Hackathon Project: E-Commerce Website Development

Project Overview

This project focuses on developing a feature-rich **E-commerce website** where users can explore a wide range of products, including:

- **Computer-related items**: Headphones, gaming accessories, monitors, and furniture (e.g., computer desks).
- **General products** with an emphasis on dynamic data handling and interactive user experience.

Key functionalities include:

1. Dynamic product listing and routing.
2. Advanced search and filtering.
3. Add to cart and wishlist systems.
4. User authentication and checkout flow.
5. Integration of Sanity CMS for data management.

Features and Functionalities

1. **Dynamic Product Routing**
    a. Users can click on any product to view detailed information on a dynamically generated page.
    b. Includes data such as product title, price, description, specifications, and stock availability.
2. **Search System**
    a. Users can search products by category or title.
    b. Optimized search functionality using Sanity's GROQ queries.
3. **Filtering and Sorting**
    a. **Filters**: By category.
    b. **Sorting Options**:
        i. Price: Low to High and High to Low.
        ii. Ratings: High to Low.
        iii. Most Reviewed products.
4. **Add to Cart and Wishlist**

   a. Users can add products to a cart or wishlist.
   b. Cart updates in real-time, showing quantities and total price.
5. **Checkout Flow**
   a. Includes user information collection (name, email, address, city, postal code, country).
   b. Order placement confirmation.
   c. Currently simulates order placement without actual payment integration (e.g., PayPal or JazzCash).
6. **Backend and CMS Integration**
   a. Custom-built API data migrated to Sanity CMS.
   b. GROQ queries used to fetch data dynamically for the frontend.
7. **Team Collaboration**
   a. Developed in collaboration with **Zohaib Shah** and **Ahmed Saeed**, with mutual assistance and knowledge sharing.

Technical Stack

- **Frontend**:
  - Next.js (React Framework)
  - Tailwind CSS for responsive design
  - TypeScript for type safety
  - React Toastify for notifications
- **Backend**:
  - Custom-built APIs
  - Sanity CMS for data management and GROQ queries
- **Hosting**:
  - Vercel for live deployment

Key Code Snippets

Below are some crucial code snippets highlighting core features:

1. **Dynamic Routing for Product Detail Pages**:

```
import { client } from '@/sanity/lib/client';
import { getProductDetailsByIdQuery } from
'@/sanity/lib/queries';
import ProductPage from "@/components/Dynamic-Page";
import RelatedItems from "@/components/Related-Item";

export default async function ProductPageDynamic({ params }:
{ params: { id: string } }) {
    const product = await
client.fetch(getProductDetailsByIdQuery, { id: params.id });

    if (!product) {
        return <div>Product not found</div>;
    }

    return (
        <main className="container mx-auto px-4 py-8">
            <ProductPage product={product} />
            <RelatedItems category={product.category} />
        </main>
    );
}
```

2. **Add to Cart Functionality:**

```
'use client';

import React, { createContext, useReducer, useContext,
ReactNode, useEffect } from 'react';
```

```
// Define the types
interface CartItem {
    id: number;
    name: string;
    price: number;
    quantity: number;
    image: string | unknown;
}

interface CartState {
    cartItems: CartItem[];
}

interface CartContextType {
    cart: CartItem[];
    addItem: (item: CartItem) => void;
    updateQuantity: (id: number, newQuantity: number) => void;
    removeItem: (id: number) => void;
    clearCart: () => void;
    getTotal: () => string; // Total as a formatted string
}

const CartContext = createContext<CartContextType |
undefined>(undefined);

// Reducer function
```

```
const cartReducer = (state: CartState, action: any): CartState
=> {
    switch (action.type) {
        case 'ADD_ITEM': {
            const existingItem = state.cartItems.find((item) =>
item.id === action.payload.id);
            if (existingItem) {
                return {
                    ...state,
                    cartItems: state.cartItems.map((item) =>
                        item.id === action.payload.id
                            ? { ...item, quantity:
item.quantity + action.payload.quantity }
                            : item
                    ),
                };
            }
            return { ...state, cartItems: [...state.cartItems,
action.payload] };
        }
        case 'UPDATE_QUANTITY': {
            const updatedItems = state.cartItems.map((item) =>
                item.id === action.payload.id
                    ? { ...item, quantity: Math.max(1,
action.payload.newQuantity) }
                    : item
            );
            return { ...state, cartItems: updatedItems };
        }

        case 'REMOVE_ITEM':
            return {
                ...state,
                cartItems: state.cartItems.filter((item) =>
item.id !== action.payload.id),
            };
        case 'CLEAR_CART':
            return { ...state, cartItems: [] };
        default:
            throw new Error(`Unhandled action type:
${action.type}`);
    }
};

// Context Provider
export const CartProvider = ({ children }: { children:
ReactNode }) => {
    const [state, dispatch] = useReducer(cartReducer,
{ cartItems: [] });

    // Load cart from localStorage on initial load
    useEffect(() => {
        const savedCart = localStorage.getItem('cart');
        if (savedCart) {
            const parsedCart: CartItem[] =
JSON.parse(savedCart);
            parsedCart.forEach((item) => {
                // Ensure cart state is updated correctly with
the saved items
                dispatch({ type: 'ADD_ITEM', payload: item });
            });
        }
    }, []);

    // Save cart to localStorage whenever the cart changes
    useEffect(() => {
        if (state.cartItems.length > 0) {
            localStorage.setItem('cart',
JSON.stringify(state.cartItems));
        } else {
            localStorage.removeItem('cart');
        }
    }, [state.cartItems]);

    const addItem = (item: CartItem) => dispatch({ type:
'ADD_ITEM', payload: item });

    const updateQuantity = (id: number, newQuantity: number) =>
        dispatch({ type: 'UPDATE_QUANTITY', payload: { id,
newQuantity } });
```

```
    const removeItem = (id: number) => dispatch({ type:
'REMOVE_ITEM', payload: { id } });

    const clearCart = () => dispatch({ type: 'CLEAR_CART' });

    const getTotal = () => {
        const total = state.cartItems.reduce(
            (sum, item) => sum + item.price * item.quantity,
            0
        );
        return total.toFixed(2); // Ensure 2 decimal places
    };

    return (
        <CartContext.Provider
            value={{
                cart: state.cartItems,
                addItem,
                updateQuantity,
                removeItem,
                clearCart,
                getTotal, // Expose the total calculation
            }}
        >
            {children}
        </CartContext.Provider>
    );
```

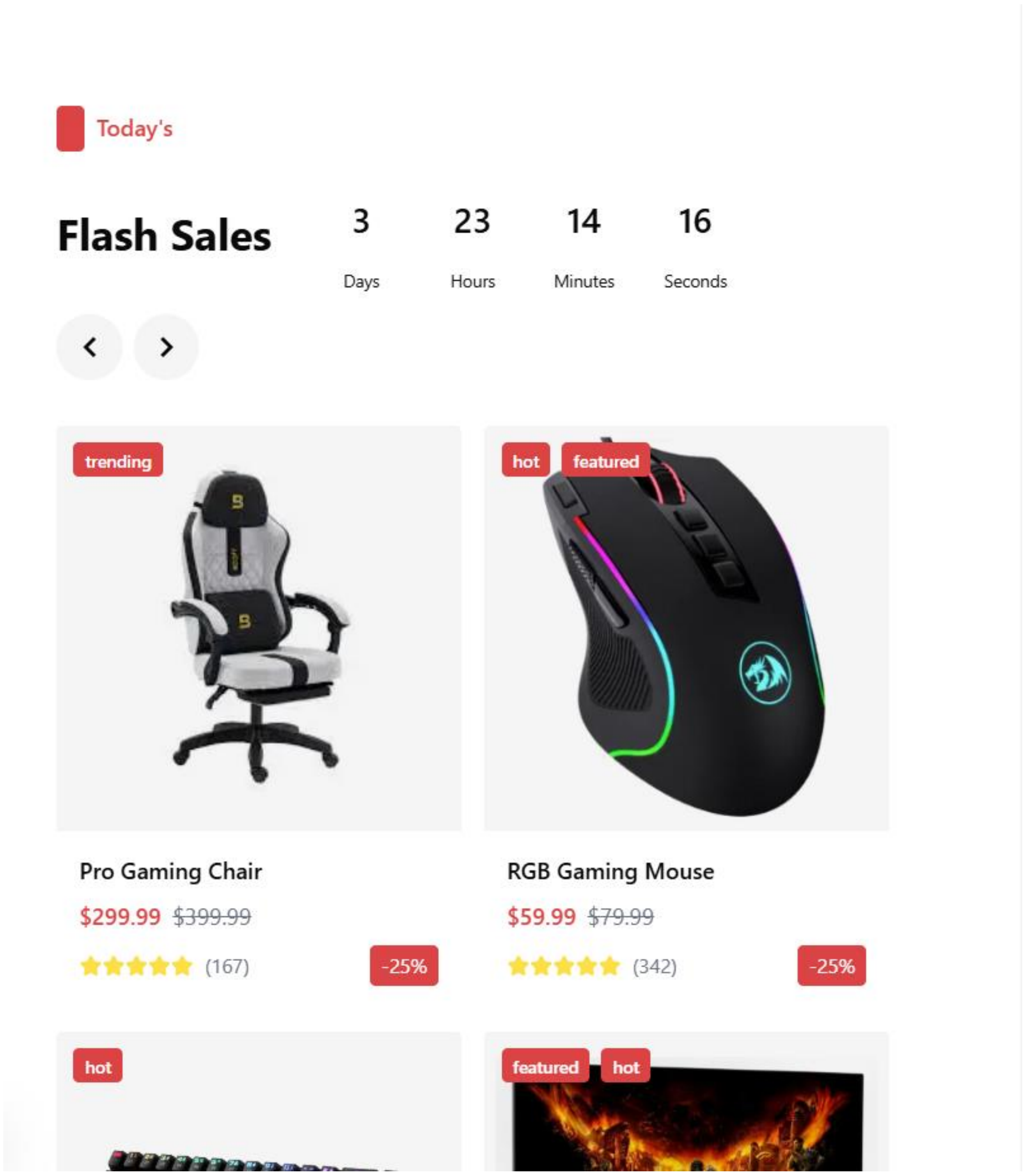
```
};

// Custom Hook
export const useCart = () => {
    const context = useContext(CartContext);
    if (!context) {
        throw new Error('useCart must be used within a
CartProvider');
    }
    return context;
};
```

Visuals

Here are some suggested screenshots to include in the final PDF:

1. **Homepage**:



Today's

Flash Sales

3	23	14	16
Days	Hours	Minutes	Seconds

Pro Gaming Chair

$299.99 ~~$399.99~~

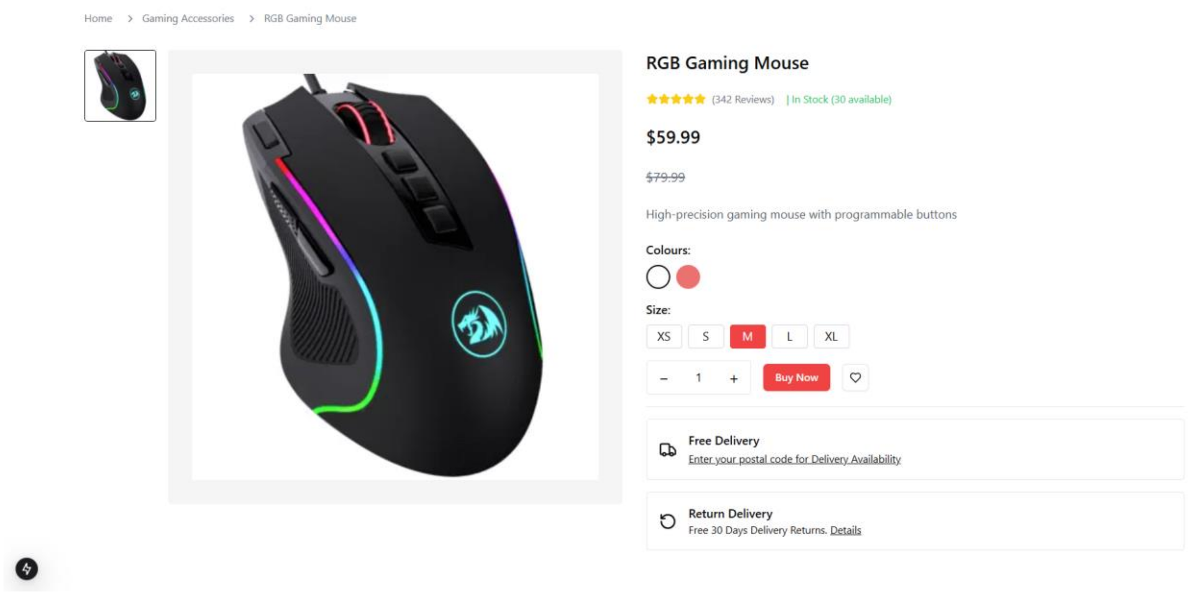
(167) -25%

RGB Gaming Mouse

$59.99 ~~$79.99~~

(342) -25%

2. **Product Detail Page**:


Home > Gaming Accessories > RGB Gaming Mouse

RGB Gaming Mouse

★★★★★ (342 Reviews) | In Stock (30 available)

$59.99

~~$79.99~~

High-precision gaming mouse with programmable buttons

Colours:

Size:

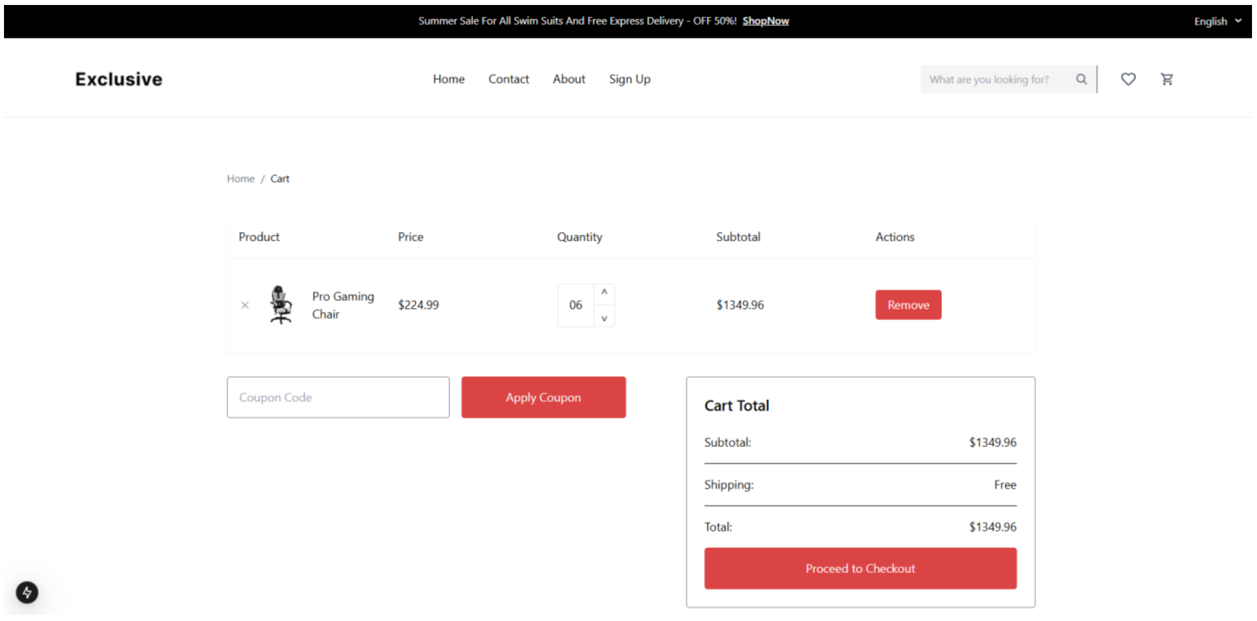
XS S M L XL

– 1 + Buy Now

Free Delivery
Enter your postal code for Delivery Availability

Return Delivery
Free 30 Days Delivery Returns. Details

3. **Cart Page:**


Summer Sale For All Swim Suits And Free Express Delivery - OFF 50%! ShopNow

English

Exclusive

Home Contact About Sign Up

What are you looking for?

Home / Cart

Product	Price	Quantity	Subtotal	Actions
× Pro Gaming Chair	$224.99	06	$1349.96	Remove

Coupon Code

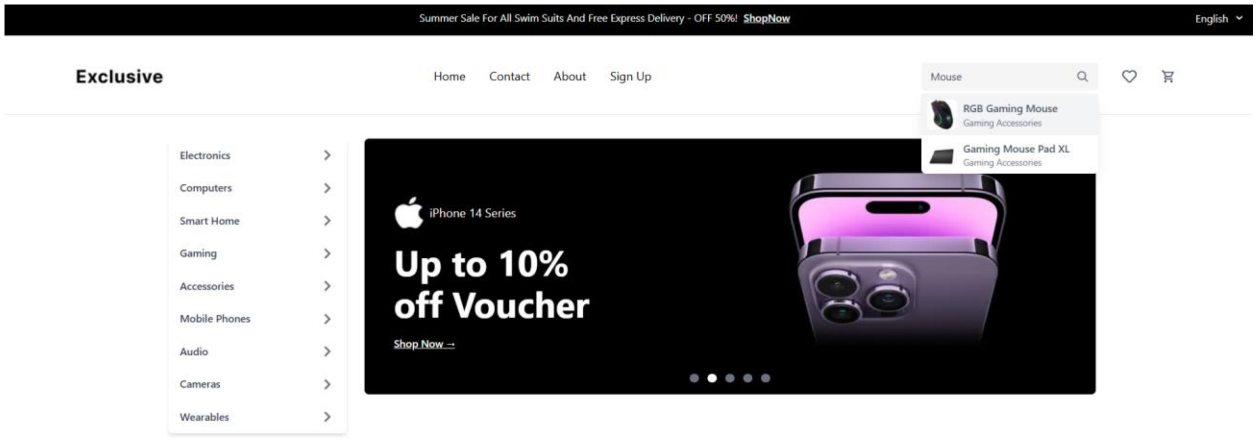
Apply Coupon

Cart Total

Subtotal:	$1349.96
Shipping:	Free
Total:	$1349.96

Proceed to Checkout

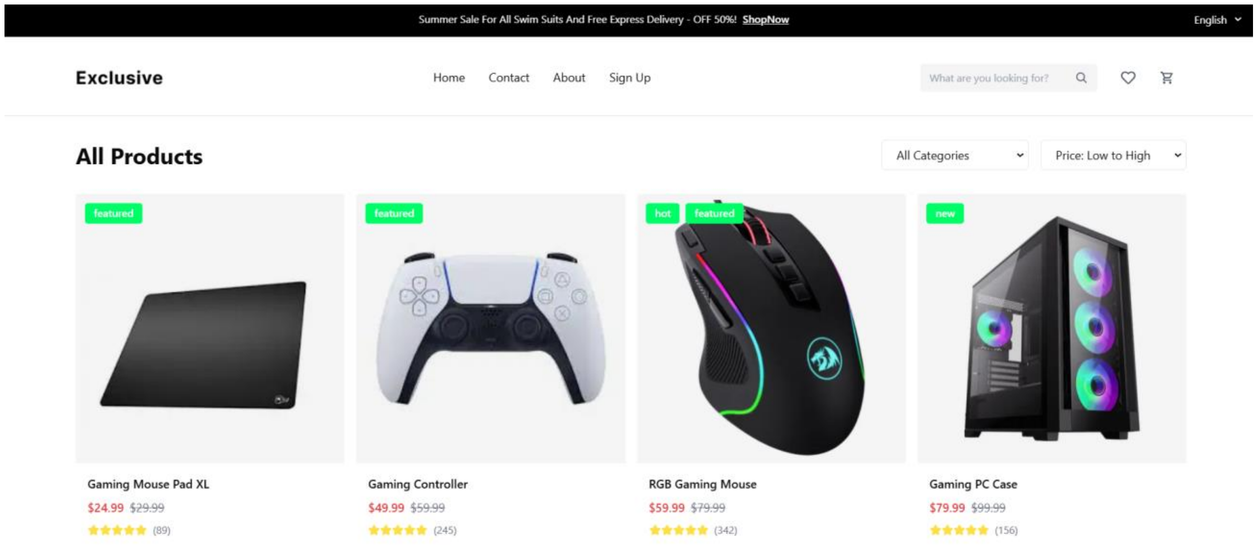
4. **Search and Filter:**


Summer Sale For All Swim Suits And Free Express Delivery - OFF 50%! ShopNow
English

Exclusive
Home Contact About Sign Up
Mouse
RGB Gaming Mouse
Gaming Accessories
Gaming Mouse Pad XL
Gaming Accessories

Electronics
Computers
Smart Home
Gaming
Accessories
Mobile Phones
Audio
Cameras
Wearables

iPhone 14 Series
Up to 10% off Voucher
Shop Now →


Summer Sale For All Swim Suits And Free Express Delivery - OFF 50%! ShopNow
English

Exclusive
Home Contact About Sign Up
What are you looking for?

All Products
All Categories
Price: Low to High

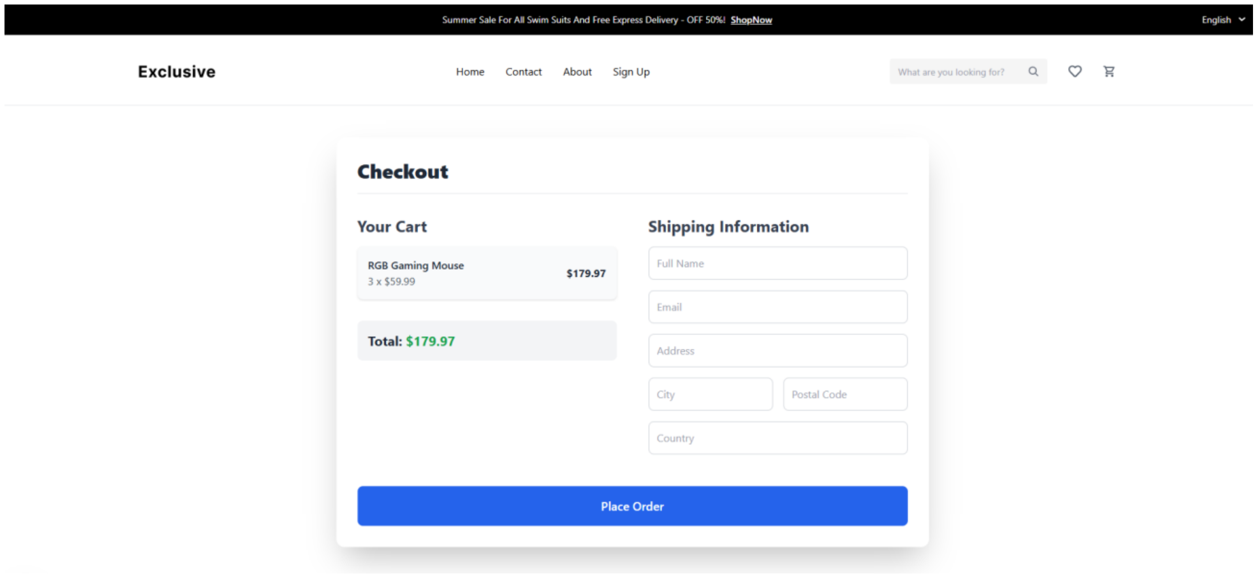
featured
Gaming Mouse Pad XL
$24.99 $29.99
(89)

featured
Gaming Controller
$49.99 $59.99
(245)

hot featured
RGB Gaming Mouse
$59.99 $79.99
(342)

new
Gaming PC Case
$79.99 $99.99
(156)

5. **Checkout Flow**:


Summer Sale For All Swim Suits And Free Express Delivery - OFF 50%! ShopNow
English

Exclusive
Home Contact About Sign Up
What are you looking for?

Checkout

Your Cart
RGB Gaming Mouse
3 x $59.99
$179.97

Total: $179.97

Shipping Information
Full Name
Email
Address
City
Postal Code
Country

Place Order

Future Improvements

- Add payment gateway integrations (e.g., PayPal, JazzCash).
- Enhance the ordering system with real-time status updates.
- Implement admin-side order and stock management.

Live Project Links

- **GitHub Repository**: https://github.com/Shahheerr/3rdHackathon-4thDay