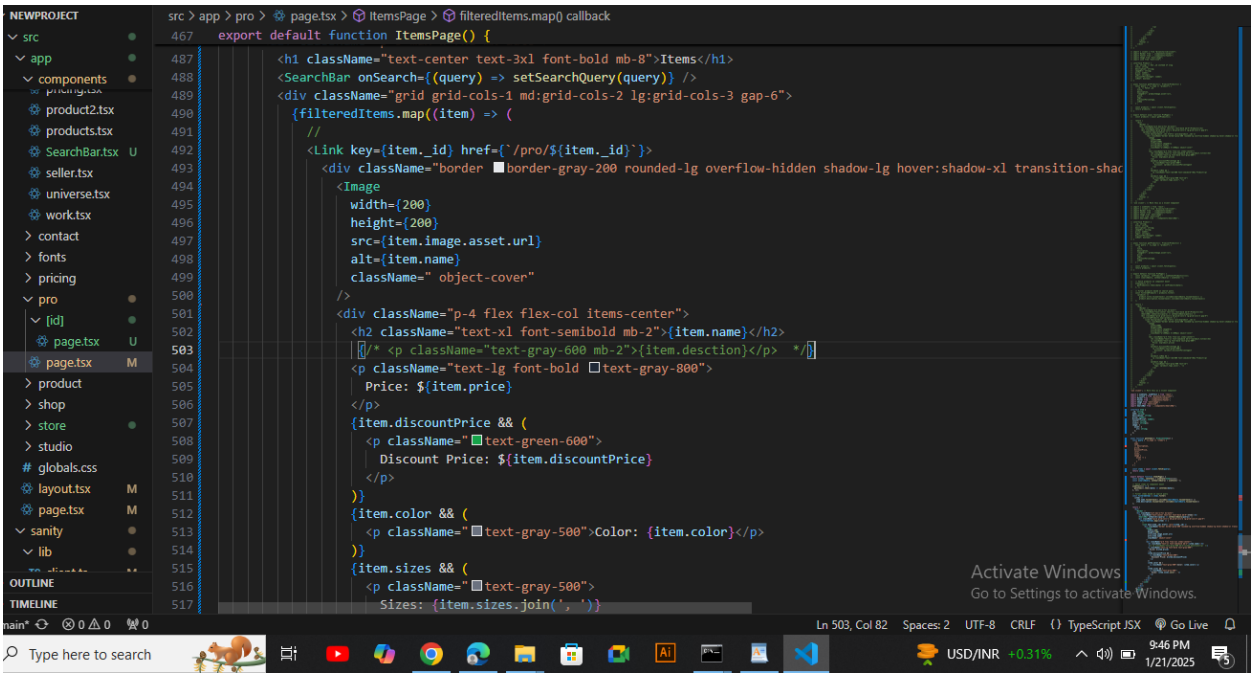
Day 4 Documentation

Building_Dynamic_Frontend_componet

The **Builder Hackathon** is an exciting platform where I can unleash my creativity and technical skills to build innovative solutions. During the event, I'll be focusing on creating dynamic and interactive components for my project, integrating real-time data from Sanity CMS to , search functionality, and user dashboards. With a strong emphasis on collaboration and problem-solving, this hackathon is the perfect opportunity to learn, experiment, and deliver a functional, user-friendly marketplace. Let's code, create, and conquer!

1.Dynamic Routing

Dynamic routing allows your application to handle URLs dynamically, such as `/products/:id` or `/category/:slug`, where placeholders like `:id` or `:slug` are replaced with actual values at runtime. This approach is ideal for creating scalable, user-friendly pages like product details, user profiles, or blog posts without hardcoding each route. It ensures a seamless and flexible navigation experience.



Fancy Dresses

Price: $700

Discount Price: $650

Color: Red,Blue,Green

Sizes: S, M, L

Gown style

Price: $400

Discount Price: $370

Color: red,green

Sizes: S, M, L

casual wear

Price: $400

Discount Price: $340

Color: Red,Blue,Green

Sizes: S, M, L

Jacket

Price: $500

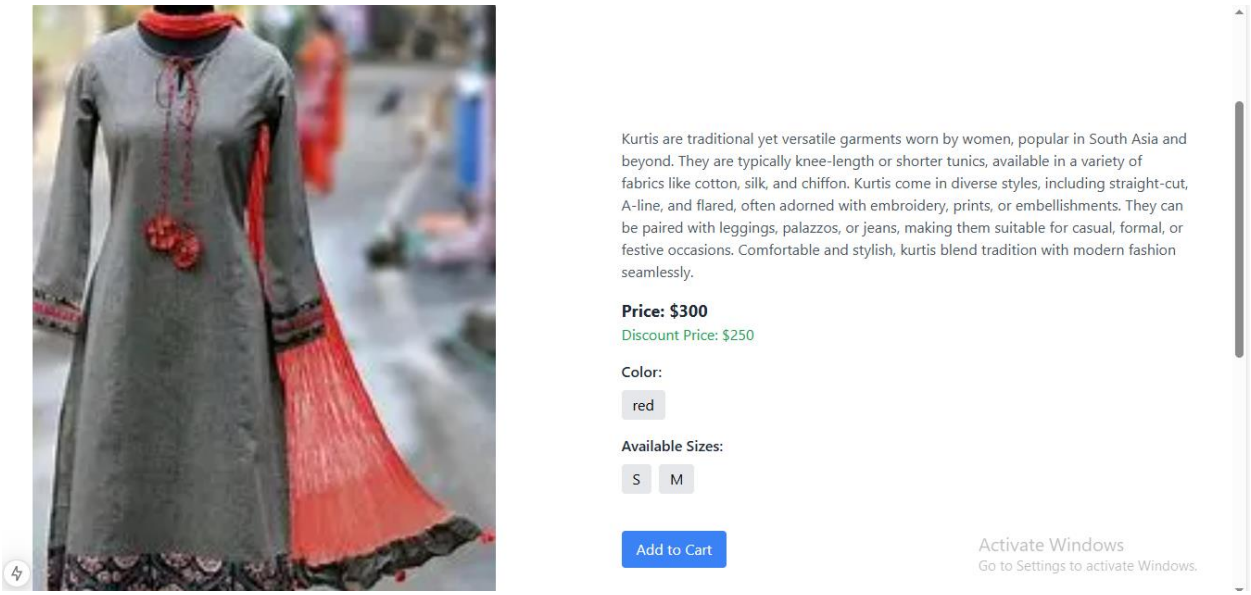
Discount Price: $450

Color: red ,blue,gree

Sizes: S, M, L

shirts

Kurtis style

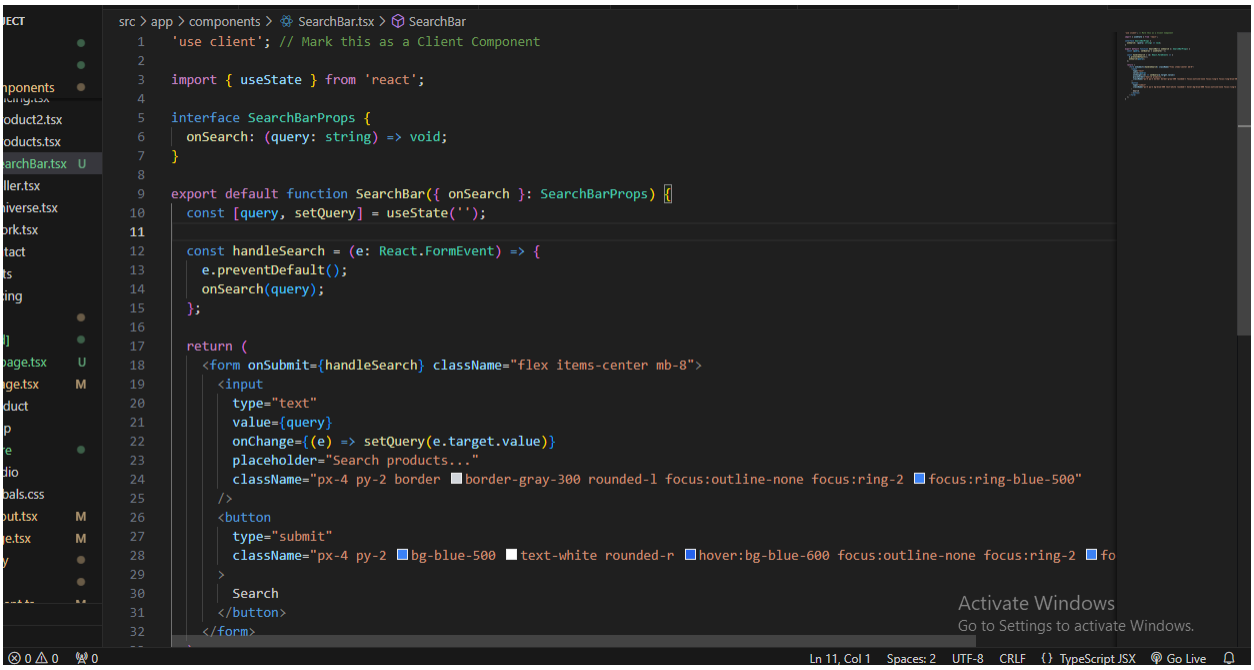


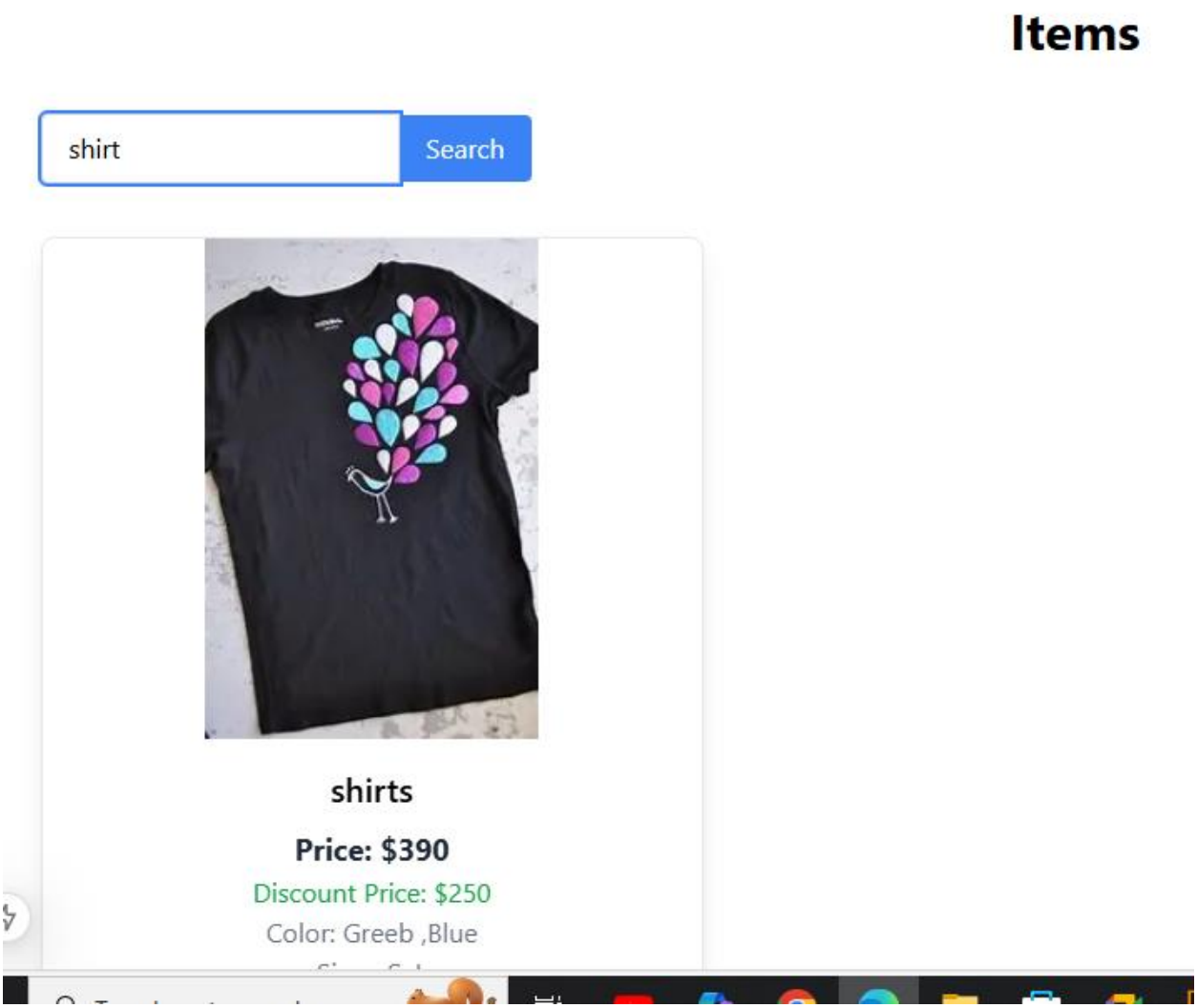
Successfully Done!

Dynamic routing has been implemented, allowing the application to handle URLs like `/products/:id` or `/category/:slug` dynamically. This ensures scalable, user-friendly navigation for pages such as product details, user profiles, or blog posts.

2.Search Functionality;

A quick and efficient search feature has been added, enabling users to find products or content instantly. With real-time suggestions, filters, and dynamic results, it ensures a smooth and user-friendly experience.


src > app > components > SearchBar.tsx > SearchBar
'use client'; // Mark this as a Client Component
import { useState } from 'react';
interface SearchBarProps {
onSearch: (query: string) => void;
}
export default function SearchBar({ onSearch }: SearchBarProps) {
const [query, setQuery] = useState('');
const handleSearch = (e: React.FormEvent) => {
e.preventDefault();
onSearch(query);
};
return (
<form onSubmit={handleSearch} className="flex items-center mb-8">
<input
type="text"
value={query}
onChange={(e) => setQuery(e.target.value)}
placeholder="Search products..."
/>
<button
type="submit"
>
Search
</button>
</form>
Activate Windows
Go to Settings to activate Windows.
Ln 11, Col 1
Spaces: 2
UTF-8
CRLF
TypeScript JSX
Go Live



Search Functionality Successfully Created! 🎉
The search feature is now fully functional, allowing users to find items by name or description. It handles edge cases like missing descriptions gracefully and ensures a smooth, error-free experience.

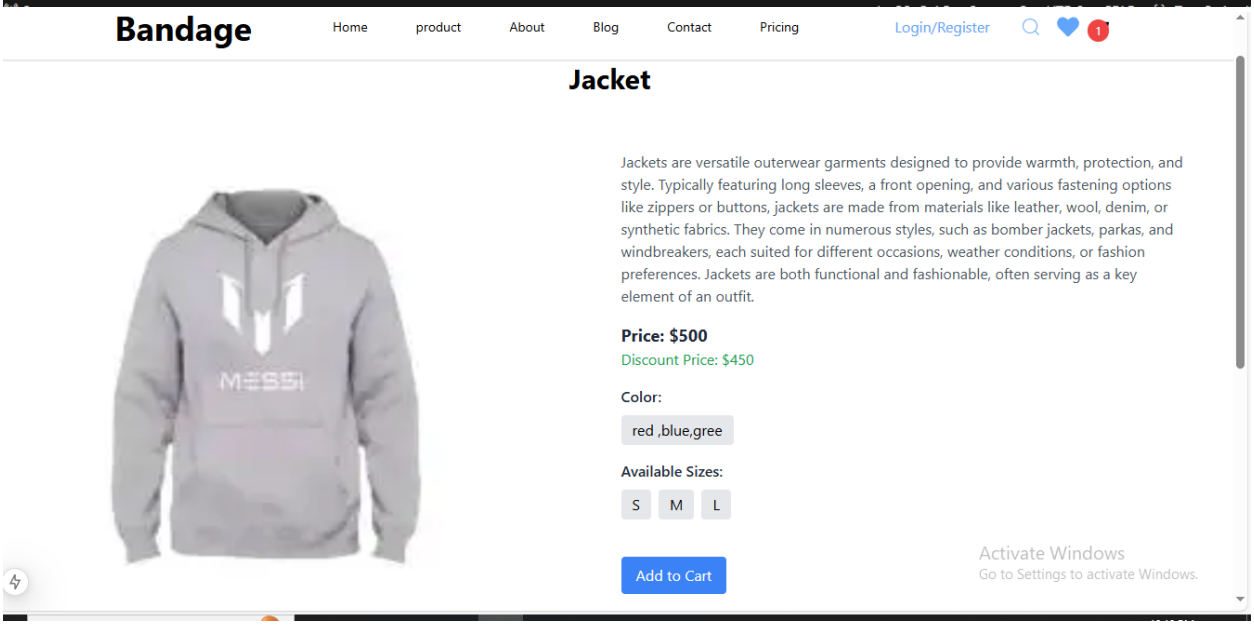
3.Add To Cart

The "Add to Cart" feature allows users to easily add products to their shopping cart. Here's how it works:

1. **Add Items**: Users can click a button to add products to their cart.
2. **Real-Time Updates**: The cart dynamically updates to show the number of items, total price, and a summary of added products.

3. **Manage Cart**: Users can adjust quantities, remove items, or proceed to checkout.

```tsx
'use client'; // Mark this as a Client Component

import { useCartStore } from "../store/cartStore";

interface AddToCartButtonProps {
  product: {
    id: string;
    name: string;
    price: number;
    imageUrl: string;
    quantity:number;
  };
}

export default function AddToCartButton({ product }: AddToCartButtonProps) {
  const addToCart = useCartStore((state:any) => state.addToCart);

  const handleAddToCart = () => {
    addToCart(product);
    alert('Product added to cart!');
  };

  return (
    <button
      onClick={handleAddToCart}
      className="mt-4 bg-blue-500 text-white px-4 py-2 rounded hover:bg-blue-600"
    >
      Add to Cart
    </button>
  );
}
```

Add to Cart Successfully Implemented! 🎉

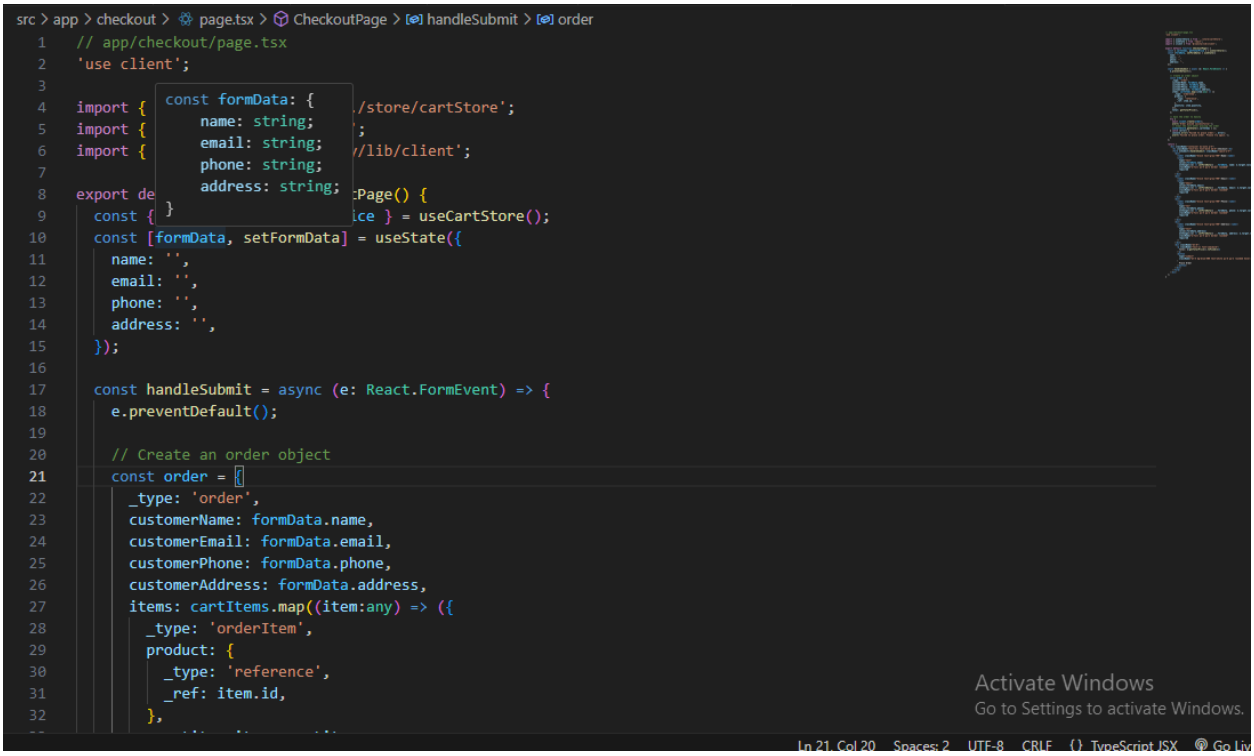
The "Add to Cart" feature is now fully functional and ready to use. Users can:

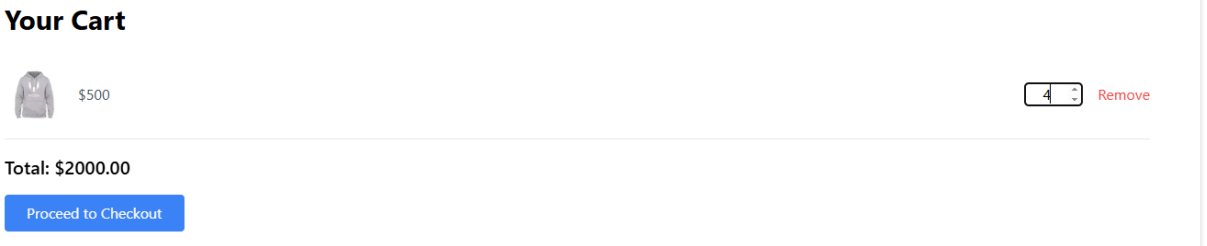
1. **Add products** to their cart with a single click.
2. **View real-time updates** (e.g., item count, total price).
3. **Manage their cart** (e.g., update quantities, remove items).

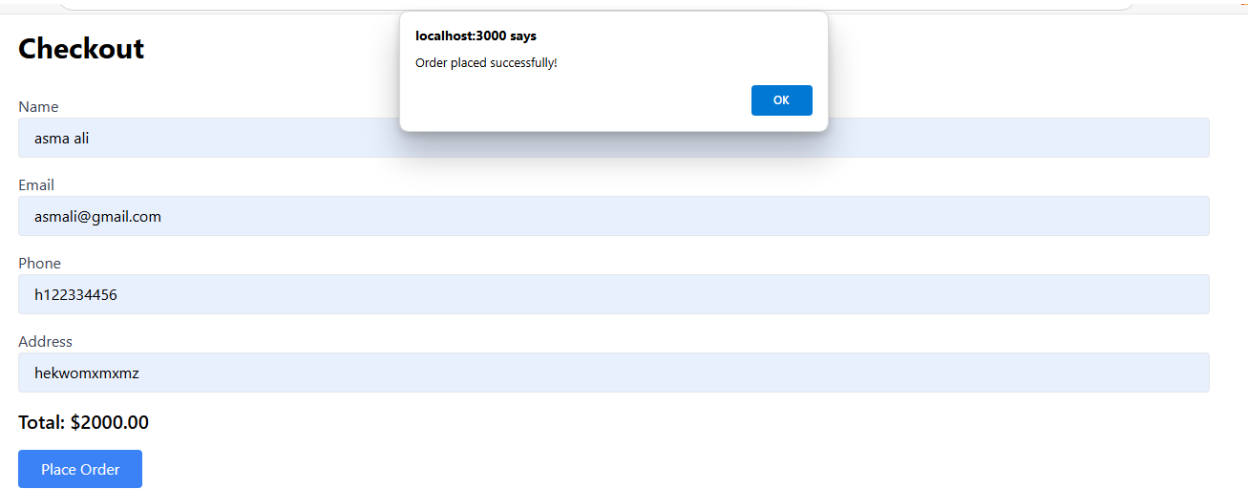
4.Checkout Proceed;

The checkout process is the step where users purchase the items in their cart. It includes these features:

1. **Order Summary**: Displays items, quantities, and total price in the cart.
2. **Shipping**: Users enter their address and choose delivery options.
3. **Payment**: Secure payment methods (cards, PayPal, etc.) are used to complete the transaction.
4. **Confirmation**: After the order is confirmed, details are shown on-screen and sent via email.







Checkout Successfully Implemented! 🎉

The checkout process is now fully functional and ready to use. Users can:

1. **Review their order** (items, quantities, total price).
2. **Enter shipping details** and choose delivery options.
3. **Pay securely** via multiple payment methods.
4. **Receive confirmation** with an order summary and email receipt.