updatedWishlist = wishlist.filter((product) => product._id !== item._id);
    setWishlist(updatedWishlist);
    localStorage.setItem("wishlist", JSON.stringify(updatedWishlist));
  };

  return (
    <div className="container min-h-screen min-w-full bg-white">
      <h1 className="text-3xl font-bold mb-6 text-center text-black ">My Wishlist</h1>
      {wishlist.length === 0 ? (
        <p className="text-center text-gray-900">Your wishlist is empty.</p>
```

7. Implemented the Checkout Flow Component

- Designed a multi-step form for checkout, including:

- Billing and shipping address fields

- Payment details (mock implementation)

8. Built the User Profile Component

- Displayed user details like:

- Name, email, and saved addresses

- Order history with links to individual orders

9. Added the Reviews and Ratings Component

- Implemented a system for users to view and submit reviews.
- Displayed average ratings dynamically.

10. Developed Pagination for Large Data Sets

- Implemented pagination to break large product lists into manageable pages.
- Added previous and next buttons for better navigation.

11. Created an Advanced Filter Panel

- Added filtering options such as:
  - Price range sliders
  - Brand selection
  - Availability toggles (e.g., 'In Stock' only)

12. Built the Related Products Component

- Displayed similar or complementary products based on:
  - Tags
  - Categories
  - Customer behavior

13. Designed the Header and Footer Components

- Ensured consistent navigation and branding.
- Included links to key pages (Home, About, Contact, etc.).
- Focused on responsiveness and accessibility.

14. Implemented the Notifications Component

- Created real-time alerts for actions like:
  - Adding to wishlist

- Errors

- Successful form submissions

- Used toast notifications and modal windows.

Expected Output by the End of Day 4:

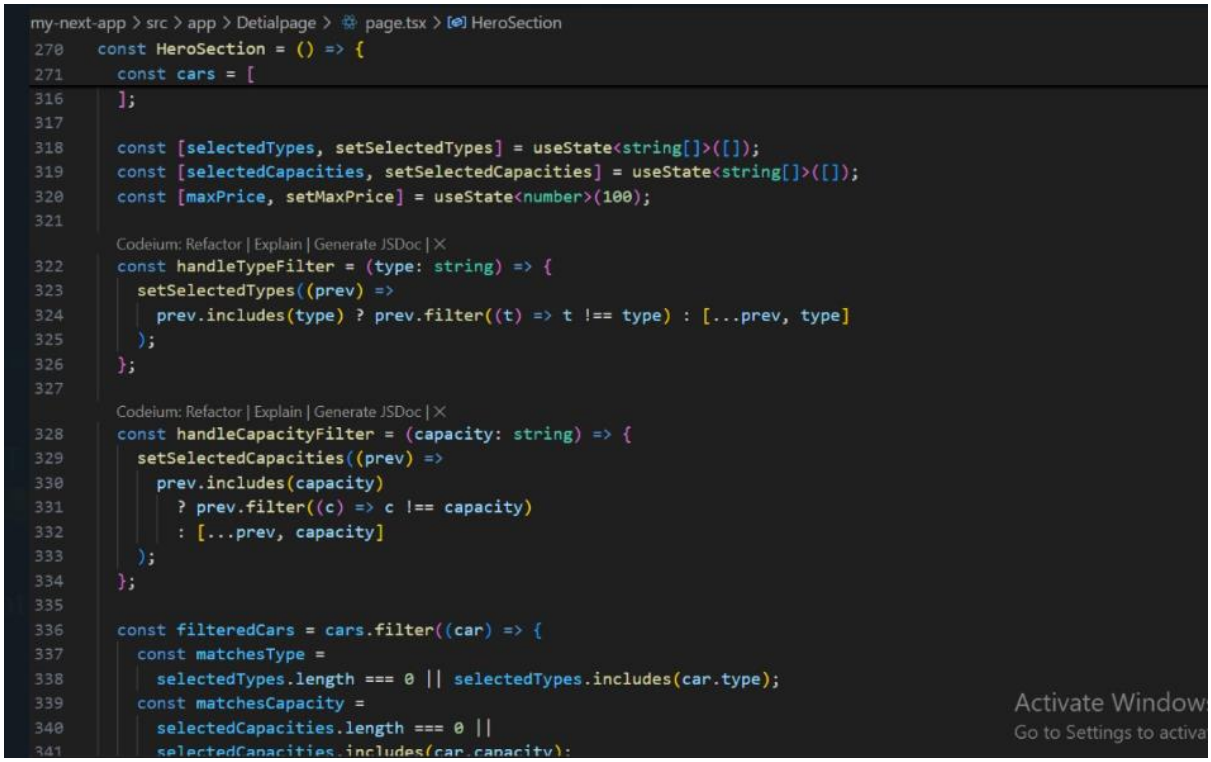
By the end of today, I successfully:

1. Built a functional product listing page displaying dynamic data from Sanity CMS or APIs.
2. Implemented individual product detail pages using dynamic routing.
3. Added advanced category filters for better product segmentation.
4. Developed a search bar to filter products efficiently.
5. Included pagination and related products for an improved user experience.
6. Styled all components for responsiveness and a professional look across devices.
7. Ensured that all components are modular and reusable for future scalability.

WISHLIST

```
"use client";

import { useEffect, useState } from "react";
import Image from "next/image";
import Link from "next/link";

Codeium: Refactor | Explain | Generate JSDoc | ×
export default function Wishlist() {
  const [wishlist, setWishlist] = useState<any[]>([]);

  useEffect(() => {
    const storedWishlist = JSON.parse(localStorage.getItem("wishlist") || "[]");
    setWishlist(storedWishlist);
  }, []);

  Codeium: Refactor | Explain | Generate JSDoc | ×
  const handleRemoveFromWishlist = (item: any) => {
    const updatedWishlist = wishlist.filter((product) => product._id !== item._id);
    setWishlist(updatedWishlist);
    localStorage.setItem("wishlist", JSON.stringify(updatedWishlist));
  };

  return (
    <div className="container min-h-screen min-w-full bg-white">
      <h1 className="text-3xl font-bold mb-6 text-center text-black ">My Wishlist</h1>
      {wishlist.length === 0 ? (
        <p className="text-center text-gray-900">Your wishlist is empty.</p>
```

HERO SECTION

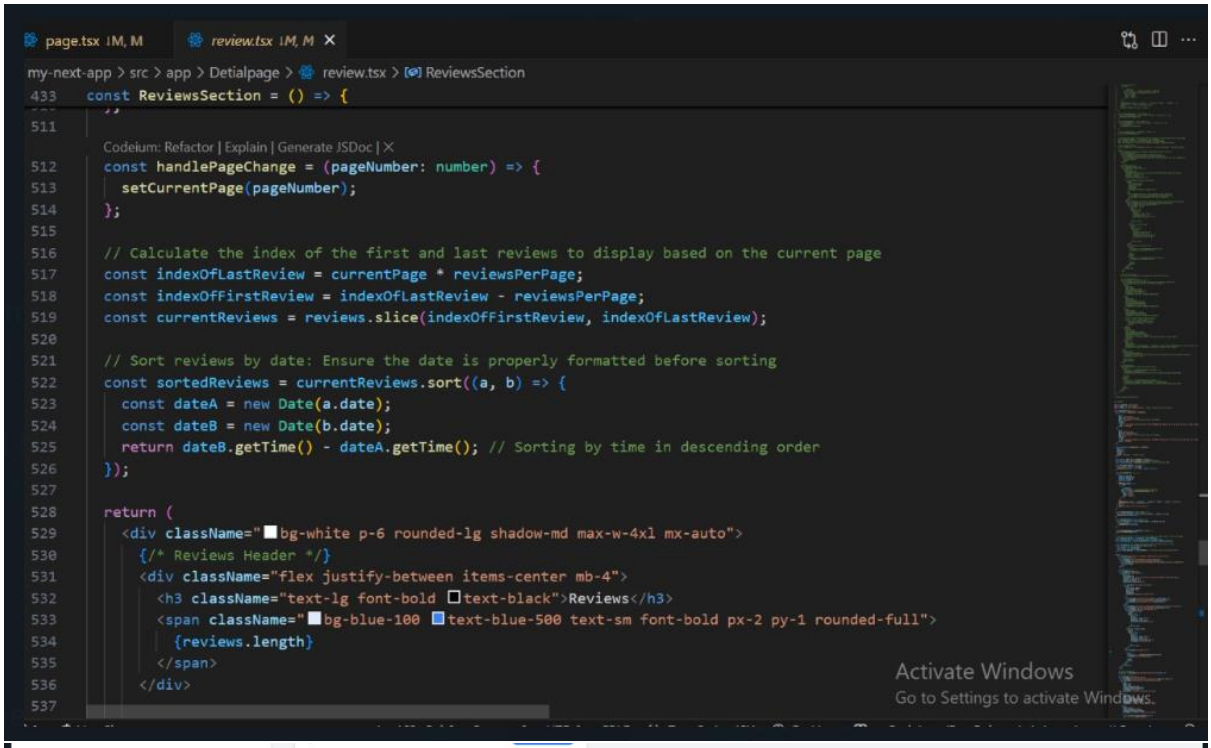


```
my-next-app > src > app > Detialpage > page.tsx > HeroSection
270  const HeroSection = () => {
271    const cars = [
316    ];
317
318    const [selectedTypes, setSelectedTypes] = useState<string[]>([]);
319    const [selectedCapacities, setSelectedCapacities] = useState<string[]>([]);
320    const [maxPrice, setMaxPrice] = useState<number>(100);
321
       Codeium: Refactor | Explain | Generate JSDoc | X
322    const handleTypeFilter = (type: string) => {
323      setSelectedTypes((prev) =>
324        prev.includes(type) ? prev.filter((t) => t !== type) : [...prev, type]
325      );
326    };
327
       Codeium: Refactor | Explain | Generate JSDoc | X
328    const handleCapacityFilter = (capacity: string) => {
329      setSelectedCapacities((prev) =>
330        prev.includes(capacity)
331          ? prev.filter((c) => c !== capacity)
332          : [...prev, capacity]
333      );
334    };
335
336    const filteredCars = cars.filter((car) => {
337      const matchesType =
338        selectedTypes.length === 0 || selectedTypes.includes(car.type);
339      const matchesCapacity =
340        selectedCapacities.length === 0 ||
341        selectedCapacities.includes(car.capacity);
```


PRODUCT DETAILS

```
my-next-app > src > app > carpost > [slug] > page.tsx > ...
       Codeium: Refactor | Explain | Generate JSDoc | X
26   export default function ProductDetail() {
27     const [product, setProduct] = useState<any>(null);
28     const [loading, setLoading] = useState(true);
29     const router = useRouter();
30     const params = useParams();
31
32     useEffect(() => {
         Codeium: Refactor | Explain | Generate JSDoc | X
33       async function fetchData() {
34         const slug = params?.slug;
35
36         if (!slug || Array.isArray(slug)) {
37           console.error("Invalid slug provided.");
38           return;
39         }
40
41         try {
42           const fetchedProduct = await getProduct(slug);
43           setProduct(fetchedProduct);
44         } catch (error) {
45           console.error("Error fetching product:", error);
46         } finally {
47           setLoading(false);
48         }
49       }
50
51     fetchData();
```

REVIEW SECTION

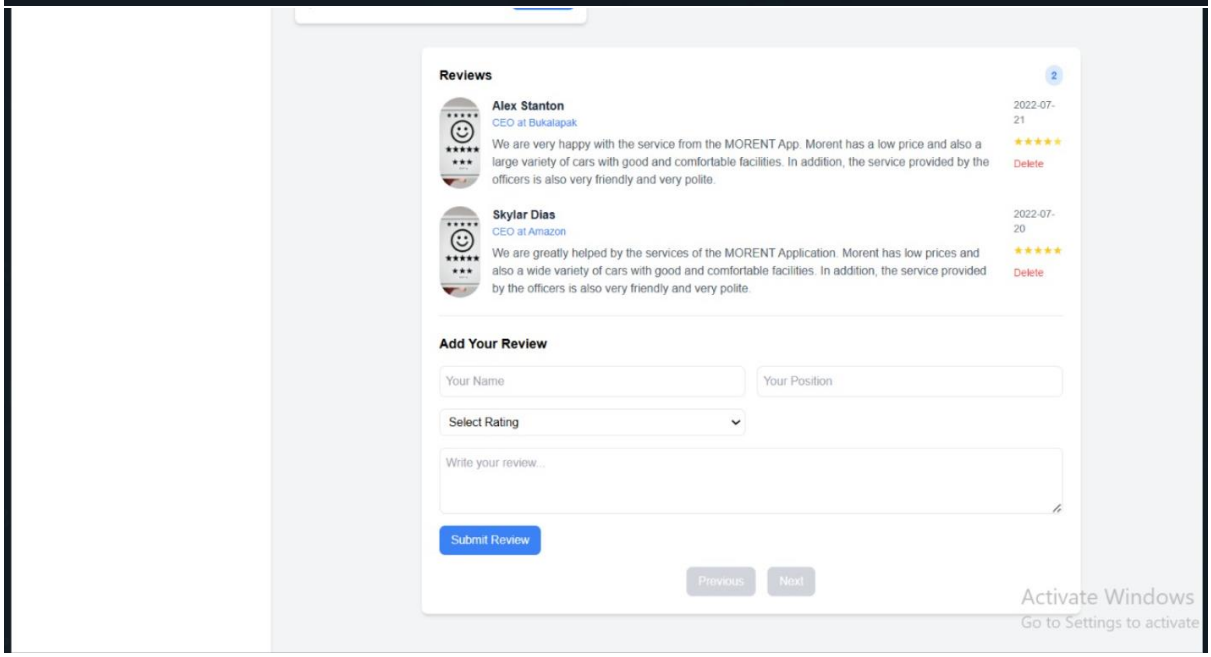

page.tsx 1M, M
review.tsx 1M, M
my-next-app > src > app > Detialpage > review.tsx > ReviewsSection
const ReviewsSection = () => {
Codeium: Refactor | Explain | Generate JSDoc | ×
const handlePageChange = (pageNumber: number) => {
setCurrentPage(pageNumber);
};
// Calculate the index of the first and last reviews to display based on the current page
const indexOfLastReview = currentPage * reviewsPerPage;
const indexOfFirstReview = indexOfLastReview - reviewsPerPage;
const currentReviews = reviews.slice(indexOfFirstReview, indexOfLastReview);
// Sort reviews by date: Ensure the date is properly formatted before sorting
const sortedReviews = currentReviews.sort((a, b) => {
const dateA = new Date(a.date);
const dateB = new Date(b.date);
return dateB.getTime() - dateA.getTime(); // Sorting by time in descending order
});
return (

{/* Reviews Header */}

<h3 className="text-lg font-bold text-black">Reviews</h3>

{reviews.length}

Activate Windows
Go to Settings to activate Windows.


Reviews
2
Alex Stanton
CEO at Bukalapak
2022-07-21
We are very happy with the service from the MORENT App. Morent has a low price and also a large variety of cars with good and comfortable facilities. In addition, the service provided by the officers is also very friendly and very polite.
Delete
Skylar Dias
CEO at Amazon
2022-07-20
We are greatly helped by the services of the MORENT Application. Morent has low prices and also a wide variety of cars with good and comfortable facilities. In addition, the service provided by the officers is also very friendly and very polite.
Delete
Add Your Review
Your Name
Your Position
Select Rating
Write your review...
Submit Review
Previous
Next
Activate Windows
Go to Settings to activate