JANUARY 2025

Furniro

Day 4 - Dynamic Frontend Components - Furniro

PRESENTED BY:

Ayesha Abdul Qadir

Objective

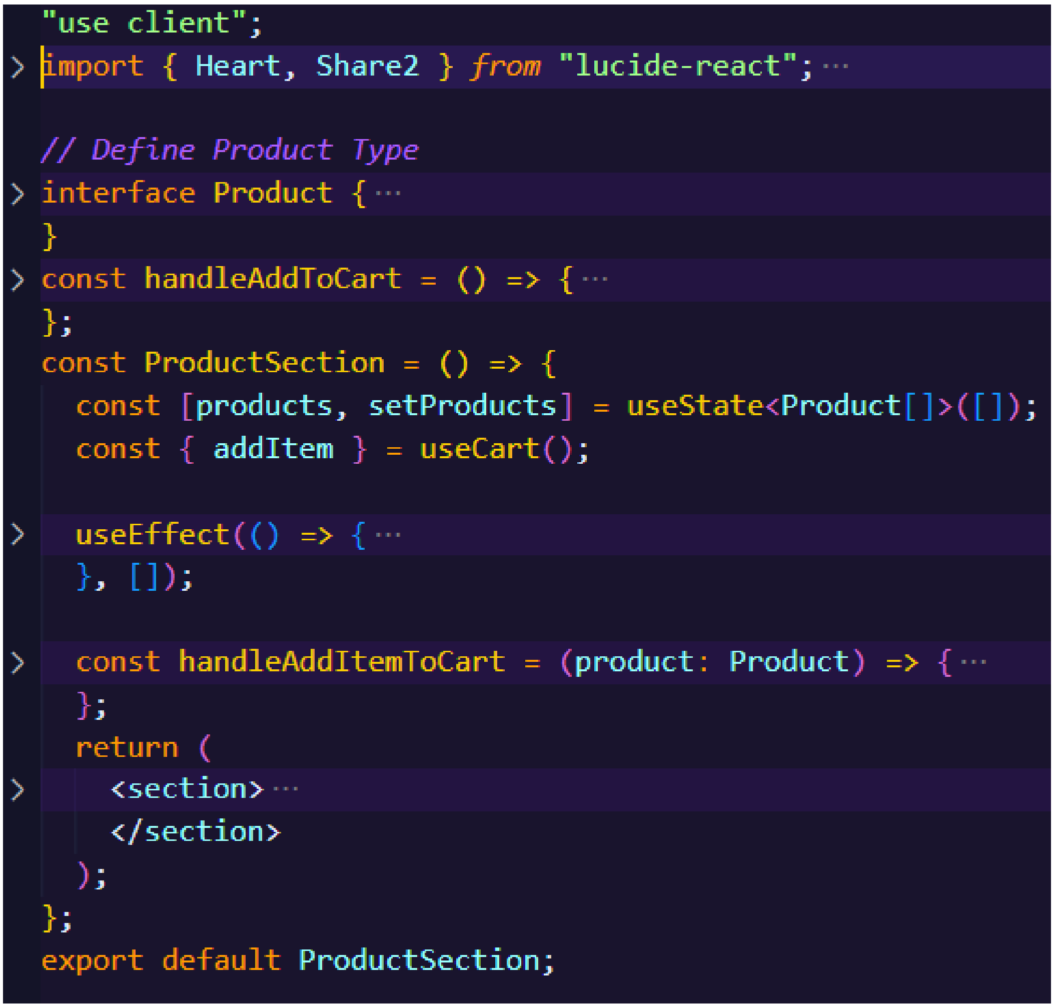
The goal of Day 4 was to design and develop dynamic frontend components to display marketplace data fetched from Sanity CMS or APIs.

The focus was on creating modular, reusable, and scalable components following industry best practices.

Implemented Components

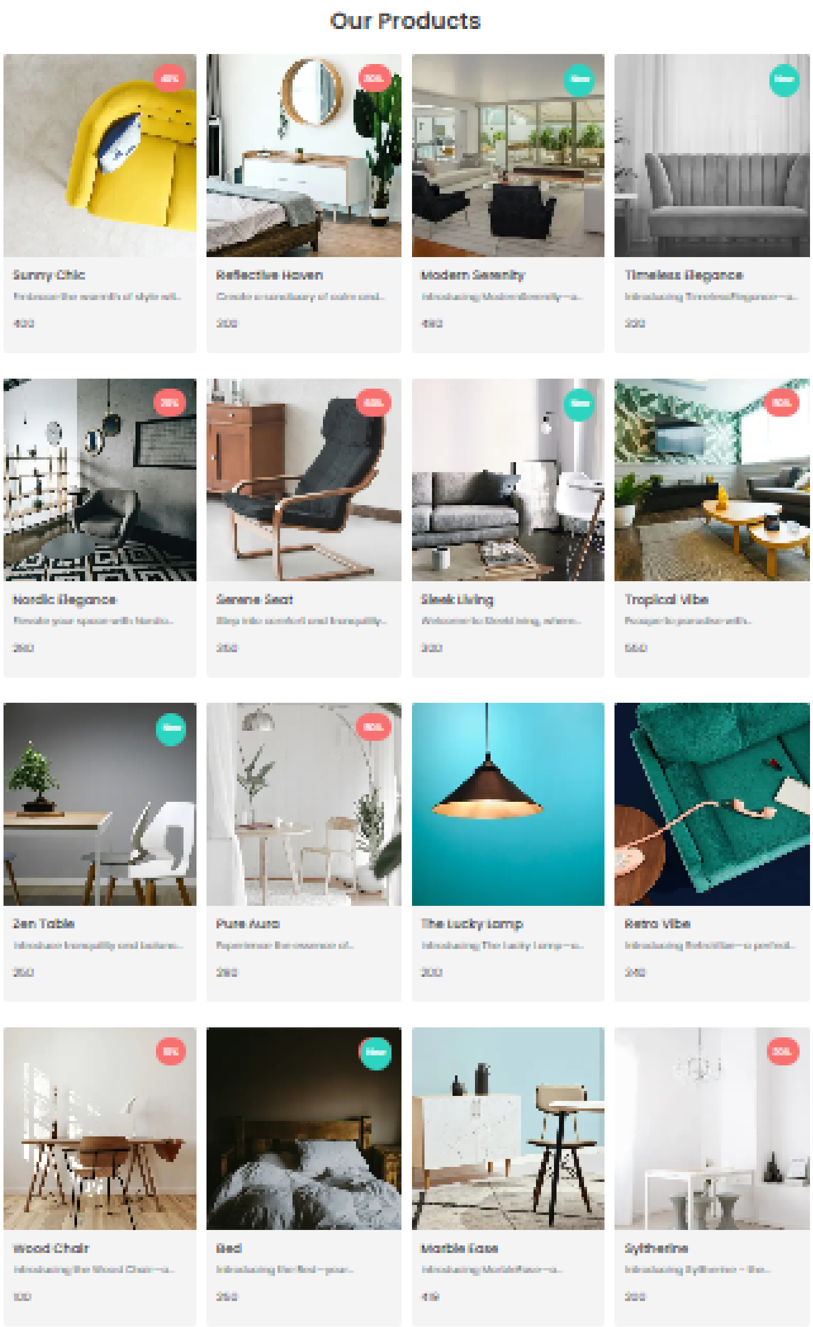
Product Listing Component

- Displays products dynamically in a grid layout.
- Includes fields such as:
  - Product Name
  - Price
  - Image
  - Tags
- Implemented using Next.js dynamic rendering and CSS Grid for layout.



```
"use client";
import { Heart, Share2 } from "lucide-react";⋯

// Define Product Type
interface Product {⋯
}
const handleAddToCart = () => {⋯
};
const ProductSection = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const { addItem } = useCart();

  useEffect(() => {⋯
  }, []);

  const handleAddItemToCart = (product: Product) => {⋯
  };
  return (
    <section>⋯
    </section>
  );
};
export default ProductSection;
```

Product Listing Component



Our Products

Sunny Chic
Embrace the warmth of style wit...
400

Reflective Haven
Create a sanctuary of calm and...
200

Modern Serenity
Introducing ModernSerenity—a...
490

Timeless Elegance
Introducing TimelessElegance—a...
220

Nordic Elegance
Elevate your space with Nordic...
260

Serene Seat
Step into comfort and tranquility...
250

Sleek Living
Welcome to Sleek living, where...
320

Tropical Vibe
Escape to paradise with...
550

Zen Table
Introduce tranquility and balance...
250

Pure Aura
Experience the essence of...
280

The Lucky Lamp
Introducing The Lucky Lamp—a...
200

Retro Vibe
Introducing RetroVibe—a perfect...
240

Wood Chair
Introducing the Wood Chair—a...
100

Bed
Introducing the Bed—your...
250

Marble Ease
Introducing MarbleEase—a...
410

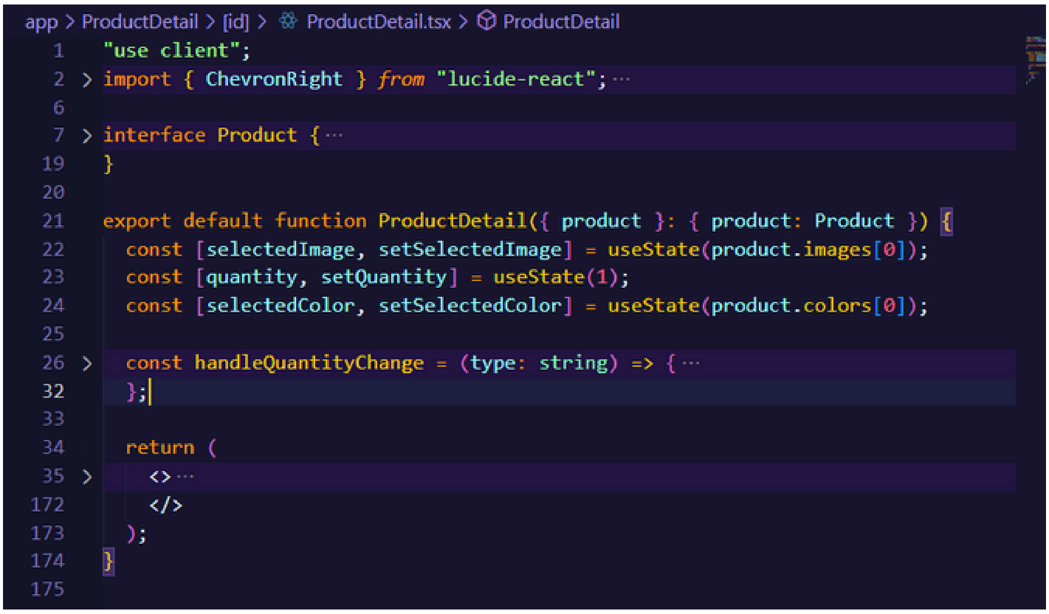
Sytherine
Introducing Sytherine – the...
200

Product Detail Component

- Displays individual product details using Next.js dynamic routing

- Shows additional fields such as:
  - Product Description
  - Price
  - Available Sizes/Colors

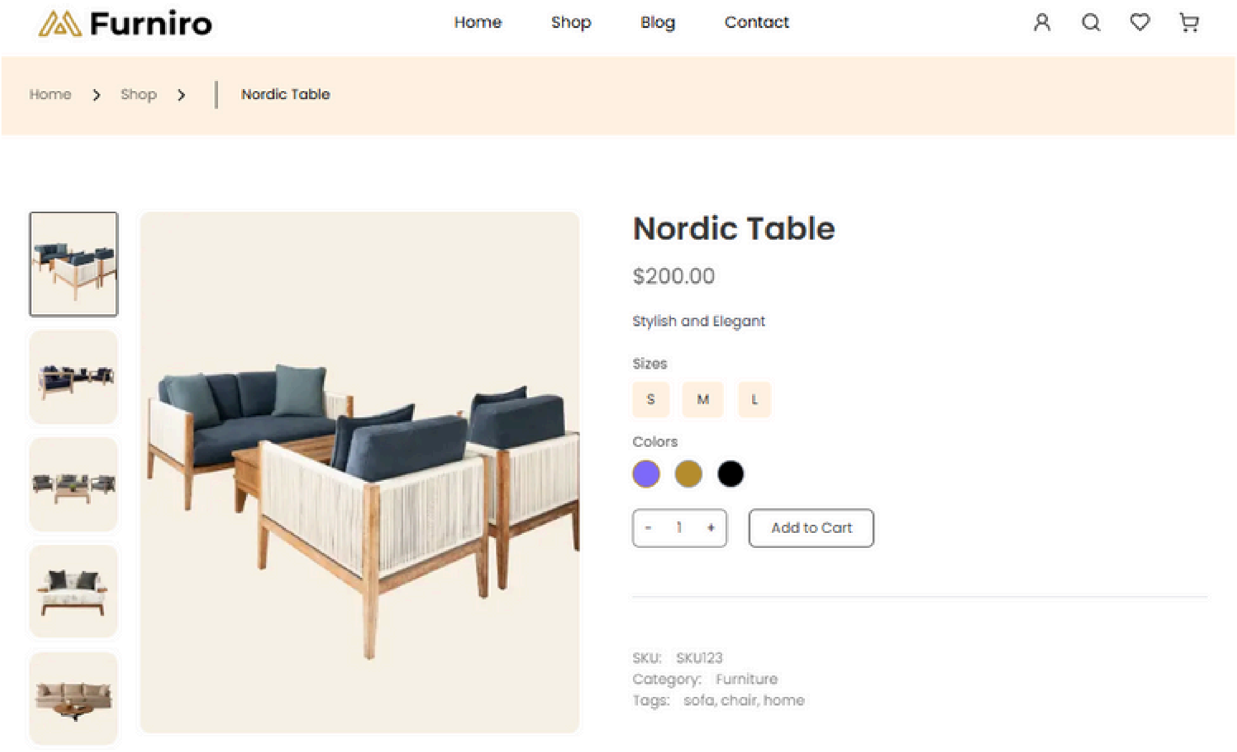
- Implemented using Next.js dynamic rendering and CSS Grid for layout.



app > ProductDetail > [id] > ProductDetail.tsx > ProductDetail

```
"use client";
import { ChevronRight } from "lucide-react";···

interface Product {···
}

export default function ProductDetail({ product }: { product: Product }) {
  const [selectedImage, setSelectedImage] = useState(product.images[0]);
  const [quantity, setQuantity] = useState(1);
  const [selectedColor, setSelectedColor] = useState(product.colors[0]);

  const handleQuantityChange = (type: string) => {···
  };

  return (
    <>···
    </>
  );
}
```

Product Detail Component



Home > Shop > | Nordic Table

Nordic Table

$200.00

Stylish and Elegant

Sizes

S M L

Colors

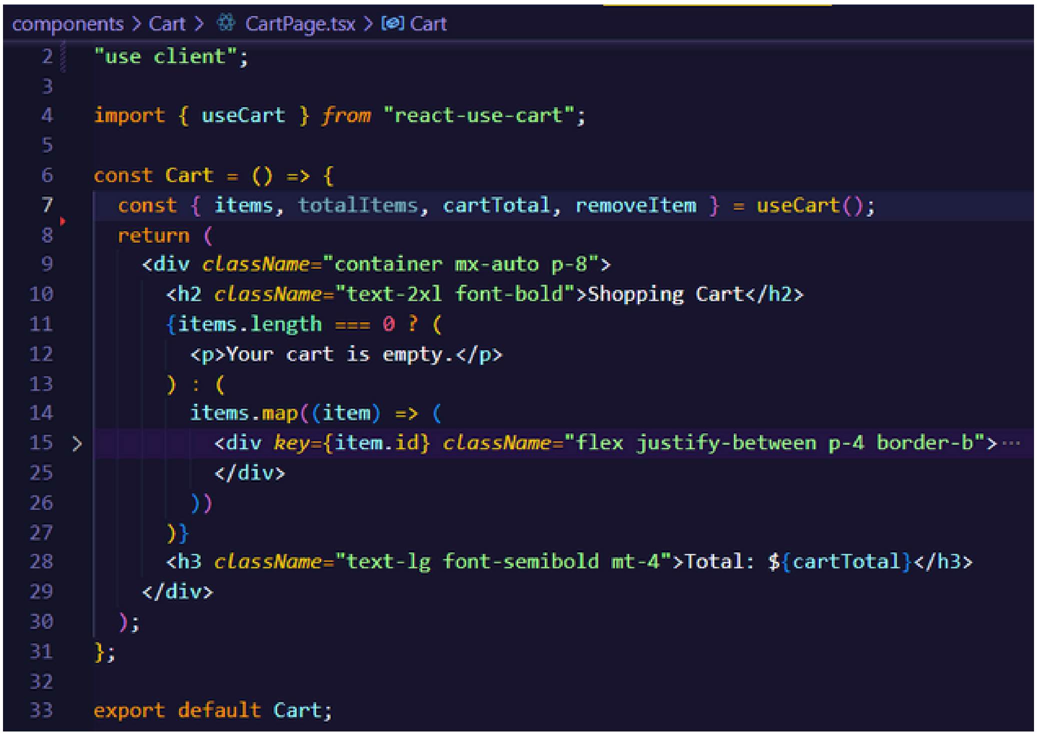
● ● ●

- 1 + Add to Cart

SKU: SKU123
Category: Furniture
Tags: sofa, chair, home

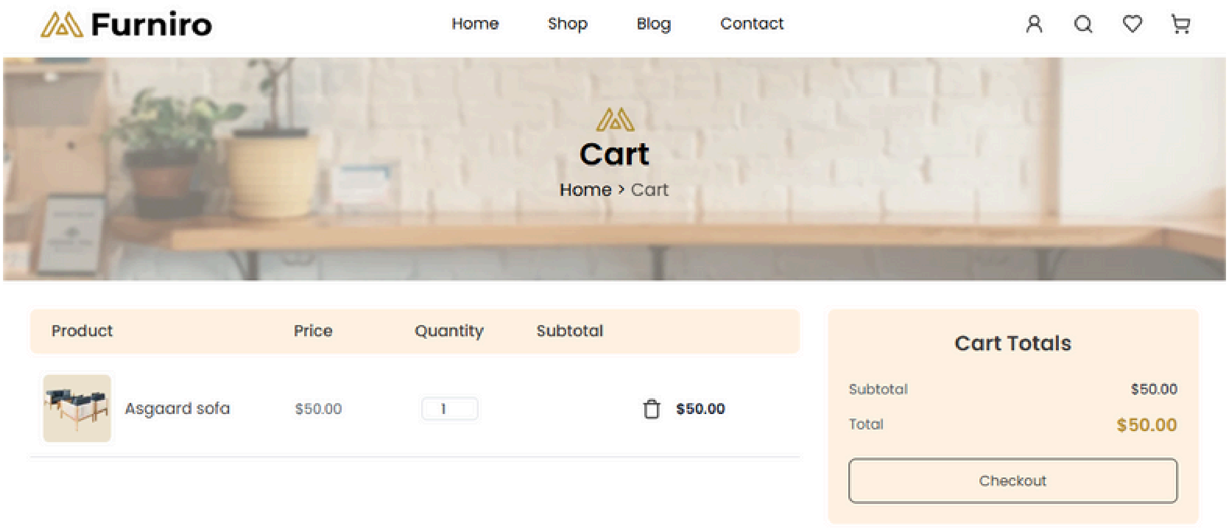
Cart Component

- Allows users to add/remove products from the cart.
- Uses react-use-cart for local cart management.
- Displays quantity and total price dynamically.



```
"use client";

import { useCart } from "react-use-cart";

const Cart = () => {
  const { items, totalItems, cartTotal, removeItem } = useCart();
  return (
    <div className="container mx-auto p-8">
      <h2 className="text-2xl font-bold">Shopping Cart</h2>
      {items.length === 0 ? (
        <p>Your cart is empty.</p>
      ) : (
        items.map((item) => (
          <div key={item.id} className="flex justify-between p-4 border-b">···
          </div>
        ))
      )}
      <h3 className="text-lg font-semibold mt-4">Total: ${cartTotal}</h3>
    </div>
  );
};

export default Cart;
```

Cart Component



Product	Price	Quantity	Subtotal
Asgaard sofa	$50.00	1	$50.00

Cart Totals

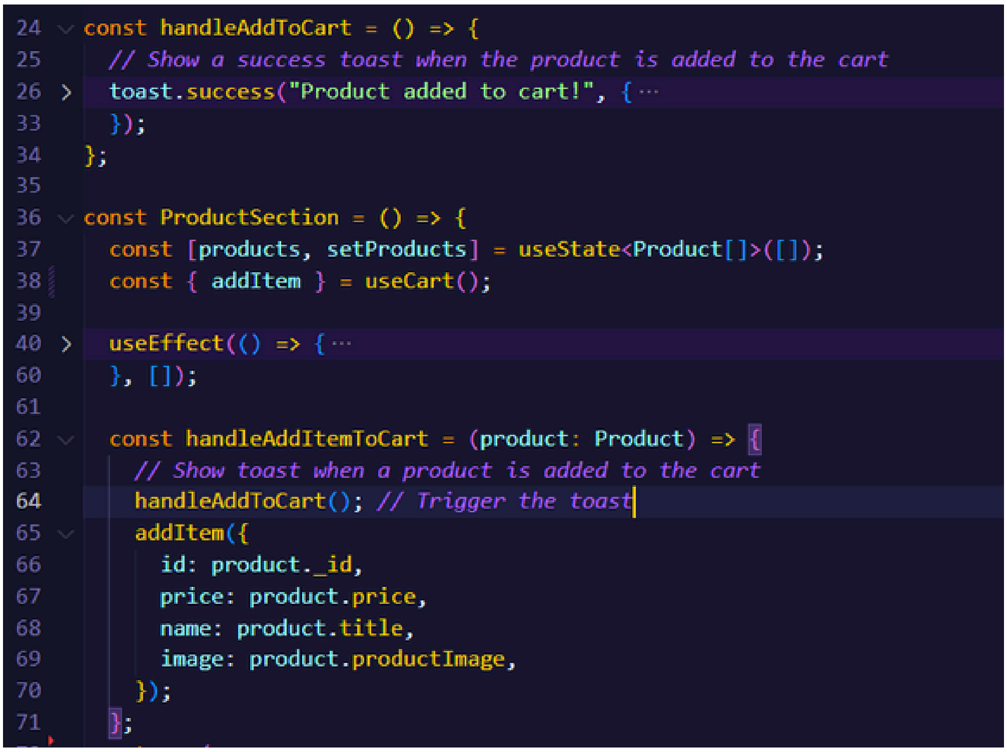
Subtotal	$50.00
Total	$50.00

Notifications Component

Provides real-time alerts for:

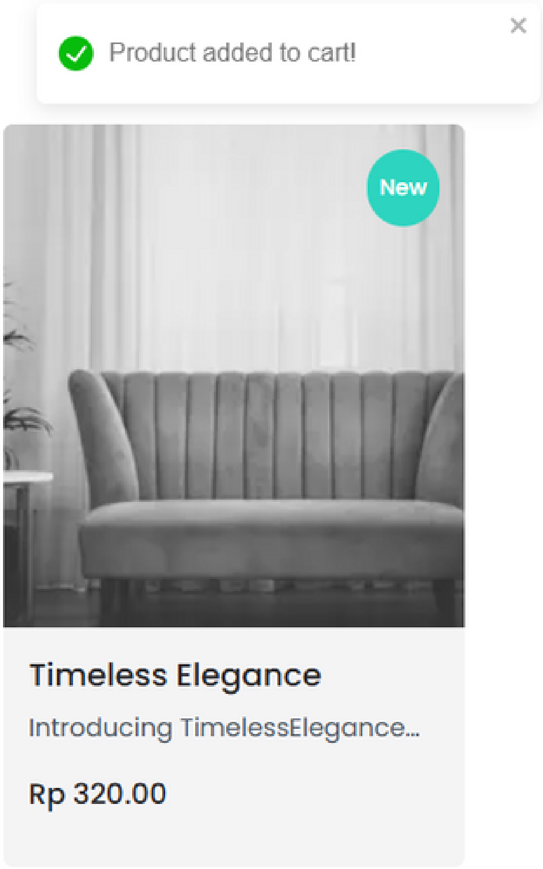
- Adding to cart
- Errors
- Successful purchases

- Implemented using React Toast Notifications.



```
const handleAddToCart = () => {
  // Show a success toast when the product is added to the cart
  toast.success("Product added to cart!", {···
  });
};

const ProductSection = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const { addItem } = useCart();

  useEffect(() => {···
  }, []);

  const handleAddItemToCart = (product: Product) => {
    // Show toast when a product is added to the cart
    handleAddToCart(); // Trigger the toast
    addItem({
      id: product._id,
      price: product.price,
      name: product.title,
      image: product.productImage,
    });
  };
```

Notifications Component



✓ Product added to cart! ×

Timeless Elegance

Introducing TimelessElegance...

Rp 320.00

Challenges and Solutions

State Management for Cart

Challenge: Managing cart data across multiple components.
Solution: Used use-react-cart to maintain global cart state.

Notification

Challenge: Adding notification on adding to cart, errors
Solution: Used react toastify

Responsive Design

Challenge: Ensuring mobile responsiveness.
Solution: Used Tailwind CSS for flexible layout adjustments.

Dynamic Frontend Components

PRESENTED BY:

Ayesha Abdul Qadir