Implementing a dynamic Product Listing component:

Task 4:

Objective: Building Dynamic Frontend Components

The primary objective of building dynamic frontend components for an e-commerce marketplace is to create reusable, interactive, and responsive UI elements that adapt based on user interactions and backend data. These components aim to enhance the user experience, improve performance, and streamline development.
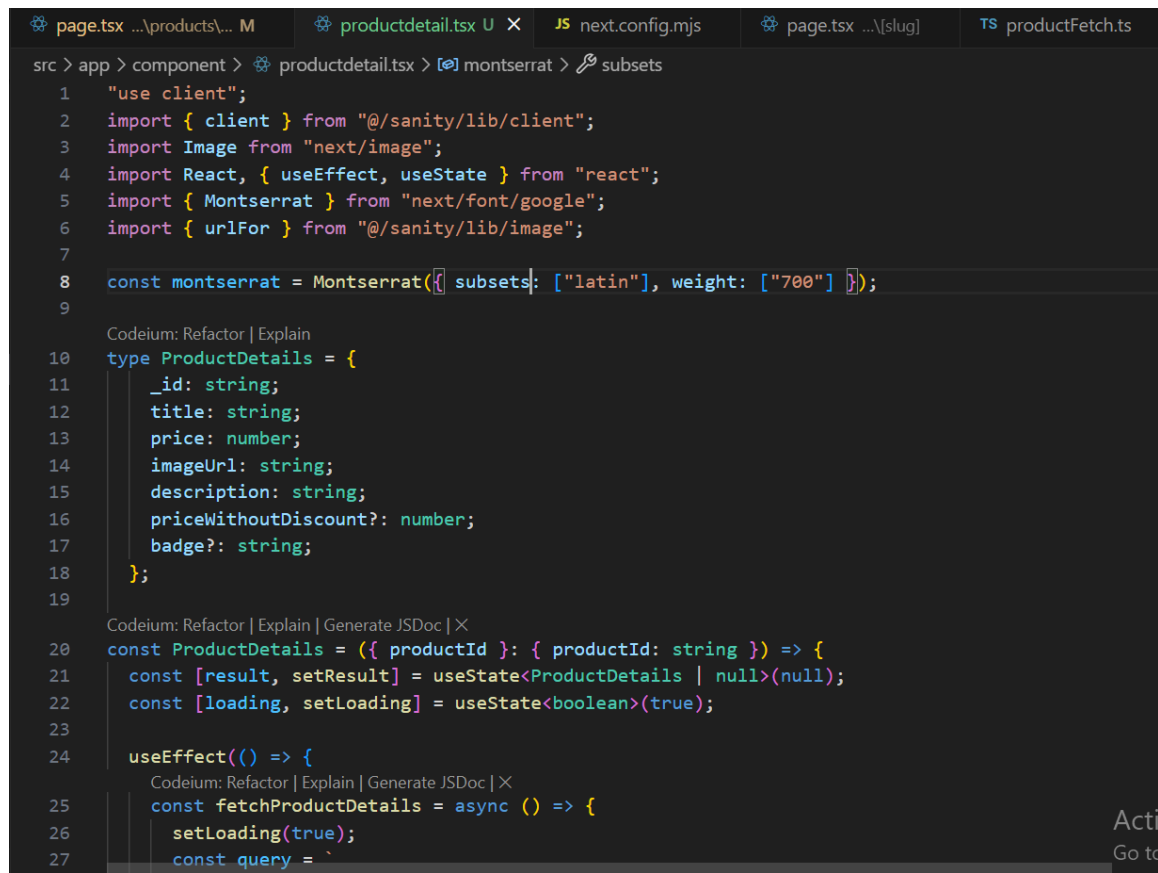
Key Objectives:

Reusability

Build components that can be reused across different parts of the application by leveraging props, state, and context. For example:

A product card component that dynamically adjusts content based on the chair's attributes (e.g., name, image, price).

Product detail:

```
page.tsx ...\products\... M        productdetail.tsx U  ×      JS next.config.mjs        page.tsx ...\[slug]      TS productFetch.ts

src > app > component > ⚙ productdetail.tsx > [@] montserrat > 🔧 subsets
  1    "use client";
  2    import { client } from "@/sanity/lib/client";
  3    import Image from "next/image";
  4    import React, { useEffect, useState } from "react";
  5    import { Montserrat } from "next/font/google";
  6    import { urlFor } from "@/sanity/lib/image";
  7
  8    const montserrat = Montserrat({ subsets: ["latin"], weight: ["700"] });
  9
       Codeium: Refactor | Explain
 10    type ProductDetails = {
 11        _id: string;
 12        title: string;
 13        price: number;
 14        imageUrl: string;
 15        description: string;
 16        priceWithoutDiscount?: number;
 17        badge?: string;
 18    };
 19
       Codeium: Refactor | Explain | Generate JSDoc | ✕
 20    const ProductDetails = ({ productId }: { productId: string }) => {
 21      const [result, setResult] = useState<ProductDetails | null>(null);
 22      const [loading, setLoading] = useState<boolean>(true);
 23
 24      useEffect(() => {
           Codeium: Refactor | Explain | Generate JSDoc | ✕
 25        const fetchProductDetails = async () => {                        Acti
 26          setLoading(true);                                             Go to
 27          const query = `
```

Tools and technologies:

- Framework:React or next js

- CMS:sanity CMS

- state managemente:using React Hooks

IMplementation Plan:

1:set up data fetching:

.integrate sanity CMS or api endpoint to fetch the product data dynamically.

.use React hooks (useEffect) use

2:Design Resuable Components:

.cart component:displays indiisual product detail add or remove

3:Responsive design:Apply Tailwand css to ensure the layout adaps to all screen

sizes like small or medium or large devices on the screen.

```tsx
const ProductDetails = ({ productId }: { productId: string }) => {
  const [result, setResult] = useState<ProductDetails | null>(null);
  const [loading, setLoading] = useState<boolean>(true);

  useEffect(() => {
    // Codeium: Refactor | Explain | Generate JSDoc | X
    const fetchProductDetails = async () => {
      setLoading(true);
      const query = `
      *[_type == "products"] {
        _id,
        title,
        price,
        "imageUrl": image.asset->url,
        description,
        priceWithoutDiscount,
        badge
      }
      `

      const data = await client.fetch(query );

      const product = data.find((item: ProductDetails) => item._id === productId);
      setResult(product || null);

      setLoading(false);
    };

    fetchProductDetails();
  }, [productId]);
```

S Default   + Create   Q   Structure   Vision   Schedules

Content   Products   +   ...

📁 Products   >   Q Search list

📁 Categories   >   Ivory Charm

                                    Gray Elegance

                                    Nordic Spin

                                    Scandi Dip Set

                                    Modern Cozy

                                    Library Stool Chair

                                    SleekSpin
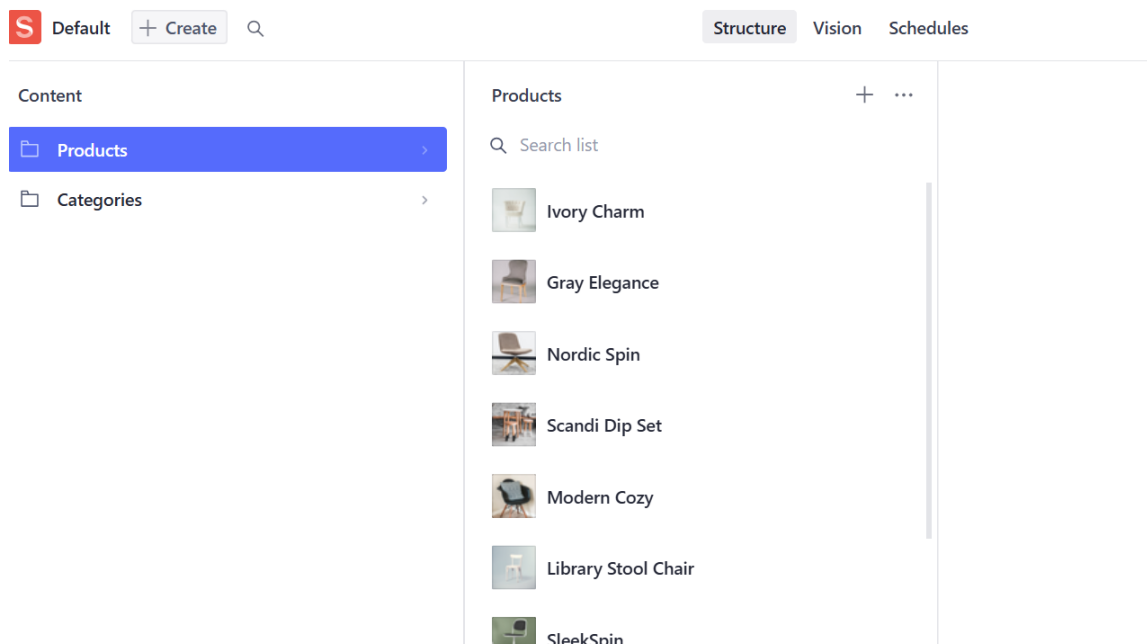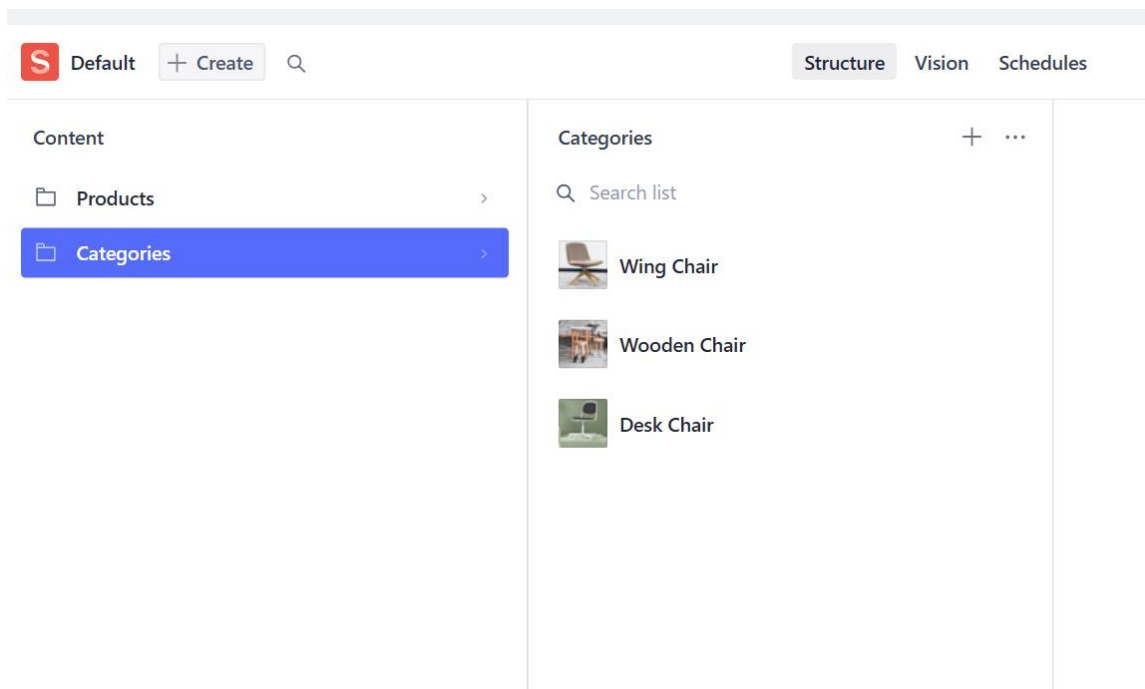
Cart Page:

All product dynamic include card page.

```
"use client";
import React, { useState } from "react";

Codeium: Refactor | Explain | Generate JSDoc | X
const ShoppingCart = () => {
  const [cartItems, setCartItems] = useState([
    { id: 1, name: "Library Stool Chair", price: 120, size: "L", quantity: 1, image: "/Frame.webp" },
    { id: 2, name: "Wooden Chair", price: 150, size: "L", quantity: 1, image: "/Image2.webp" },
  ]);
  const [checkoutSuccess, setCheckoutSuccess] = useState(false); // State to track checkout success

  const sizes = ["S", "M", "L", "XL"];

  Codeium: Refactor | Explain | Generate JSDoc | X
  const updateQuantity = (id: number, action: "increase" | "decrease") => {
    setCartItems((items) =>
      items.map((item) =>
        item.id === id
          ? { ...item, quantity: Math.max(1, item.quantity + (action === "increase" ? 1 : -1)) }
          : item
      )
    );
  };

  Codeium: Refactor | Explain | Generate JSDoc | X
  const updateSize = (id: number, size: string) => {
    setCartItems((items) => items.map((item) => (item.id === id ? { ...item, size } : item)));
  };
```

Shopping Cart

	Library Stool Chair Size: L	- 1 +	$120 Remove

Order Summary

Subtotal	$270.00
Discount	-$20.00
Tax (8%)	$21.60
Delivery	Free

Total	**$271.60**

Checkout

	Wooden Chair Size: L	- 1 +	$150 Remove

Shopping Cart

Checkout successful! Your order has been placed.

Order Summary

Subtotal	$0.00
Discount	-$20.00
Tax (8%)	$0.00
Delivery	Free

Total	**$-20.00**

Checkout

2:Product Detail component:

objective:

Develop a indivisual product detail pages using dynamic routing in Next js. This pages will display the product detail information of each product ,including:

.Name

.Product Description

.price

.category

.price with discount

.badge

Implementation plan:

1:Dynamic Routing:

.creating dynamic routes using the product/[id],page.tsx file in the app folder directory.

.fetch product data based on the product ID from CMS like sanity or an Api.

2:Data Fields:

Each product detail page should include the following fields:

.**Product description:**A detailed explanation of the product,fetch from backend sanity data CMS.

.**Price:**Developed promiently for clear visiblity.

3:Integration with Product   listning:

.Link each product cart in the product detail listning component to its corressponding detail page using the Link component in Next js.

4:styling:

.use tailwand css or css for a clean code and responsive design to see the ui/ux interface.

```
import ProductDetails from "@/app/component/productdetail";

Codeium: Refactor | Explain
interface Iparams {
  id: string;
}

Codeium: Refactor | Explain | Generate JSDoc | X
export default async function ProductDetailsPage ({ params }: { params: Promise<Iparams> }) {
  const { id } = await params;
  return (
    <ProductDetails productId={id} />      Ctrl+I Codeium Command
  );
};
```

Additional Components for Chair E-commerce

Here's a brief explanation of essential additional components for a dynamic chair
e-commerce platform:

Additional Components for Chair E-commerce

Here's a brief explanation of essential additional components for a dynamic chair
e-commerce platform:

1. Shopping Cart Component

The shopping cart is the hub for managing selected products before checkout.

```
"use client";
import React, { useState } from "react";
import { useRouter } from "next/navigation";

Codeium: Refactor | Explain | Generate JSDoc | ×
const ShoppingCart = () => {
  const [cartItems, setCartItems] = useState([
    { id: 1, name: "Cirtus Edge", price: 20, image: "/Frame.webp", quantity: 1 },
    { id: 2, name: "Rose Luxre Armchair", price: 20, image: "/Pinksofa.webp", quantity: 1 },
    { id: 3, name: "Library Stool Chair", price: 20, image: "/Image.webp", quantity: 1 },
  ]);

  const router = useRouter();

Codeium: Refactor | Explain | Generate JSDoc | ×
  const handleRemove = (id: number) => {
    setCartItems(cartItems.filter((item) => item.id !== id));
  };

Codeium: Refactor | Explain | Generate JSDoc | ×
  const handleQuantityChange = (id: number, action: "increase" | "decrease") => {
    setCartItems(
      cartItems.map((item) =>
        item.id === id
          ? { ...item, quantity: Math.max(1, item.quantity + (action === "increase" ? 1 : -1)) }
          : item
      )
    );
  };
};
```

Activate Windows

Your Shopping Cart

	Cirtus Edge Remove	- 1 +	$20
	Rose Luxre Armchair Remove	- 1 +	$20
	Library Stool Chair Remove	- 1 +	$20
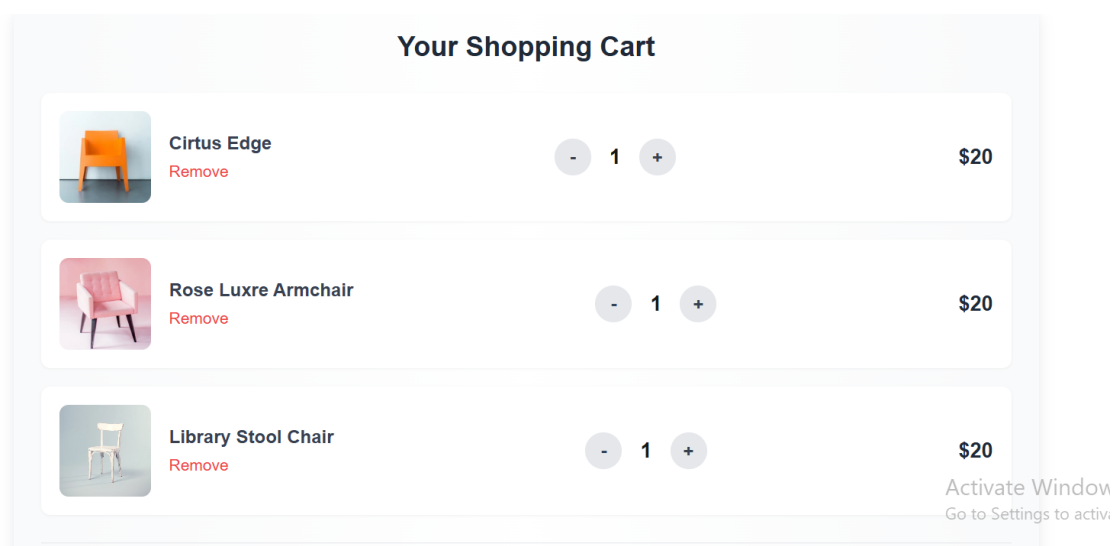
Features:

Product Overview: Displays the list of chairs added to the cart with quantity, price, and an image.

Quantity Adjustment: Allows users to increase or decrease the quantity of items.

Dynamic Pricing: Updates the total cost in real-time as items are added, removed, or quantities are changed.

Save for Later: Option to save items for future purchases.

2. Checkout Component

The checkout component processes orders and gathers user details for fulfillment.

```jsx
import React from 'react';
import { FaCcMastercard, FaCcVisa, FaPaypal } from 'react-icons/fa'; // Importing icons

Codeium: Refactor | Explain | Generate JSDoc | X
const Checkout = () => {
  return (
    <div className="font-sans bg-gradient-to-r from-blue-50 via-white to-blue-50 p-8">
      <div className="max-w-4xl mx-auto">
        <div className="text-center">
          <h2 className="text-4xl font-extrabold text-gray-800 inline-block border-b-[4px] border-blue-600 pb-2 t
            Checkout
          </h2>
        </div>

        <div className="mt--12">
          {/* Personal Details Section */}
          <div className="grid md:grid-cols-3 gap-8">
            <div>
              <h3 className="text-4xl font-bold text-gray-600">01</h3>
              <h3 className="text-2xl font-semibold text-gray-800 mt-1">Personal Details</h3>
            </div>

            <div className="md:col-span-2">
              <form>
                <div className="grid sm:grid-cols-2 gap-6">
                  <div>
                    <input
                      type="text"
                      placeholder="First name"
                      className="px-6 py-4 bg-white text-gray-800 w-full text-lg border-2 rounded-lg focus:bord
```

Checkout

01
Personal Details

First name

Last name

Email address

Phone number

02
Shopping Address

Street address

City

State

Zip Code

Activate Wind
Go to Settings to a

03

Features:

User Details: Collects shipping and billing information.

Order Summary: Displays a breakdown of selected chairs, taxes, and shipping charges.

Payment Gateway: Integrates with services like Stripe or PayPal for secure transactions.

Promo Code Application: Supports discount codes to reduce the total cost.

Guest Checkout: Allows users to checkout without creating an account.

3. Login/Authentication Component

The login component manages user accounts for a personalized shopping experience.

```jsx
import React from 'react';
import Link from 'next/link';
import { FaFacebook, FaGithub } from 'react-icons/fa';

Codeium: Refactor | Explain | Generate JSDoc | ✕
const Login = () => {
  return (
    <div className="min-h-screen flex items-center justify-center bg-gradient-to-r ■from-blue-50 ■via-purple-50
      <div className="max-w-4xl w-full ■bg-white p-6 rounded-lg shadow-2xl flex">
        {/* Left Side (Form) */}
        <div className="w-full sm:w-1/2 px-6">
          <h2 className="text-3xl font-extrabold □text-gray-800 text-center">
            Welcome Back!
          </h2>
          <p className="text-sm ■text-gray-600 text-center mt-2">
            Sign in to access your account
          </p>
          <form className="mt-6 space-y-4">
            <input
              type="email"
              name="email"
              className="w-full px-4 py-3 border ■border-gray-300 rounded-lg focus:outline-none focus:ring-2 ■fc
              placeholder="Email address"
              required
            />
            <input
              type="password"
              name="password"
              className="w-full px-4 py-3 border ■border-gray-300 rounded-lg focus:outline-none focus:ring-2 ■fc
              placeholder="Password"
```
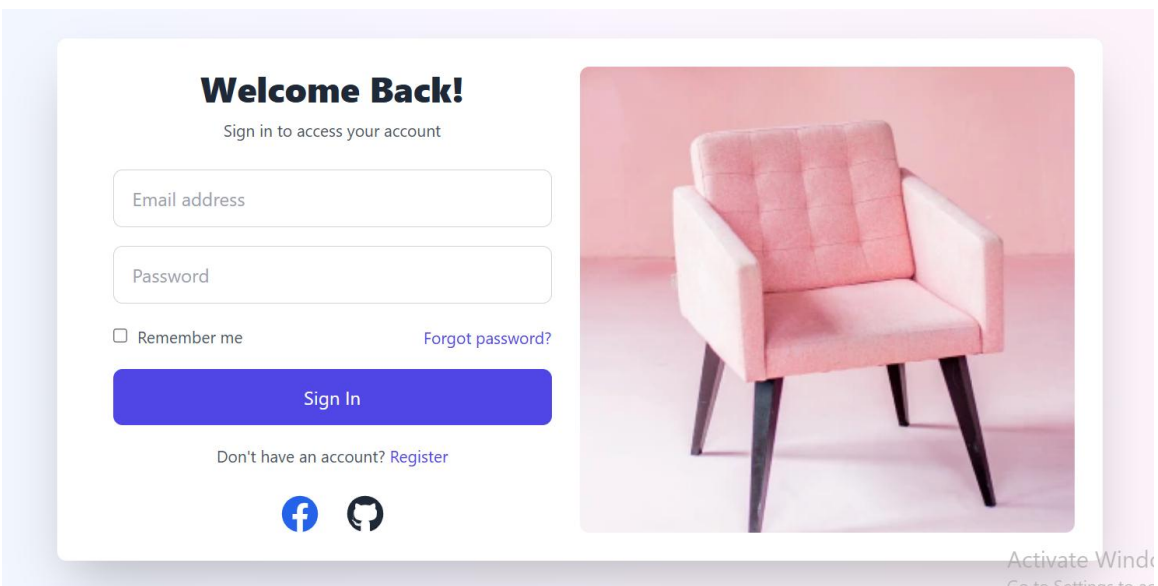
Activate Windows
Go to Settings to activate

Welcome Back!

Sign in to access your account

Email address

Password

☐ Remember me   Forgot password?

Sign In

Don't have an account? Register

Activate Windo
Go to Settings to a

Features:

Login/Signup: Enables users to log in or create accounts using email, phone, or social logins (e.g., Google, Facebook).

Persistent Login: Keeps users logged in using tokens or cookies.

Password Recovery: Allows users to reset forgotten passwords.

Account Management: Provides options for users to update their profile, view past orders, and manage saved addresses.

Integration Between Components

Shopping Cart to Checkout: Pass cart data to the checkout component seamlessly.

Login and Checkout: If a user is logged in, prefill their details during checkout for faster processing.

Dynamic Updates: Ensure cart and checkout updates reflect across the platform in real-time.

Conclusion:

In conclusion, building dynamic frontend components for a chair e-commerce marketplace involves creating reusable, interactive, and responsive UI elements that enhance user experience and streamline development. Leveraging modern tools like React or Next.js, Sanity CMS, and React Hooks ensures efficient data fetching, state management, and dynamic routing for seamless user interactions. Core components such as the shopping cart, checkout, and product detail pages are designed to offer features like real-time updates, secure transactions, and personalized user experiences. Tailwind CSS ensures responsive design across devices, while integrating these components ensures smooth data flow and a cohesive user journey. This approach not only meets functional and aesthetic requirements but also provides scalability, performance optimization, and a strong foundation for future enhancements.

-