

# HINA NADIR MUGHAL



## Day 4 Implementat

Our rental e-commerce website simplifies finding and renting properties. Users can browse a variety of listings, from living room furniture to vacation rentals, with advanced search filters and detailed property information.

For property owners, the platform offers easy tools to list rentals, manage bookings, and connect with tenants. With a user-friendly interface and secure transactions, it creates a seamless and efficient rental experience for all!

### Purpose of the Furniture Rental E-Commerce Website

The purpose of this website is to provide a convenient platform for users to rent furniture based on their needs and budget. It aims to make furniture accessible, affordable, and hassle-free, offering flexible rental plans for homes, offices, or events. By connecting users with reliable furniture providers, the platform promotes sustainability through reuse and eliminates the need for costly purchases.

### Steps to Build and Integrate Components

1. **Planning:** Defined goals, target audience, and designed wireframes.
2. **Setup:** Installed Next.js, Tailwind CSS, and configured the development environment.
3. **Frontend Development:** Built responsive pages (Home, Listings, Product Details, Dashboard) using Tailwind CSS.
4. **Backend Integration:** Connected a database for managing users, products, and orders.
5. **Core Features:** Added search, filters, cart, and secure checkout with payment gateway.

That's the thing that we're gonna learn in next quarter

6. **User Authentication:** Implemented login, registration, and account management.
7. **Admin Panel:** Created tools to manage products, transactions, and user queries.
8. **Testing & Deployment:** Tested functionality, resolved bugs, and deployed on Vercel...

### Conclusion

The furniture rental e-commerce website successfully bridges the gap between affordability and convenience, offering users a flexible solution to meet their furniture needs. By integrating a

user-friendly interface, advanced search filters, secure payment systems, and efficient backend management, the platform ensures a seamless experience for both renters and providers.

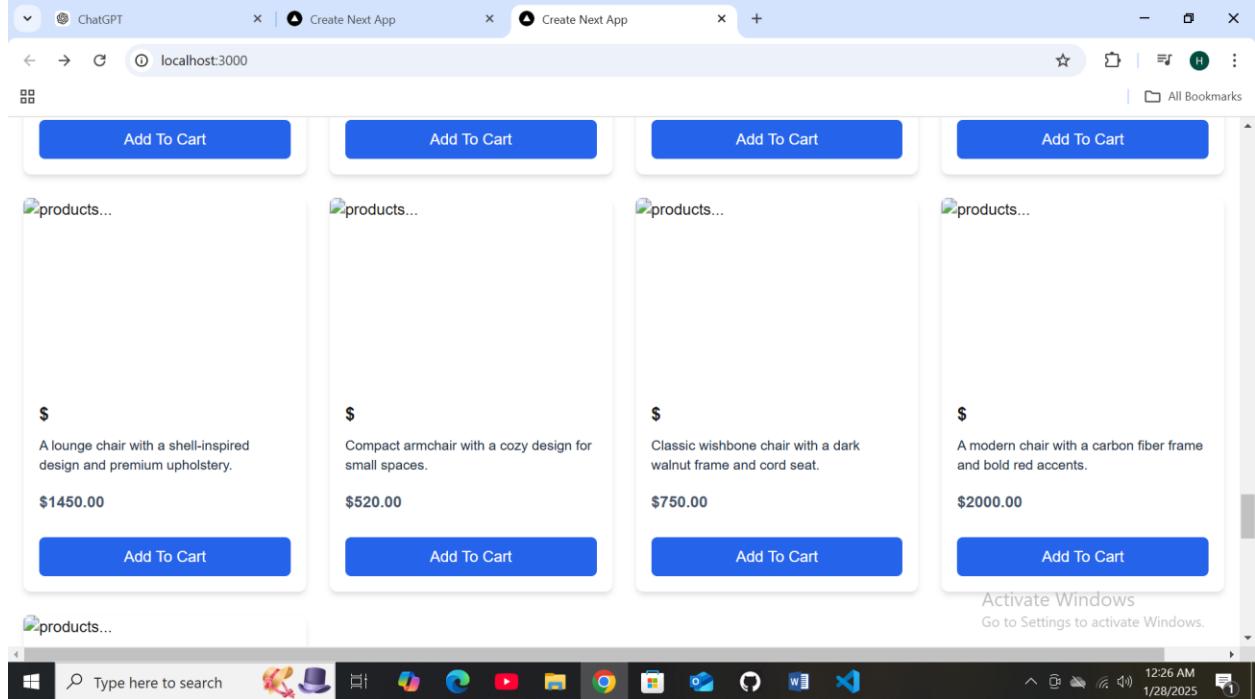
This project highlights the importance of combining innovative technology with sustainable practices, promoting reuse and reducing waste. It provides a scalable foundation for future enhancements, such as expanding product categories, introducing subscription plans, and implementing AI-driven recommendations, making it a valuable asset for the rental industry.

## Fatching data and Add To Cart

## Functionality

A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree for a project named "HINABLOGDYNAMIC". The current file, "page.tsx", is open in the editor. The code uses React hooks like useState and useEffect, along with the sanityClient library to fetch products from a database. A large preview panel on the right displays the rendered application interface, which includes a header with "Blog" and "Asgard", a sidebar with navigation links, and a main content area showing a list of products.

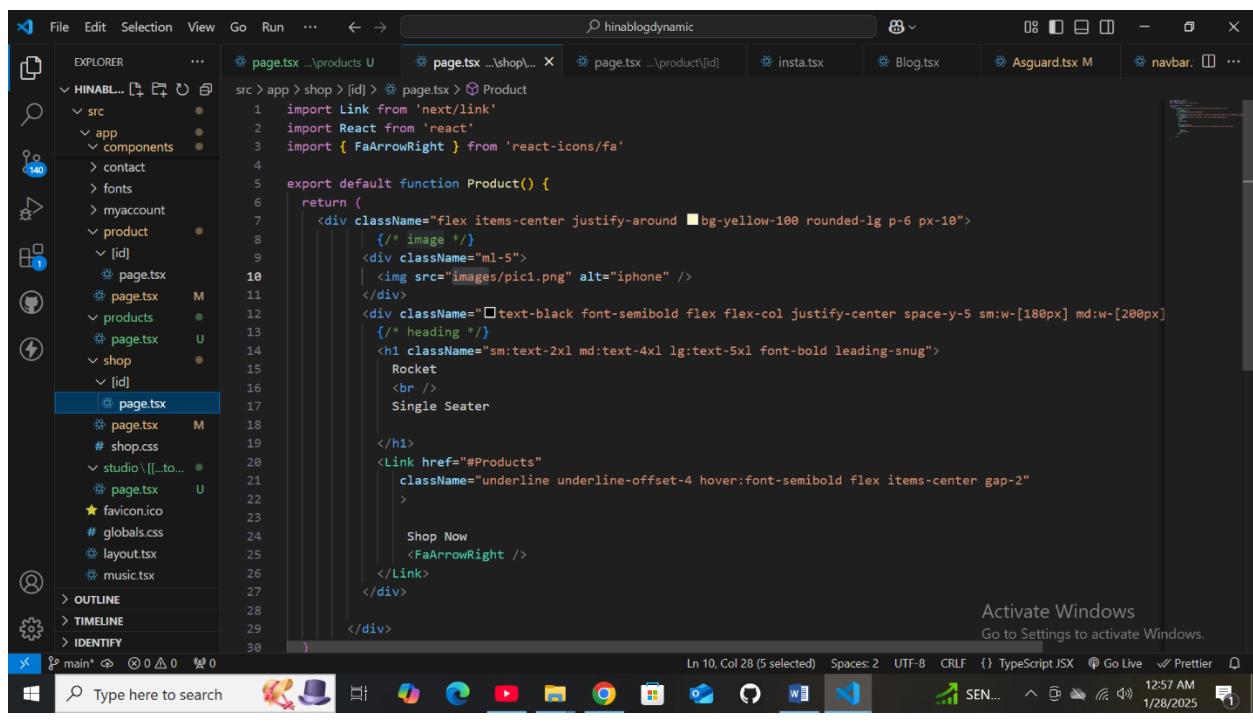
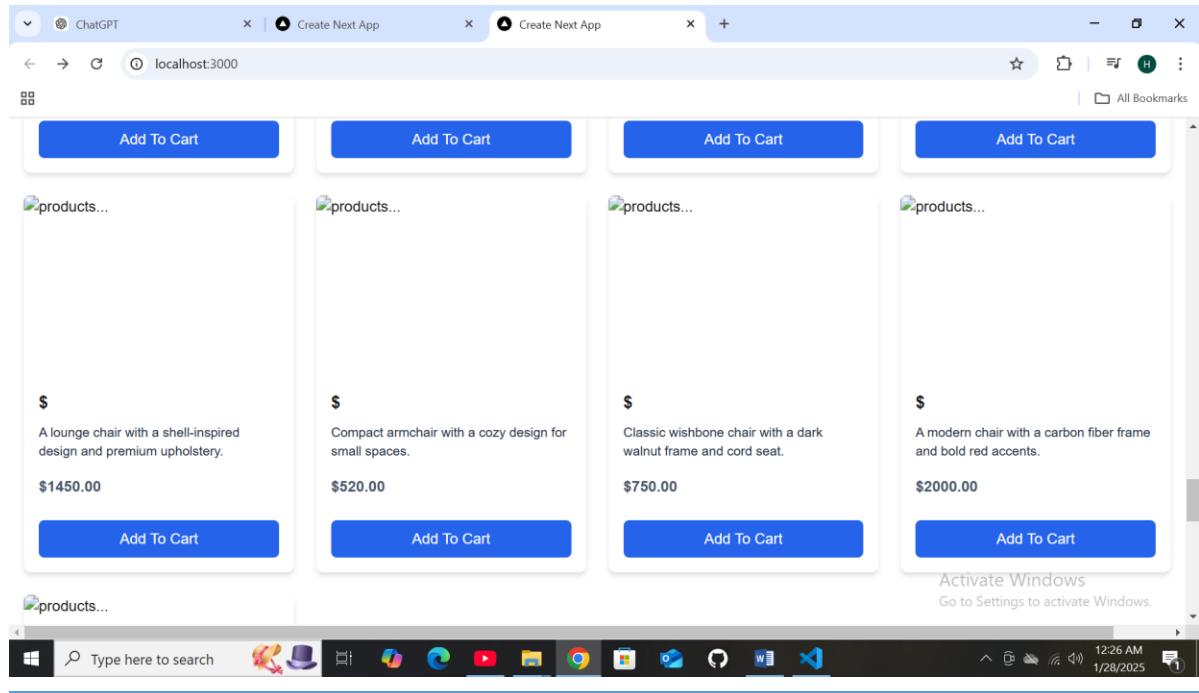
```
src > app > products > page.tsx [E] ProductCards @ products.map() callback
  1 import React, { useEffect, useState } from "react";
  2 import sanityClient from "@sanity/client";
  3 import Image from "next/image";
  4
  5 const sanity = sanityClient({
  6   projectId: "2whir9dz",
  7   dataset: "production",
  8   apiVersion: "2023-01-01",
  9   useCdn: true,
 10 });
 11
 12 interface Product {
 13   _id: string;
 14   title: string;
 15   price: number;
 16   description: string;
 17   discountpercentage: number;
 18   imageurl: string;
 19   tags: string[];
 20 }
 21
 22 const ProductCards = () => {
 23   const [products, setProducts] = useState<Product[]>([]);
 24   const [cart, setCart] = useState<Product[]>([]);
 25
 26   const fetchProducts = async () => {
 27     try {
 28       const query = `
 29         *[_type == "product"]{
 30           id,
```



# Dynamic Routing

**Product 1**

---



# “Tag Corner”

File Edit Selection View Go Run ... 🔍 Hinablogdynamic

EXPLORER HINABLOGDYNAMIC

src app products page.tsx ↗ ProductCards

```
const ProductCards = () => {
  const products = [
    {
      id: 1,
      name: "React.js Course",
      description: "Learn React.js from scratch with this comprehensive course. Includes 10+ hours of video, quizzes, and projects.",
      price: 199,
      discountPercentage: 0
    },
    {
      id: 2,
      name: "Node.js Course",
      description: "Master Node.js and build real-world applications. This course covers everything from basics to advanced topics like MongoDB and Express.js.",
      price: 149,
      discountPercentage: 10
    },
    {
      id: 3,
      name: "React Native Course",
      description: "Get started with React Native and build mobile apps for iOS and Android. This course includes hands-on projects and a guide to building a full app from scratch.",
      price: 129,
      discountPercentage: 15
    }
  ];
  return (
    <div>
      <h2>Our Products</h2>
      <p>Check out our latest products below!</p>
      <div>
        {products.map((product) => (
          <div>
            <h3>{product.name}</h3>
            <p>{product.description}</p>
            <div>
              <p>${product.price}</p>
              <small>${product.discountPercentage} OFF</small>
            </div>
            <button>Buy Now</button>
          </div>
        ))}
      </div>
    </div>
  );
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

node + 🗑️ 🔍 ... 🔍

Activate Windows  
Go to Settings to activate Windows.

LN 29, COL 7 SPACES: 2 UTF-8 CRLF ⚙️ TypeScript JSX ⚙️ Go Live ⚙️ Prettier

main\* 0 0 0 0 0 0

Type here to search

The screenshot shows a web browser window displaying a furniture store's product grid. The grid consists of four items per row. Each item has a large image, a title, a brief description, a price, a discount percentage, a category tag, and a blue 'Add To Cart' button.

Product	Description	Price	Discount	Category
Chair Wibe	A sleek outdoor chair with natural wooden elements and a modern look.	\$1200.00	10% OFF	Chair
Rapson Thirty-Nine Sofa	A mid-century modern chair with clean lines and durable materials.	\$1300.00	15% OFF	Sofa
High Quality Modern Sofa	A simple and lightweight chair ideal for dining or casual use.	\$150.00		Sofa
Alpha Table	A sturdy oak chair with a sleek and minimalist design.	\$900.00	10% OFF	Table

At the bottom right of the grid, there is a small message: "Activate Windows Add To Cart Go to Settings to activate Windows."

# Tag and fetching data.

This screenshot shows a similar product grid on the same website. The layout and individual item details are identical to the first screenshot, featuring four products per row with their respective images, titles, descriptions, prices, discounts, categories, and 'Add To Cart' buttons.

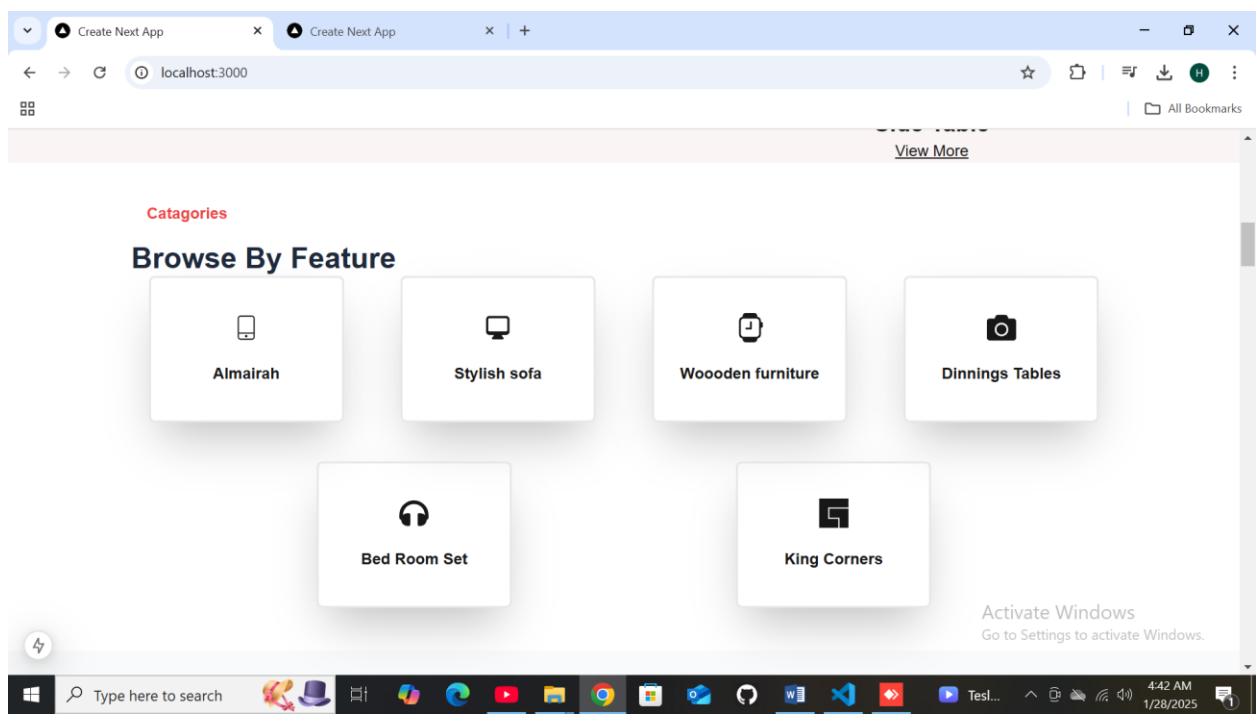
Product	Description	Price	Discount	Category
Leisure Sofa Chair Set	A comfortable set of chairs with soft cushions for relaxation.	\$1800.00	18% OFF	Sofa
Diondre Chair	A tufted chair with acrylic legs for a chic, modern touch.	\$720.00	7% OFF	Chair
Solid Bed	Durable and lightweight plastic chair for everyday use.	\$100.00	25% OFF	Bed
Red Bed	An acrylic dining chair with a sleek and minimalist Nordic design.	\$320.00	10% OFF	Bed

At the bottom right of the grid, there is a small message: "Activate Windows Add To Cart Go to Settings to activate Windows."

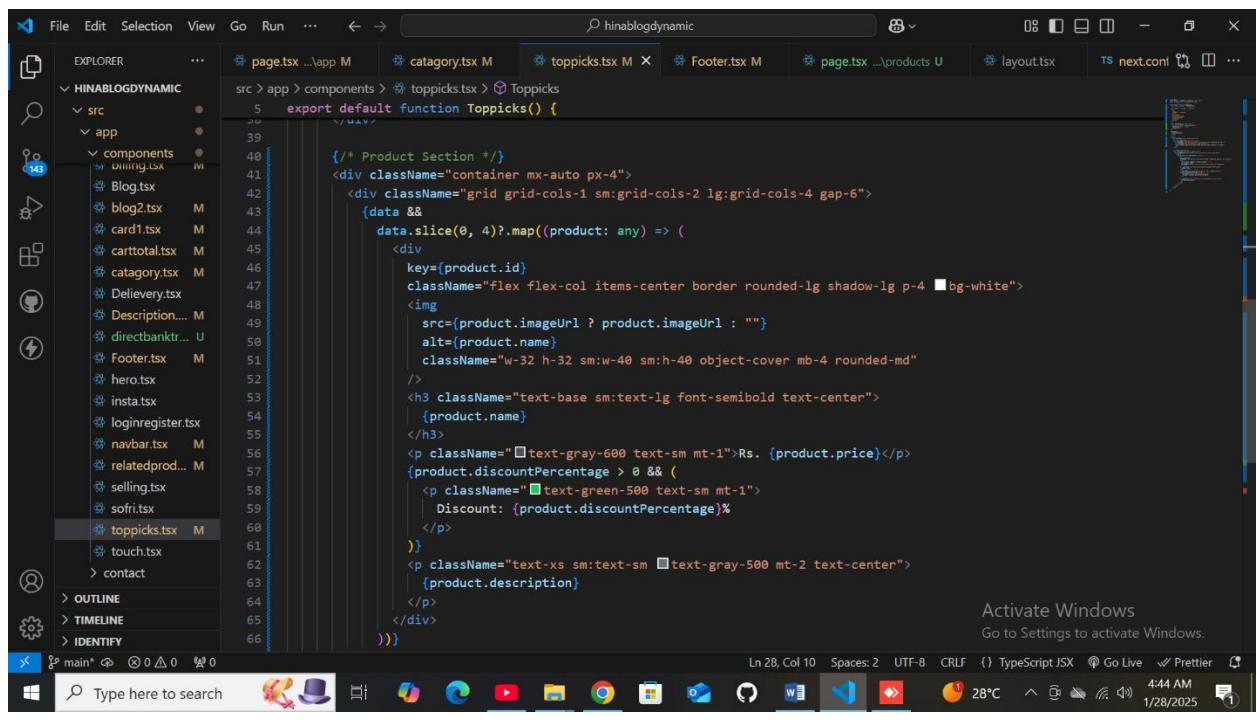
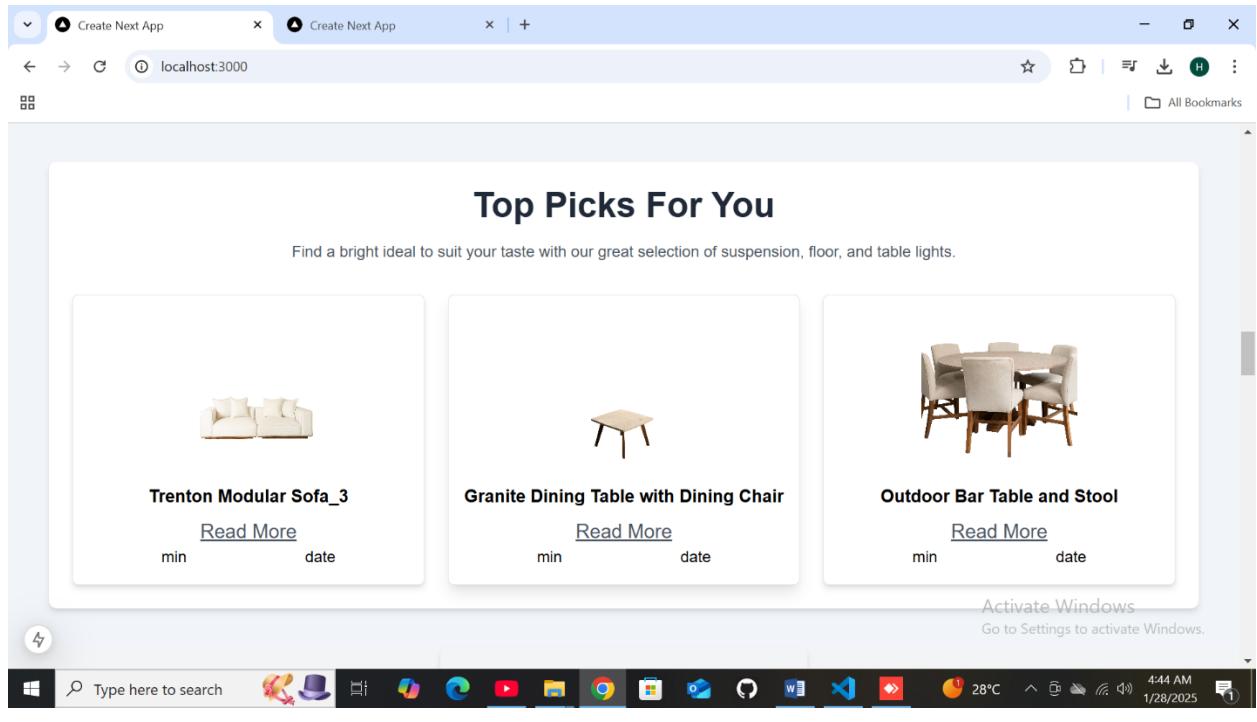
# 1. Catagories

The screenshot shows a code editor interface with the following details:

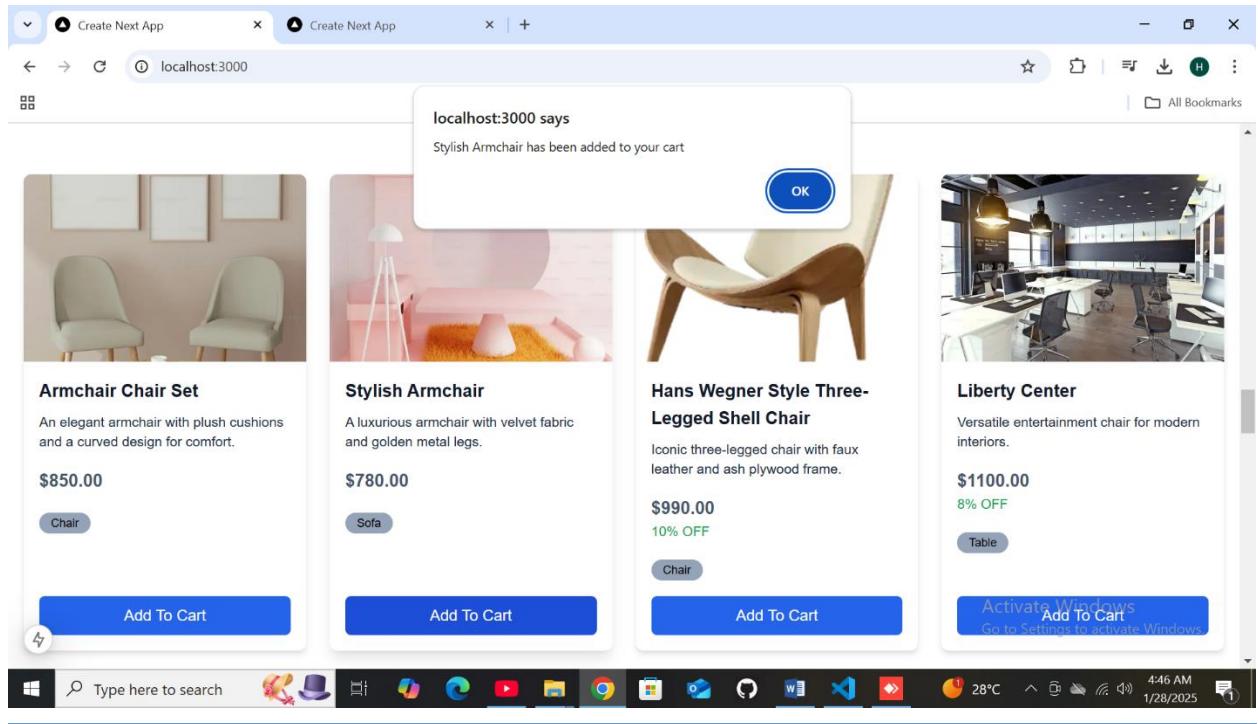
- File Explorer:** Shows the project structure under 'HINABLOGDYNAMIC'. The 'src/app/components' folder contains files like 'Billing.tsx', 'Blog.tsx', 'blog2.tsx', 'card1.tsx', 'carttotal.tsx', and 'category.tsx' (which is currently open).
- Code Editor:** The 'category.tsx' file is displayed. It contains TypeScript code for a category component, utilizing CSS-in-JS (emotion) and styled components. The component displays a grid of items with icons and names.
- Terminal:** Shows the command 'node' followed by a series of symbols.
- Status Bar:** Displays build statistics: 'Compiled in 91ms' and 'Compiled / in 144ms'. It also shows the current line (Ln 28), column (Col 63), and encoding (UTF-8). Other status indicators include 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'PORTS', 'COMMENTS', and 'TypeScript JSX'.



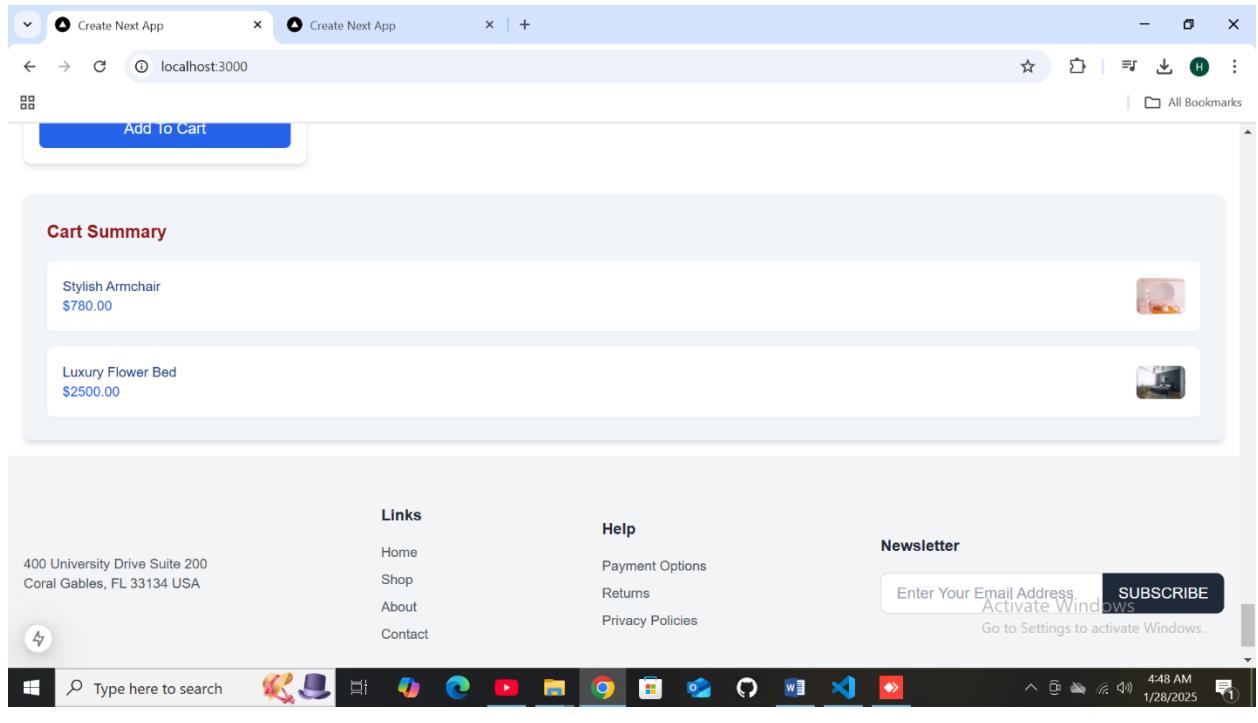
# Top pick For YOU



# Notification



# Cart Summary



```
File Edit Selection View Go Run ... < > Hinablogdynamic Explorer page.tsx ...app M catagy.tsx M toppicks.tsx M Footer.tsx M page.tsx ...products U layout.tsx ts next.conf ...  
HINABLE... app M  
src  
app  
components hero.tsx insta.tsx loginregister.tsx navbar.tsx relatedprod... M selling.tsx sofri.tsx toppicks.tsx M touch.tsx  
> contact  
> fonts  
> myaccount  
> product  
> products  
> shop  
> studio  
★ favicon.ico  
# globals.css  
& layout.tsx  
> OUTLINE  
> TIMELINE  
> IDENTIFY  
main.tsx 0 0 0 0 Type here to search 448 AM 1/28/2025  
src > app > products > page.tsx > [e]ProductCards  
29 const ProductCards = () => {  
75     {products?.map((product: Product) => (  
116         <div>  
117             <button onClick={() => addToCart(product)}>  
118                 Add To Cart  
119             </button>  
120         </div>  
121     ))}  
122 </div>  
123  
124 /* Cart Summary */  
125 <div className="mt-8 bg-slate-100 p-6 rounded-lg shadow-md">  
126     <h2 className="text-lg font-bold text-red-800 mb-4">Cart Summary</h2>  
127     {cart.length > 0 ? (  
128         <ul className="space-y-4">  
129             {cart.map((item, index) => (  
130                 <li key={index} className="flex justify-between items-center bg-white shadow-sm p-4 rounded-md">  
131                     <div>  
132                         <p className="text-sm text-blue-900 font-medium">{item.name}</p>  
133                         <p className="text-sm text-blue-600">| ${item.price.toFixed(2)}</p>  
134                     </div>  
135                     <Image src={(urlFor(item.image).url() || "/placeholder.png") alt={item.name}}>  
136                 </li>  
137             ))  
138         </ul>  
139     ) : (  
140         <div>  
141             <p>Your cart is empty!</p>  
142         </div>  
143     )  
144 </div>  
145  
146 Ln 29, Col 7 Spaces: 2 UTF-8 CRLF () TypeScript JSX ⚡ Go Live ⚡ Prettier
```

Billing Details

Checkout;

The screenshot shows a web browser window with two tabs: 'Create Next App' and 'localhost:3000/checkout'. The main content area displays a 'Billing Details' form with fields for Firstname (hina), Last name (last name), Country/ Region (Country), Street Address (Street), and Town/City (Town). To the right, there is a sidebar titled 'Direct Bank Transfer' containing payment instructions and a checkbox for 'Direct Bank Transfer'. Below the sidebar, there is a note about activating Windows and a link to 'Activate Windows'. The bottom of the screen shows a taskbar with various icons.

The screenshot shows a code editor with the 'billing.tsx' file open. The code defines a function named 'billing' that returns a JSX component. The component includes a title 'Billing Details', several input fields for personal information (Firstname, Last name, Country, Street, Town/City, Province, Zip Code, Postal Address, E-mail Address, Reference No, and Ref), and a note about activating Windows. The code editor interface includes an Explorer sidebar showing project files like 'page.tsx', 'category.tsx', 'toppicks.tsx', 'Footer.tsx', and 'page.tsx'. The bottom of the screen shows a taskbar with various icons.

```

export default function billing() {
  return (
    <div>
      <h1 className='font-bold text-4xl text-center mb-9'>Billing Details</h1>
      <div className='space-y-3'>
        <h1>Firstname</h1>
        <input type="text" placeholder='name' className='outline rounded-sm px-3 py-2' />
        <h1>Last name</h1>
        <input type="text" className='outline rounded-sm px-3 py-2' placeholder='last name' />
        <h1>Country/ Region</h1>
        <input type="text" placeholder='Country' className='outline rounded-sm px-3 py-2' />
        <h1>Street Address</h1>
        <input type="text" className='border-spacing-2 outline rounded-sm px-3 py-2' placeholder='Street' />
        <h1>Town/City</h1>
        <input type="text" placeholder='Town' className='outline rounded-sm px-3 py-2' />
        <h1>Province</h1>
        <input type="text" className='border-spacing-2 outline rounded-sm px-3 py-2' placeholder='province' />
        <h1>Zip Code </h1>
        <input type="text" className='border-spacing-2 outline rounded-sm px-3 py-2' placeholder='Zip' />
        <h1>Postal Address</h1>
        <input type="text" placeholder='name' className='outline rounded-sm px-3 py-2' />
        <h1>E-mail Address</h1>
        <input type="text" className='border-spacing-2 outline rounded-sm px-3 py-2' placeholder='E-mail' />
        <h1>Reference No</h1>
        <input type="text" placeholder='Ref' className='outline rounded-sm px-3 py-2' />
        <br />
        <input type="text" className='border-spacing-2 outline rounded-sm px-3 py-2' placeholder='Activation Windows inf' />
      </div>
    </div>
  )
}

```

Sanity...

A screenshot of a web browser window titled "Product | Default". The address bar shows "localhost:3000/studio/structure/product". The page has a yellow header with "Home", "Shop", "Product", and "Contact" links. A search bar and a navigation menu are also present. The main content area displays a "Content" sidebar with a "Product" section. Under "Product", there is a list of items: "Luxury Flower Bed", "Blue Bed", "Red Bed", "White Bed", "Solid Bed", "Matilda Velvet Bed", and "Diondre Chair". A "What's new" box at the bottom left mentions "Sanity Create Content Mapping, Visual Editing, and Content Releases". The Windows taskbar at the bottom shows various pinned icons.

A screenshot of a web browser window titled "Luxury Flower Bed | Default". The address bar shows "localhost:3000/studio/structure/product/gbTY3B2TxF1zfilOOutu1m". The interface is similar to the previous screenshot, with a yellow header and a "Content" sidebar. The "Product" section is expanded, showing a detailed view for "Luxury Flower Bed". The details include:

- Discount Percentage:** 0
- Is Featured Product:** (checkbox checked)
- Stock Level:** 2

A "What's new" box at the bottom left is identical to the one in the first screenshot. The Windows taskbar at the bottom is visible.

```
// Querying with "sanityFetch" will keep content automatically updated
// Before using it, import and render "<SanityLive />" in your layout, see
// https://github.com/sanity-io/next-sanity#live-content-api for more information.
import { defineLive } from "next-sanity";
import { client } from './client'

export const { sanityFetch, SanityLive } = defineLive({
  client: client.withConfig({
    // Live content is currently only available on the experimental API
    // https://www.sanity.io/docs/api-versioning
    apiVersion: 'vX'
  })
});
```

- Component buildup compilation.
- Tailwind css.
- Responsive full.
- UseEffect, useState utilization.

- Insha Allah
- **Future Progress**
- This furniture rental e-commerce website is designed with SEO optimization in mind, ensuring great visibility and a smooth user experience. The platform will continue to improve and evolve, with ongoing upgrades to enhance performance, expand features, and meet the changing needs of users.
- As time progresses, the website will be upgraded with new technologies, expanded inventory, better personalization, and even more user-friendly features to stay ahead in the market and provide the best possible service.

# THE END