HACKATHON 3

DAY 1 : Laying the Foundation for Your Marketplace Journey

- **Marketplace Type:**

  My marketplace type is General E-commerce.
  Purpose: my e-commerce marketplace allows customers to virtually view products, enabling anyone to browse and purchase them online.

- **Business Goals:**

  People may have limited furniture stores in their area, forcing them to travel far. We aim to provide them with the convince of shopping from home.
  Due to limited furniture stores, customers may have restricted options. We will provide wide range of varieties.
  We are offer customization options to cater to their specific needs.

- **Target Audience:**

  25+ year peoples
  Event Organizers, Wedding planners and providing catering services.

- **Products:**

  I will be offering furniture products like Sofas, Chairs, and Stools.

- **Marketplace Apart:**

 I wil offer my customers a delivery time of 3-5 days.
  I will provide High-quality products at affordable prices.

DAY 2 : Planning The Techincal Foundation

- ○ **Frontend Workflow:**

  UI & UX Design: Next JS (Design Structure), Tailwind CSS(Designing and Styling UI), Figma (Design UI & UX), React Hooks(State Management).

  Pages: Home, About, Contact, Category, Product, Product Detail, Cart, Checkout, FAQs.
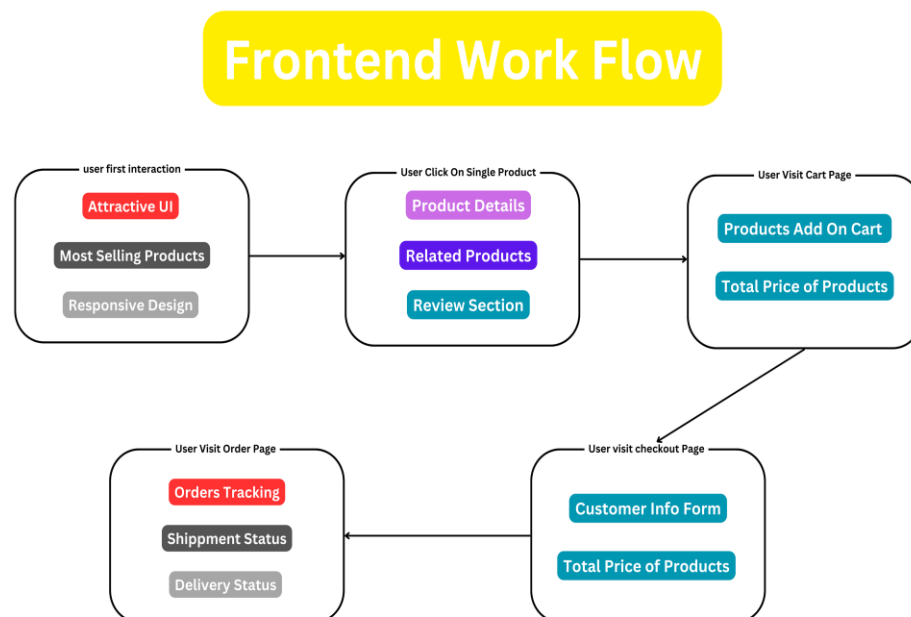
- ○ **Sanity CMS:**

  1. **Product Schema** (Manage Product Data)
  2. **Category Schema** (Manage Category Data)
  3. **Customer Schema** (Manage Customer Data)
  4. **Order Schema** (Manage Order Data to the Customer)
  5. **Contact Form Schema** (Manage Contact Info Data)

o **Third-Party APIs:**

1. **Payment** :
   Stripe API Payment secure and trusted
2. **Shippment** :
   Shipengine API for tracking and order placement
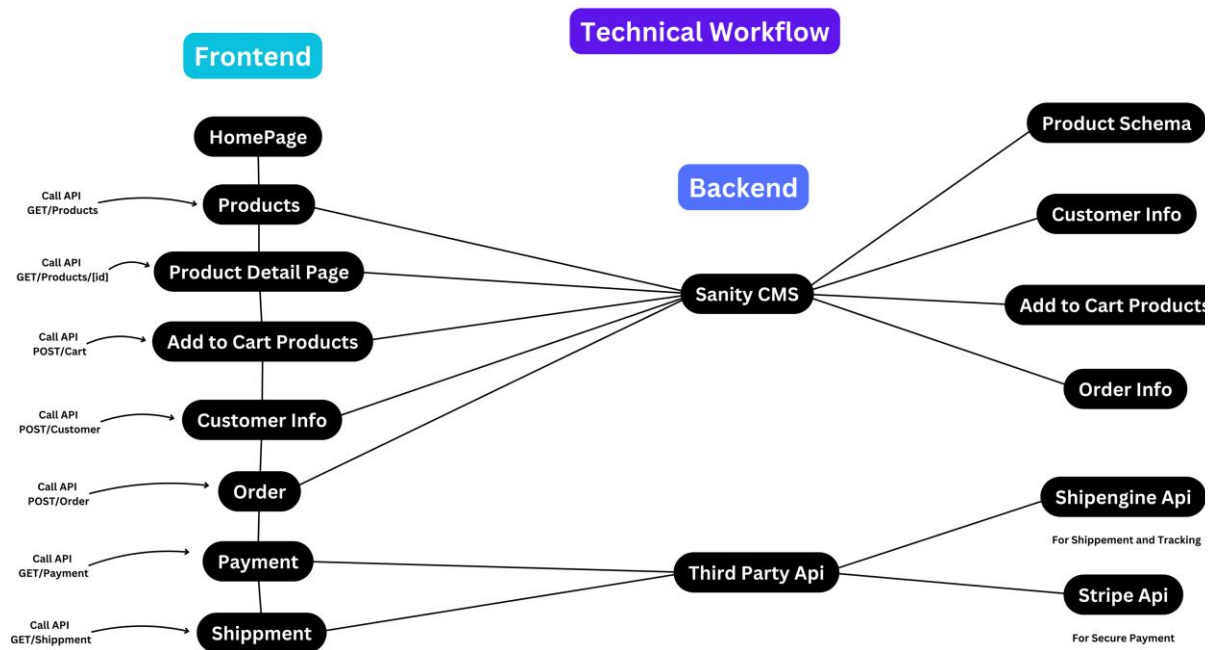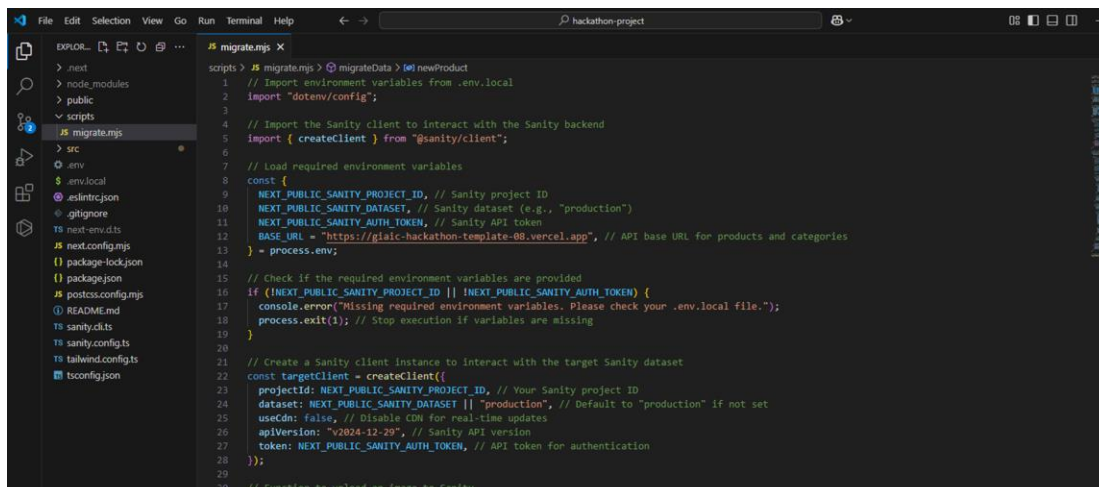
DAY 3 : API Integration & Data Migration

○ **Migration File:**

In this migration file I will upload data API to Sanity CMS. Add secure token key to send data to sanity.

○ **Add Data In Sanity (Product Data):**
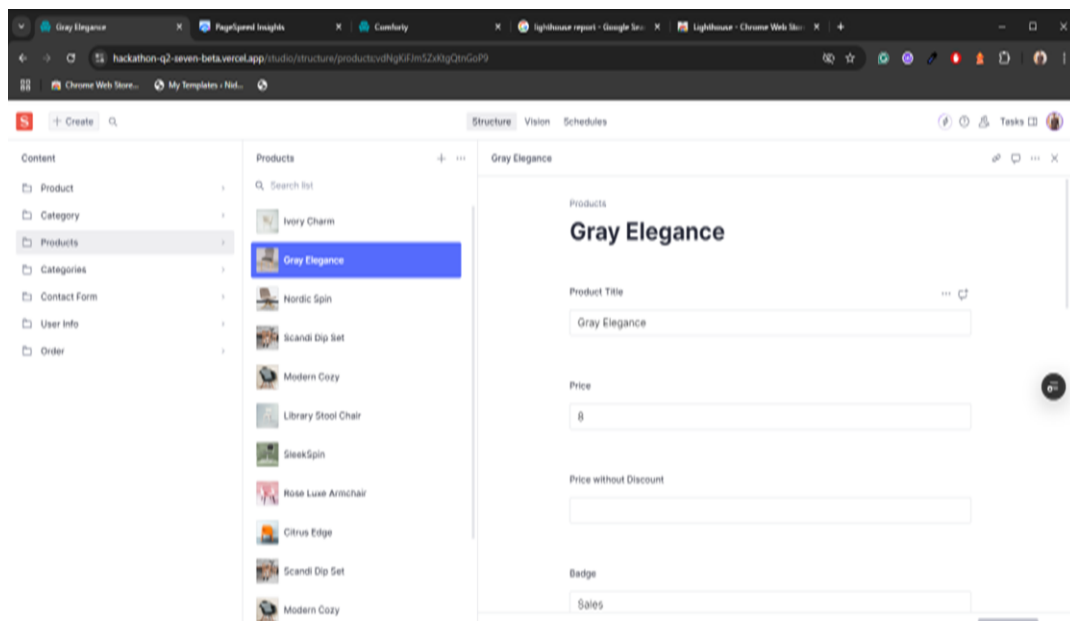
○ **Add Data In Sanity (Categories Data):**
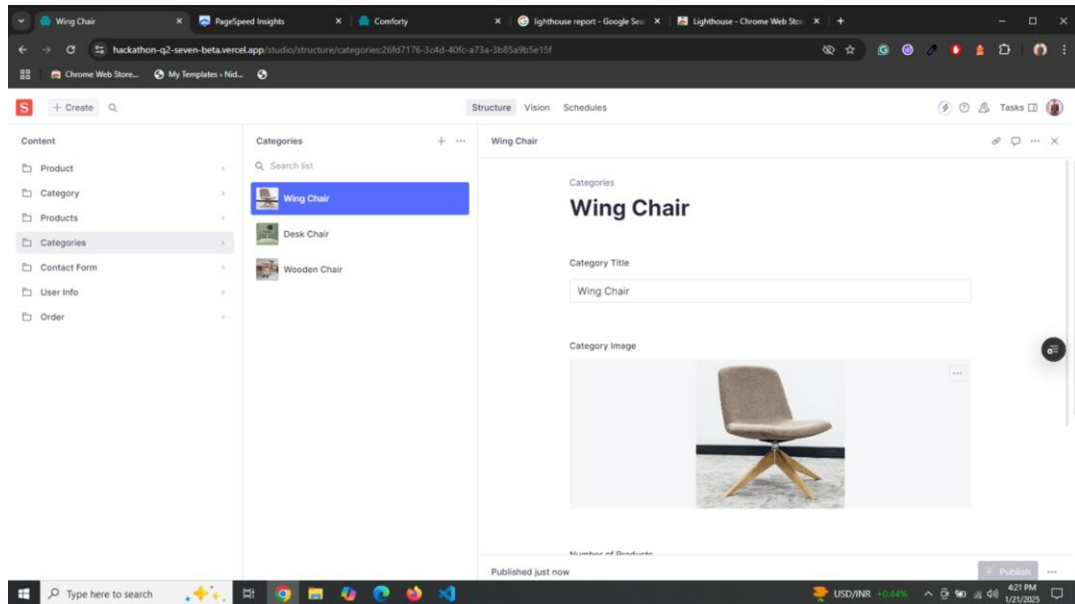
○ **Add Data In Frontend (Product Data):**
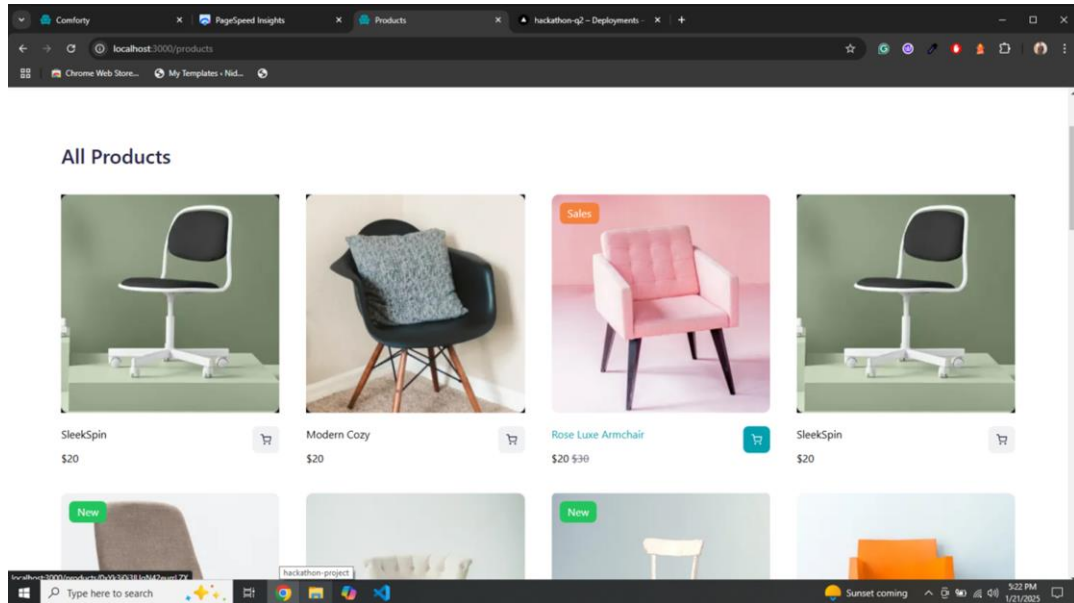
○ **Add Data In Frontend (Product Data):**
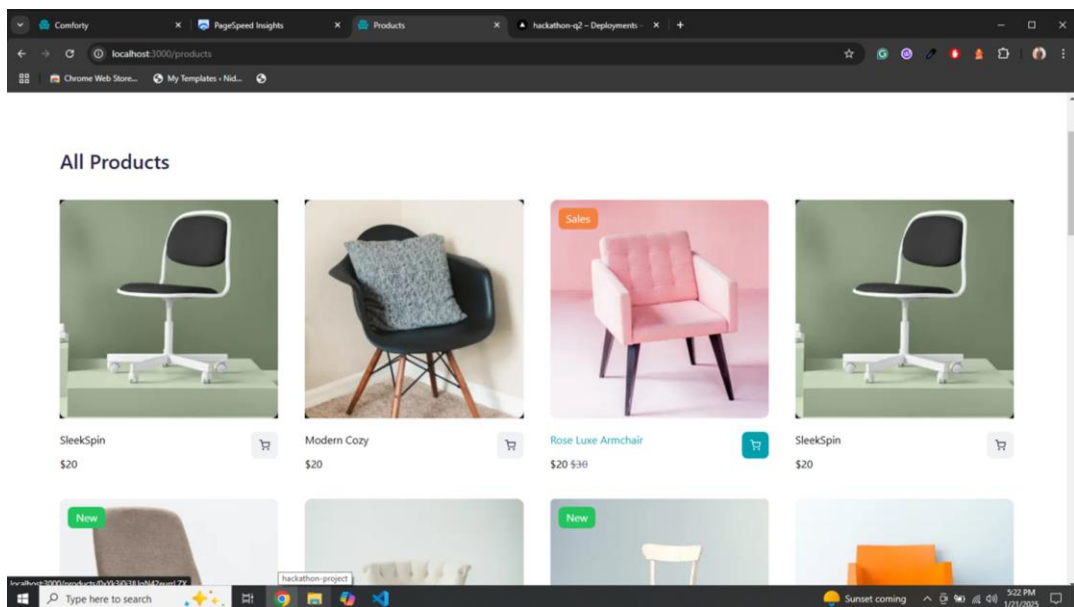
DAY 4 : Building Dynamic Frontend Components
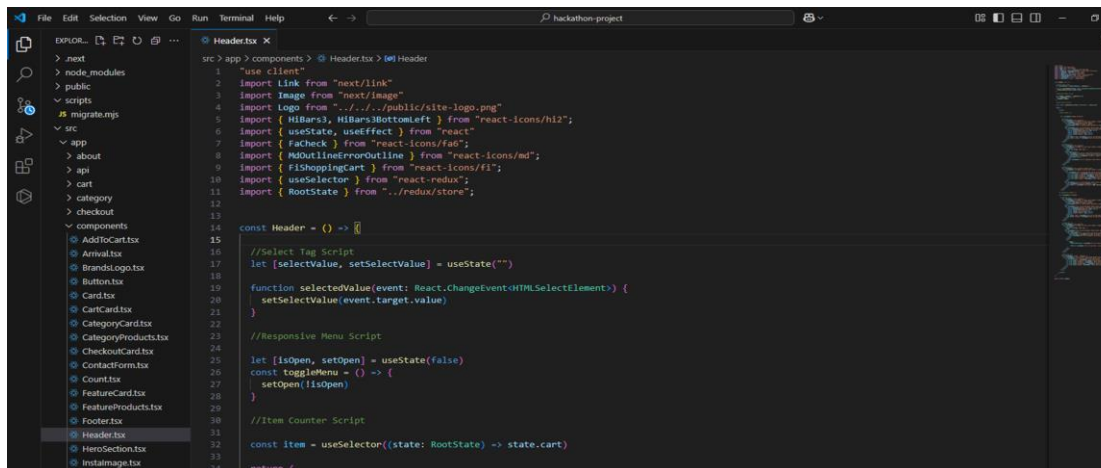
o **Overview:**

Make a component for using reusable content.

Components: Header, Footer, Product Card, Category Card, Product Details, Cart Card, Checkout Card.
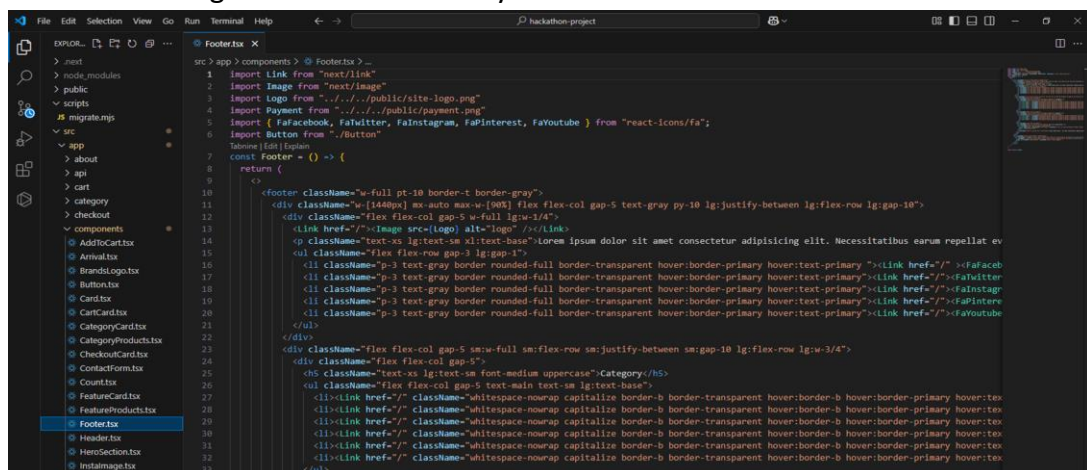
o **Header Component:**

Render in all Pages Header is necessary.

o **Footer Component:**

Render in all Pages Footer is necessary.

o **Product Card Component:**

In this component is reusable for showing products.
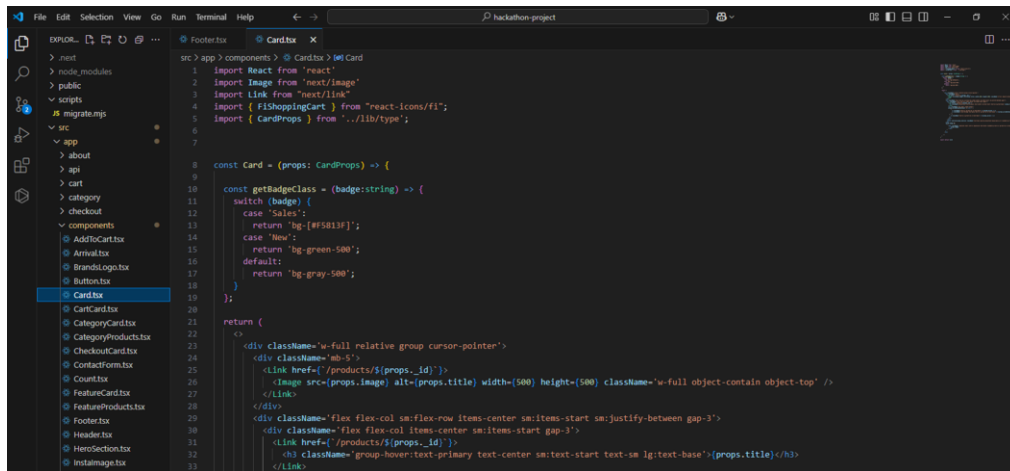
o **Product Detail Component:**

In this component is usable for showing products detail page.
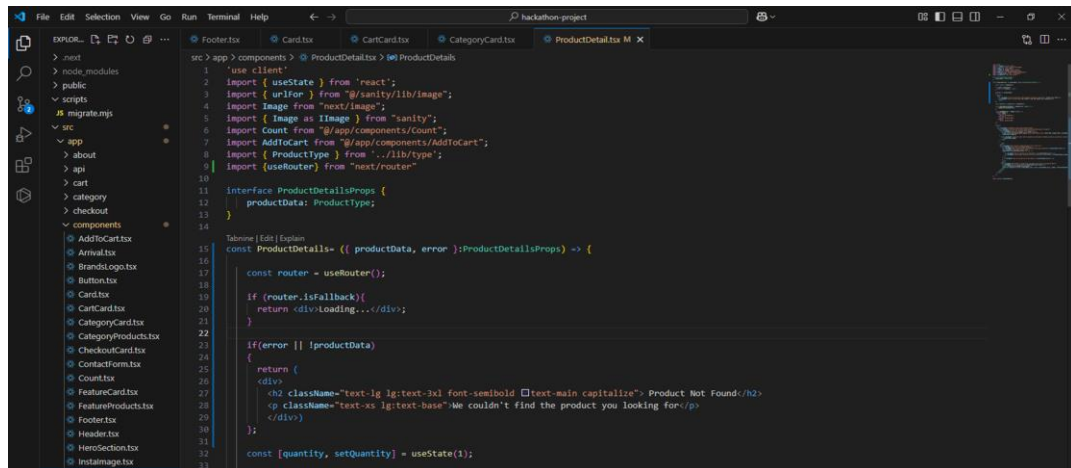
o **Category Card Component:**

In this component is usable for showing.
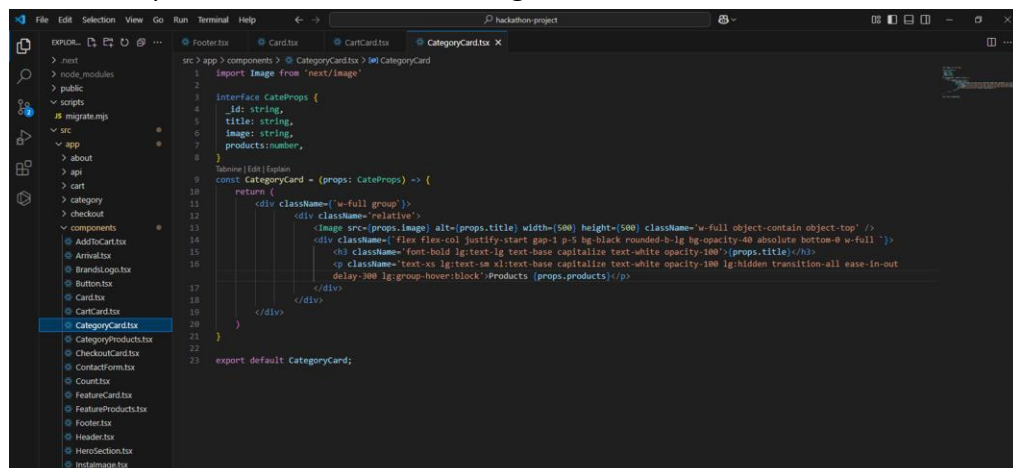
o **Cart Card Component:**

In this component is usable for list of cart items.
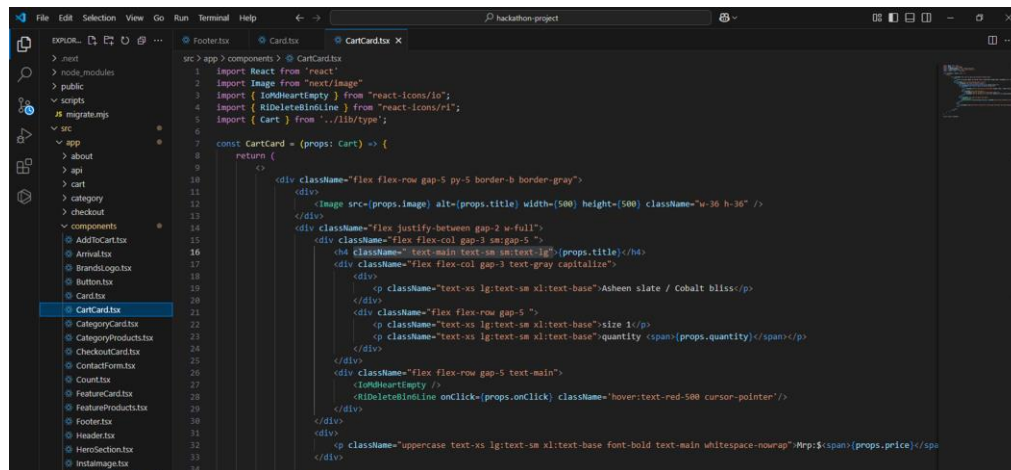
o **Checkout Card Component:**

In this component is usable for add cart items and payment Method ensure
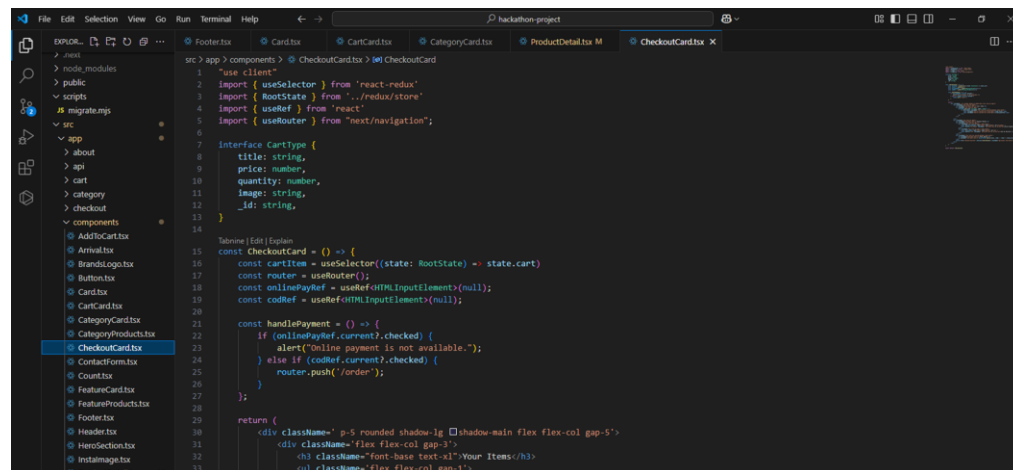
o **Products Data on Page:**

Source code and output of this code
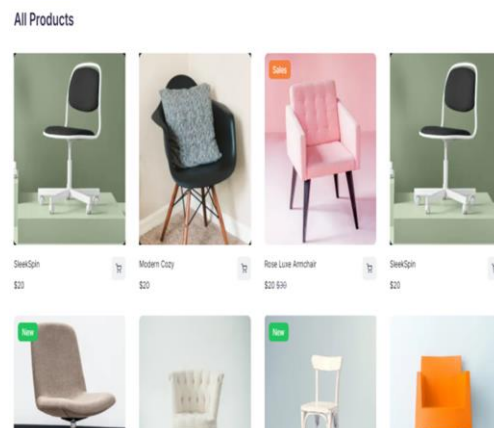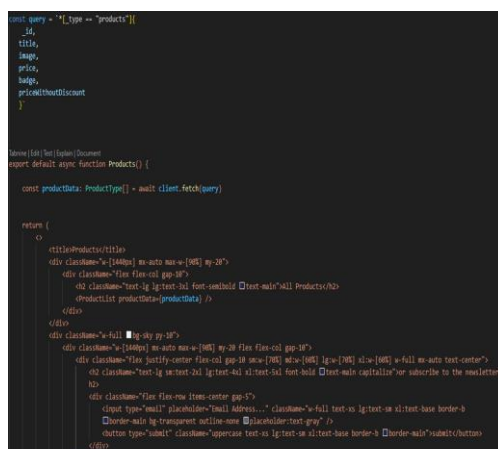
o **Products Detail Page:**

Source code and output of this code and Dynamic Routing
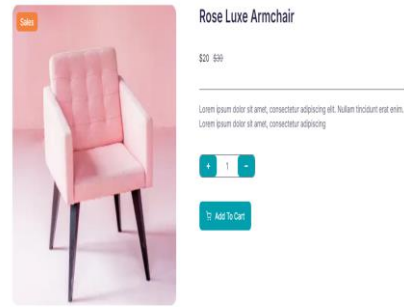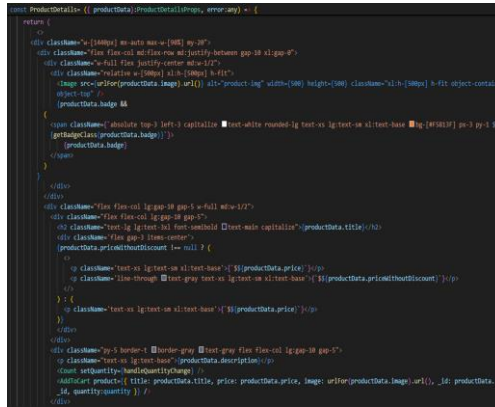
o **Cart Page:**

Source code and output of this code
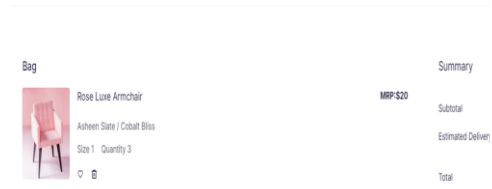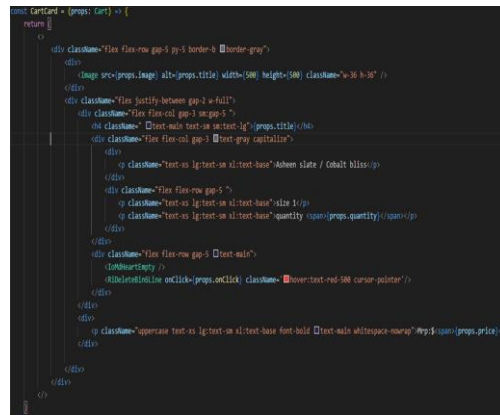
o **Checkout Page:**

Source code and output of this code
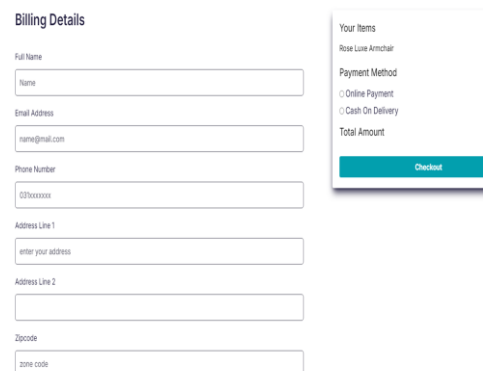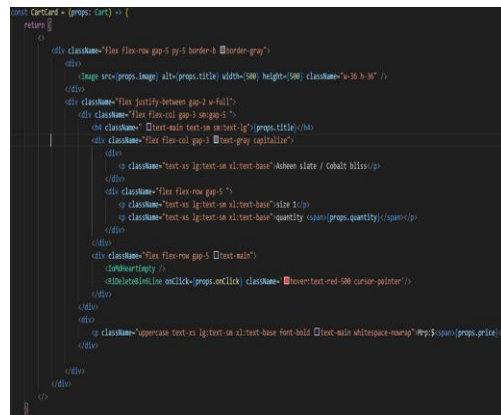
DAY 5 : TESTING, ERROR HANDLING, AND BACKEND INTEGRATION

○ **API Request & Response:**

API Request : (https://giaic-hackathon-template-08.vercel.app/api/products**)**

API Response :

API Request : (https://giaic-hackathon-template-08.vercel.app/api/categories**)**
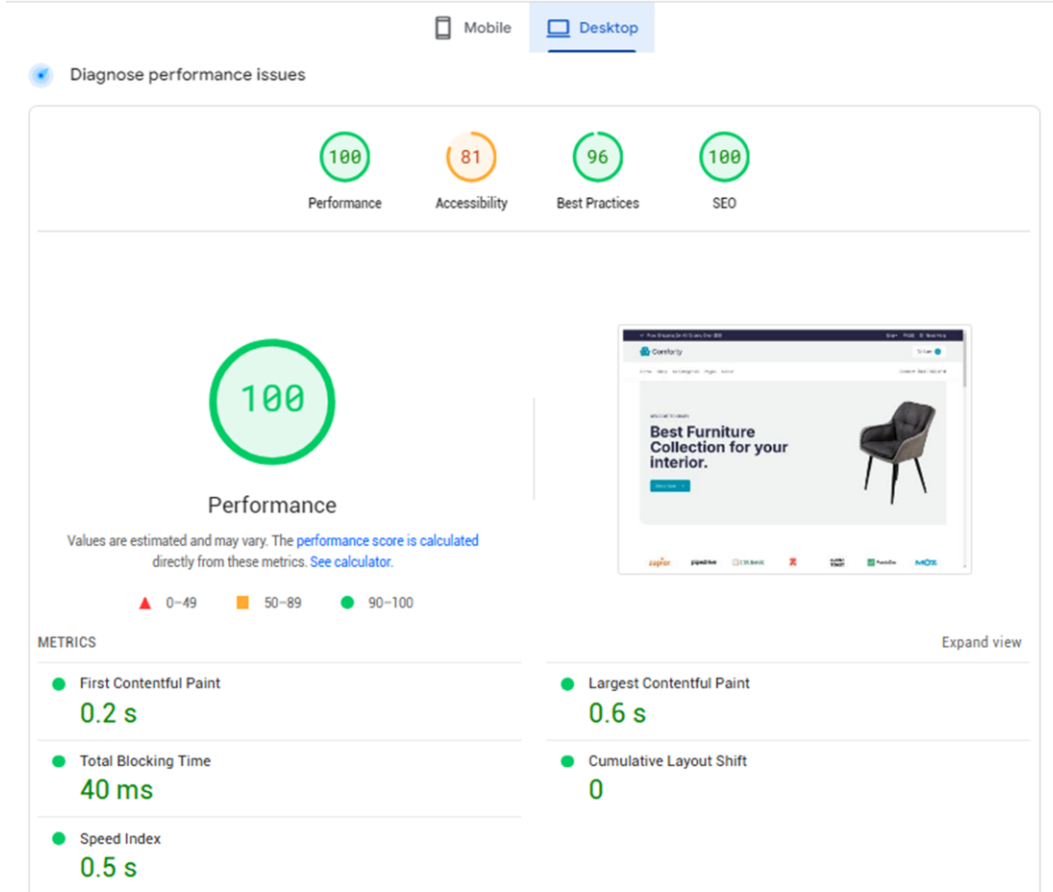
API Response :

o Light House Website Performance:

Using chrome extension to check website performance.

Mobile Performance:
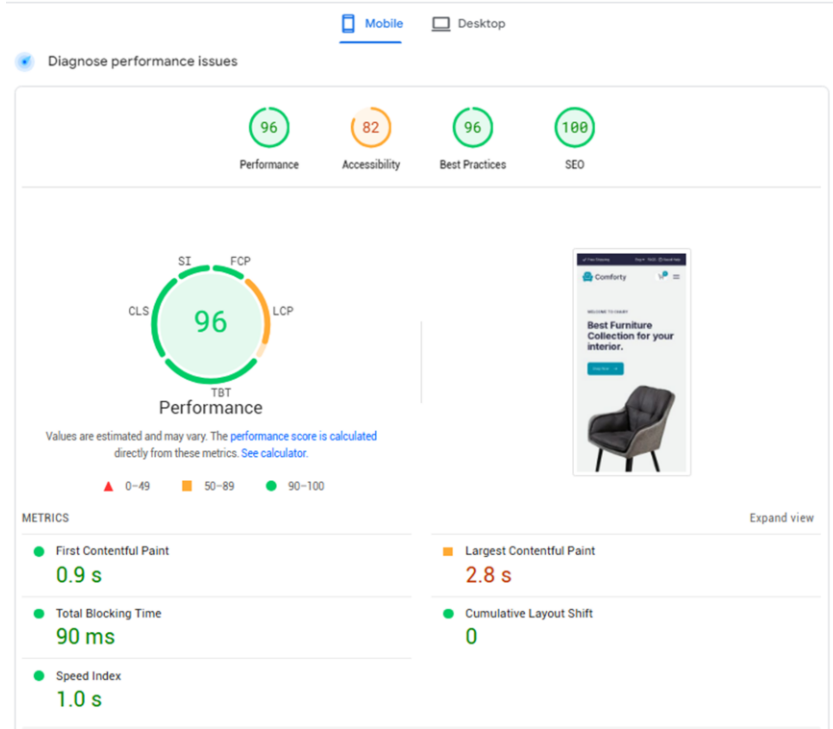
Desktop Performance:

Performance Monitor Report:
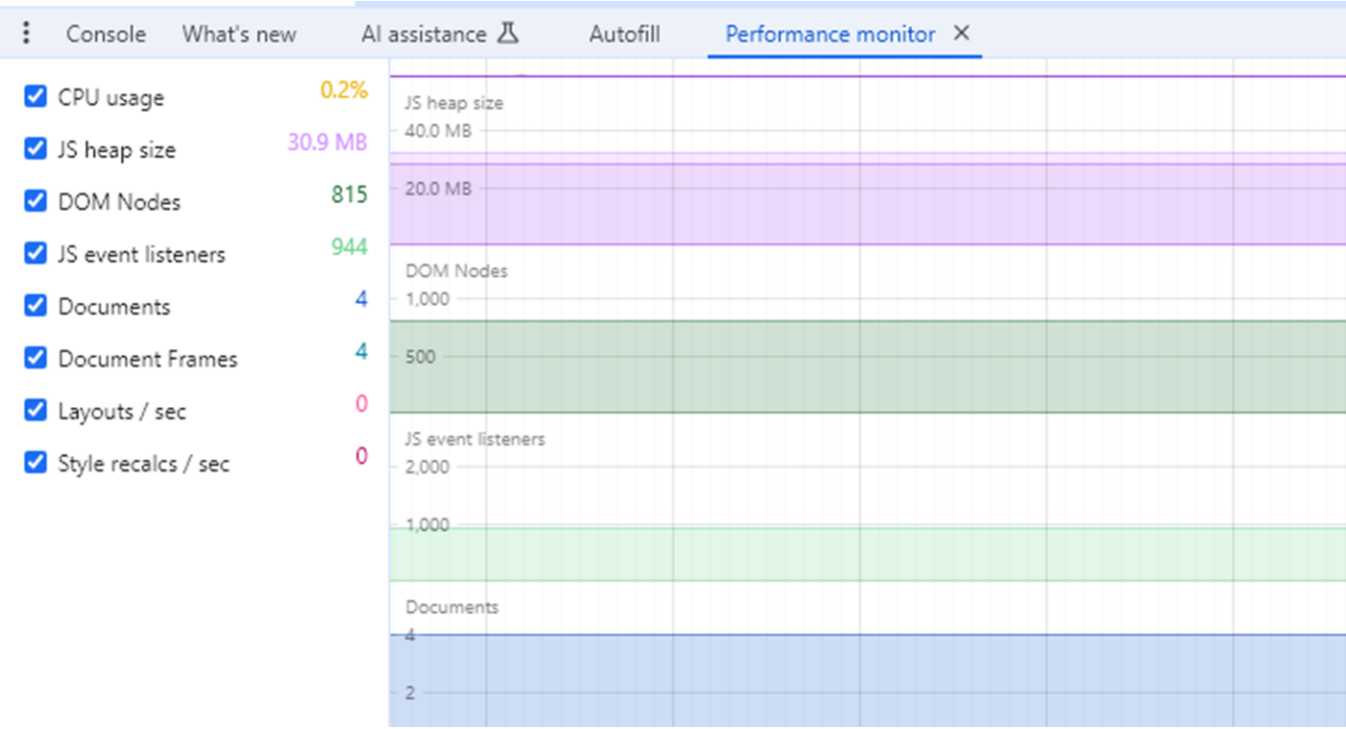
Test Case Report:

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	High	-	Works as expected
TC002	Validate product detail Page	Open product detail page > Verify Single product detail page	Product details displayed correctly in Product Detail Page	Product details displayed correctly in Product Detail Page	Passed	High	-	Works as expected
TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	Works as expected
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Layout adjusts properly to screen size	Passed	Medium	-	Test successful
TC005	Cross Browser Compatibility	Check multiple browser windows > Check layout	Layout adjusts properly to different browser	Layout adjusts properly to different browser	Passed	Medium		Test successful
TC006	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully

DAY 6 : Deployment & Staging Environment Setup

- o **Hosting Platform:**
  Deploy in **Vercel** because **Vercel** is compatible with Next JS projects files and config

- o **Setup Environment Variables:**
  Go to Deploy Project > Go to setting > Go to Environment
  Variables > set Manually or upload .env file