Day 4 - Dynamic Frontend Components - [AliCarGo]

Author:
Ali Shahid

Roll no:
00230027

Day 4 of marketplace builder hackathon

Technical Report 👨‍💻

Steps Taken to Build and Integrate Components

1. **Setting Up the Project Structure:**
   - Organized the project structure to ensure modularity and maintainability.
   - Created individual components such as Navbar, ProductCard, ProductList, ProductDetail and SearchBar.
2. **Building the Navbar with Search Functionality:**
   - Designed a responsive navbar that includes a logo, search bar, and profile icons.
   - Implemented the search functionality to filter products dynamically by name or tags.
3. **Product Listing Page:**
   - Fetched product data dynamically from the Sanity CMS using a custom query (allproducts).
   - Rendered a grid layout to display all products, each represented by a ProductCard component.
   - Added a "Show More" button to allow users to load more products.
4. **Individual Product Detail Pages:**
   - Implemented dynamic routing in Next.js to generate individual product detail pages using product IDs.
   - Ensured accurate rendering of product data, including images, specifications, and price.
5. **Category Filters and Search Bar:**
   - Created a search bar component that filters products by name or tags in real time.
   - Integrated category filtering logic to allow users to refine their searches.
6. **Pagination Implementation:**
   - Configured pagination to divide product data into manageable chunks for better user experience.
7. **Additional Features:**
   - Added related products section on individual product pages to enhance discoverability.

Challenges Faced and Solutions Implemented 💻

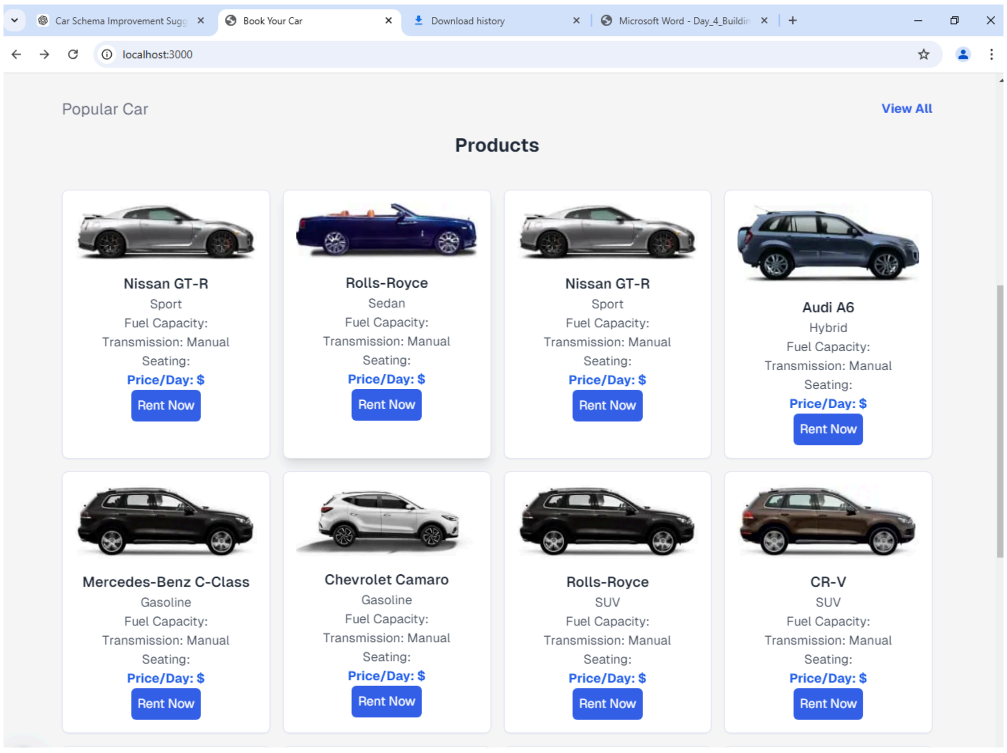
1. **Challenge: Handling large amounts of product data without performance degradation.**
   - Solution: Implemented pagination and lazy loading to manage data efficiently and improve performance.
2. **Challenge: Ensuring accurate routing and data fetching for individual product pages.**
   - Solution: Used Next.js dynamic routes (/products/[id]) and server-side data fetching to render the appropriate product details.
3. **Challenge: Implementing real-time search functionality.**
   - Solution: Utilized a combination of useState and useEffect to filter products dynamically as the user typed in the search bar.
4. **Challenge: Managing responsiveness across devices.**
   - Solution: Used Tailwind CSS utility classes to build a fully responsive UI that adapts to different screen sizes.

Best Practices Followed During Development 🧑‍💻

1. Component Reusability:
   - Designed modular components (e.g., ProductCard, Navbar, SearchBar) that can be reused across multiple pages and contexts.
2. Code Maintainability:
   - Followed clean coding principles with consistent naming conventions, clear comments, and modular file structures.
3. Performance Optimization:
   - Leveraged lazy loading, pagination, and efficient API calls to enhance performance.
4. Responsive Design:
   - Ensured a seamless user experience across all devices using Tailwind CSS.
5. Error Handling:
   - Implemented robust error handling for API calls and dynamic routing to prevent application crashes.
6. Dynamic Data Integration:
   - Used a headless CMS (Sanity) to dynamically fetch and render product data, ensuring flexibility and scalability.

Screenshots and coding snippets 🖥️

1: Product listing using dynamic data fetching

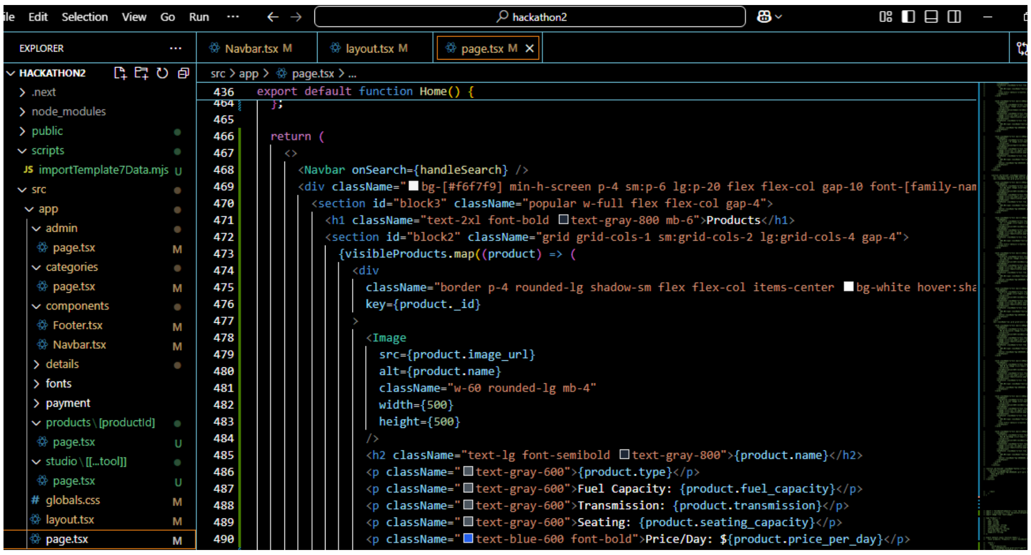


Popular Car View All

Products

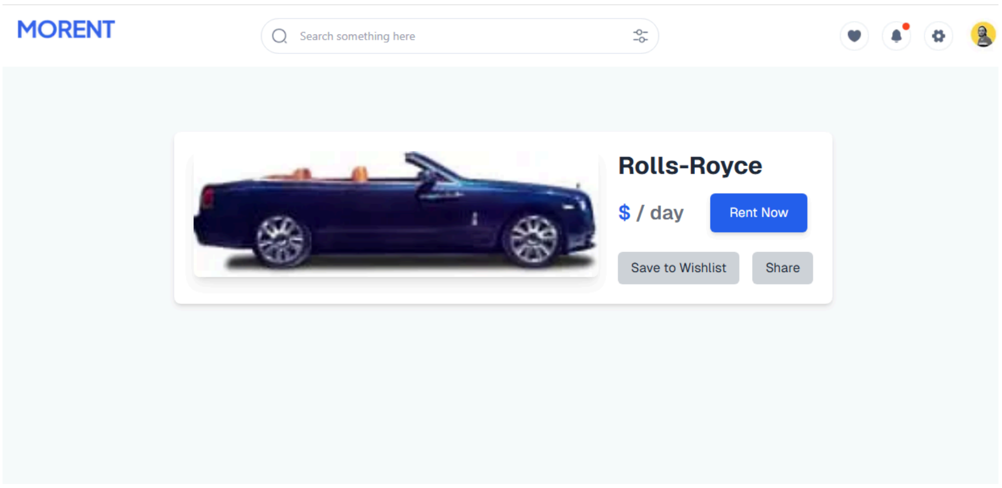
Nissan GT-R	Rolls-Royce	Nissan GT-R	Audi A6
Sport	Sedan	Sport	Hybrid
Fuel Capacity:	Fuel Capacity:	Fuel Capacity:	Fuel Capacity:
Transmission: Manual	Transmission: Manual	Transmission: Manual	Transmission: Manual
Seating:	Seating:	Seating:	Seating:
Price/Day: $	Price/Day: $	Price/Day: $	Price/Day: $
Rent Now	Rent Now	Rent Now	Rent Now
Mercedes-Benz C-Class	Chevrolet Camaro	Rolls-Royce	CR-V
Gasoline	Gasoline	SUV	SUV
Fuel Capacity:	Fuel Capacity:	Fuel Capacity:	Fuel Capacity:
Transmission: Manual	Transmission: Manual	Transmission: Manual	Transmission: Manual
Seating:	Seating:	Seating:	Seating:
Price/Day: $	Price/Day: $	Price/Day: $	Price/Day: $
Rent Now	Rent Now	Rent Now	Rent Now

Code snippet

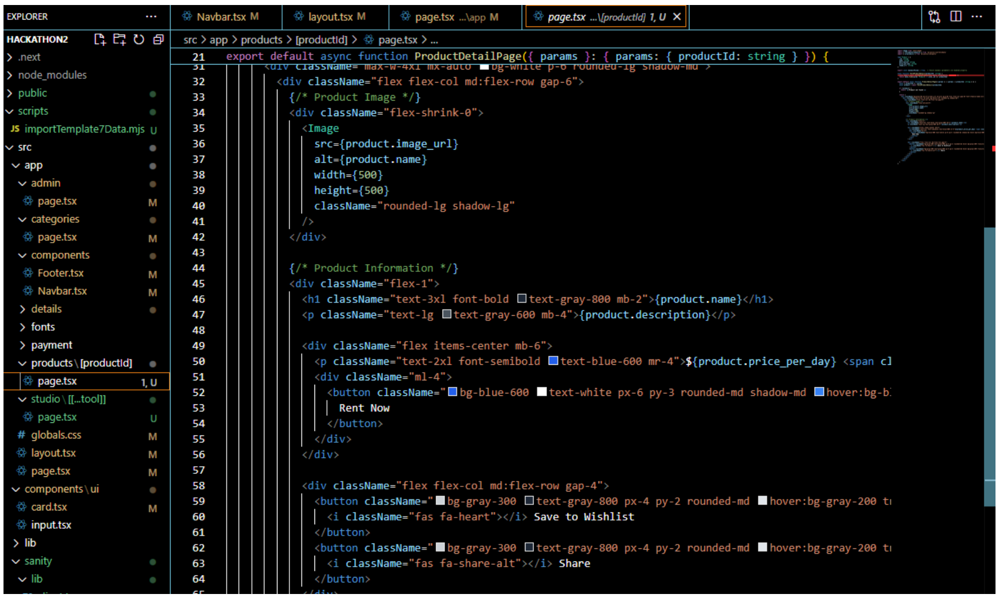


```
export default function Home() {
  };

  return (
    <>
      <Navbar onSearch={handleSearch} />
      <div className="bg-[#f6f7f9] min-h-screen p-4 sm:p-6 lg:p-20 flex flex-col gap-10 font-[family-nam
        <section id="block3" className="popular w-full flex flex-col gap-4">
          <h1 className="text-2xl font-bold text-gray-800 mb-6">Products</h1>
          <section id="block2" className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-4">
            {visibleProducts.map((product) => (
              <div
                className="border p-4 rounded-lg shadow-sm flex flex-col items-center bg-white hover:sha
                key={product._id}
              >
                <Image
                  src={product.image_url}
                  alt={product.name}
                  className="w-60 rounded-lg mb-4"
                  width={500}
                  height={500}
                />
                <h2 className="text-lg font-semibold text-gray-800">{product.name}</h2>
                <p className="text-gray-600">{product.type}</p>
                <p className="text-gray-600">Fuel Capacity: {product.fuel_capacity}</p>
                <p className="text-gray-600">Transmission: {product.transmission}</p>
                <p className="text-gray-600">Seating: {product.seating_capacity}</p>
                <p className="text-blue-600 font-bold">Price/Day: ${product.price_per_day}</p>
```

2. Product Detail Page

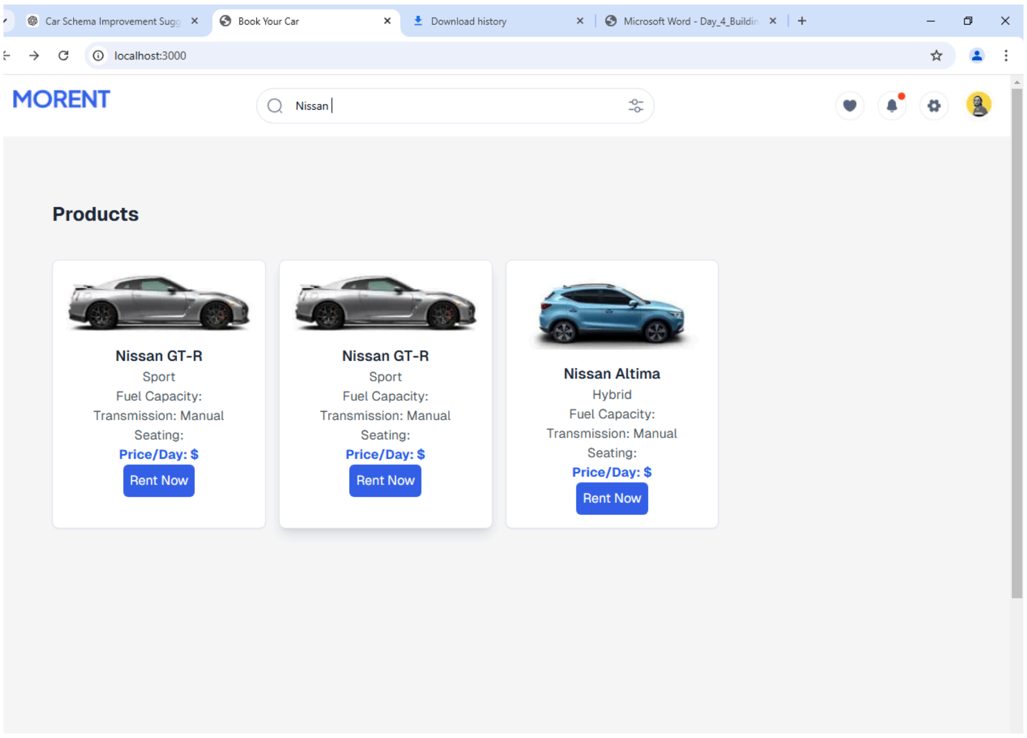


Code snippet

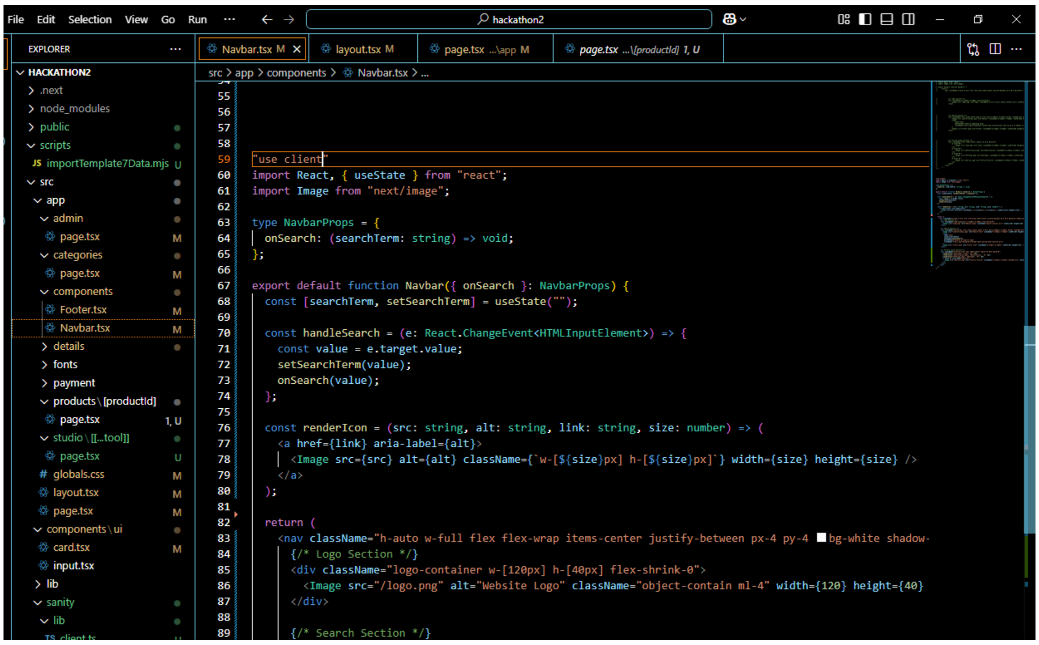


```
export default async function ProductDetailPage({ params }: { params: { productId: string } }) {
      <div className="max-w-4xl mx-auto bg-white p-6 rounded-lg shadow-md">
        <div className="flex flex-col md:flex-row gap-6">
          {/* Product Image */}
          <div className="flex-shrink-0">
            <Image
              src={product.image_url}
              alt={product.name}
              width={500}
              height={500}
              className="rounded-lg shadow-lg"
            />
          </div>

          {/* Product Information */}
          <div className="flex-1">
            <h1 className="text-3xl font-bold text-gray-800 mb-2">{product.name}</h1>
            <p className="text-lg text-gray-600 mb-4">{product.description}</p>

            <div className="flex items-center mb-6">
              <p className="text-2xl font-semibold text-blue-600 mr-4">${product.price_per_day} <span cl
              <div className="ml-4">
                <button className="bg-blue-600 text-white px-6 py-3 rounded-md shadow-md hover:bg-b
                  Rent Now
                </button>
              </div>
            </div>

            <div className="flex flex-col md:flex-row gap-4">
              <button className="bg-gray-300 text-gray-800 px-4 py-2 rounded-md hover:bg-gray-200 t
                <i className="fas fa-heart"></i> Save to Wishlist
              </button>
              <button className="bg-gray-300 text-gray-800 px-4 py-2 rounded-md hover:bg-gray-200 t
                <i className="fas fa-share-alt"></i> Share
              </button>
```

3. Search by Names or Tags



Code snippet



```
"use client"
import React, { useState } from "react";
import Image from "next/image";

type NavbarProps = {
  onSearch: (searchTerm: string) => void;
};

export default function Navbar({ onSearch }: NavbarProps) {
  const [searchTerm, setSearchTerm] = useState("");

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

  const renderIcon = (src: string, alt: string, link: string, size: number) => (
    <a href={link} aria-label={alt}>
      <Image src={src} alt={alt} className={`w-[${size}px] h-[${size}px]`} width={size} height={size} />
    </a>
  );

  return (
    <nav className="h-auto w-full flex flex-wrap items-center justify-between px-4 py-4 bg-white shadow-
      {/* Logo Section */}
      <div className="logo-container w-[120px] h-[40px] flex-shrink-0">
        <Image src="/logo.png" alt="Website Logo" className="object-contain ml-4" width={120} height={40}
      </div>

      {/* Search Section */}
```

Conclusion 🔍

The implementation of dynamic frontend components successfully enhances the user experience by enabling real-time data fetching, seamless navigation to product detail pages, and efficient search functionality. These features provide a solid foundation for a modern, user-friendly marketplace. By overcoming development challenges and adhering to best practices, this milestone demonstrates the scalability and robustness of the platform, ensuring it meets user expectations and market standards.