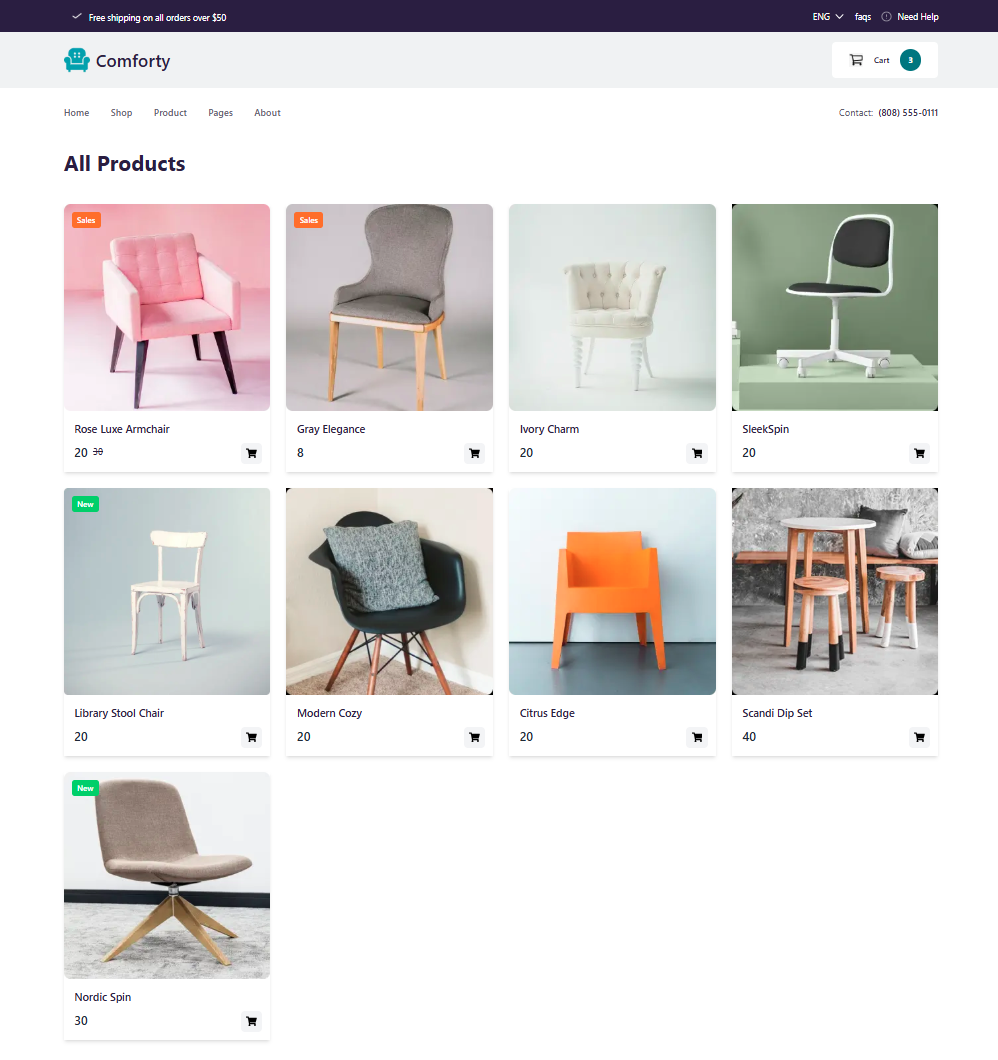
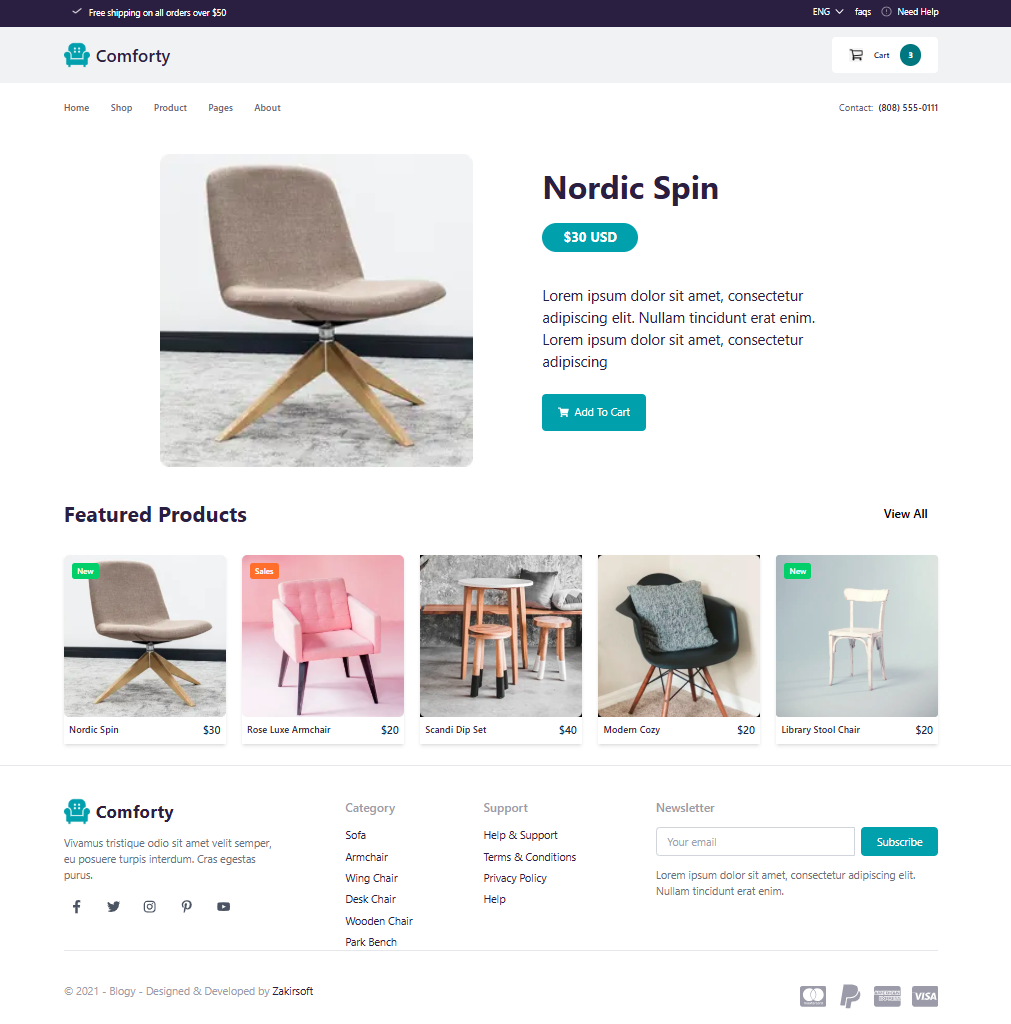
**BUILDING DYNAMIC FRONTEND**

**COMPONENTS**

1. **Product Listing page:**

* **Functionality:** I have four components which show different number of products, clicking on any of them leads to product detail page, where product can be added to cart.
* **Screenshots:**

****

****

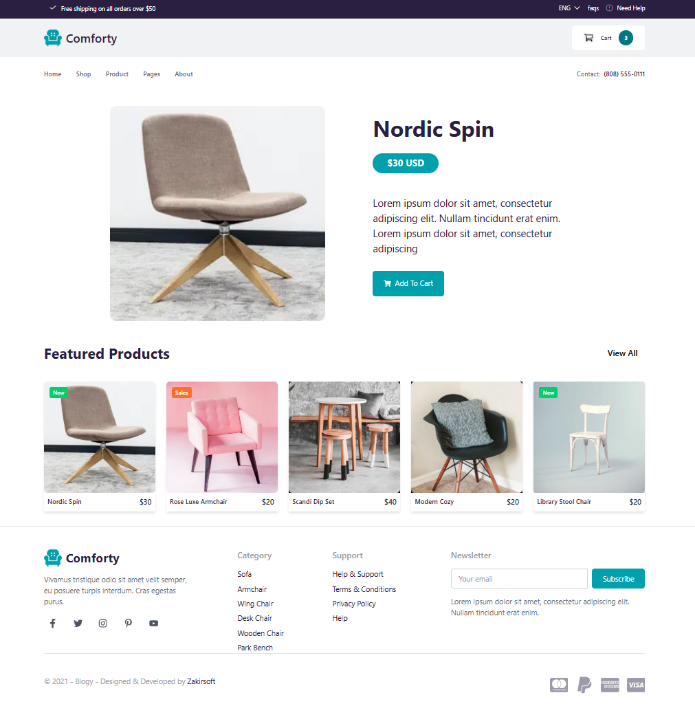
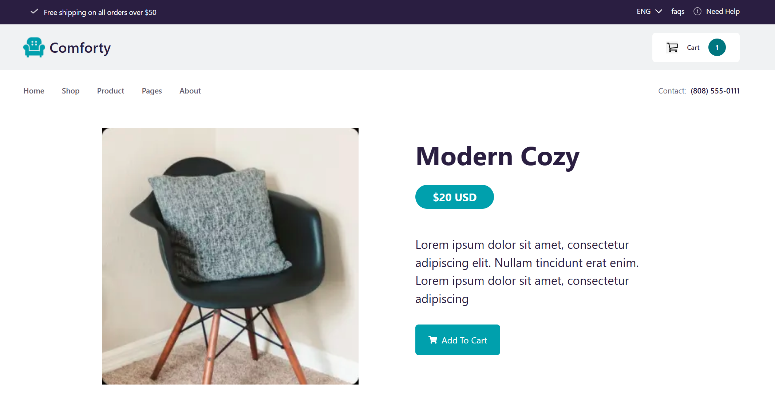
1. **Dynamic product data:**

* **Functionality and steps:**

I create a file in app folder app/pages/product-page/[id]/page.tsx , which contains the code to open Product detail page of any components individually.

* Fetch data from sanity
* Try and catch condition in case for error testing
* Design components to display product details.
* Add to the cart button to add product in cart and display successful message.
* **Screenshots:**

****

****

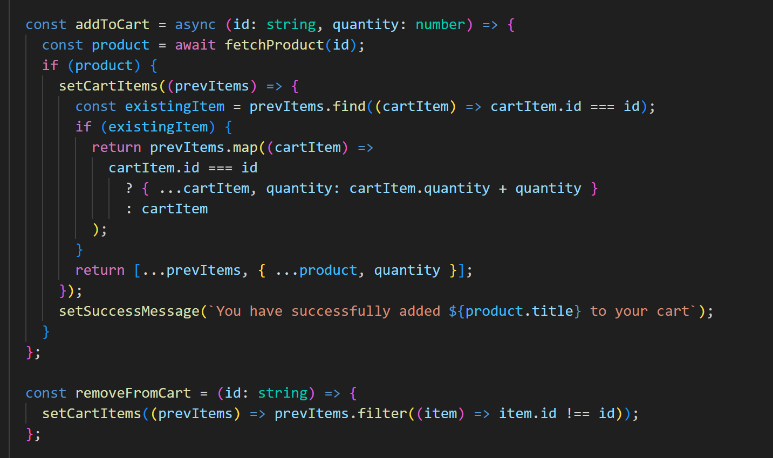
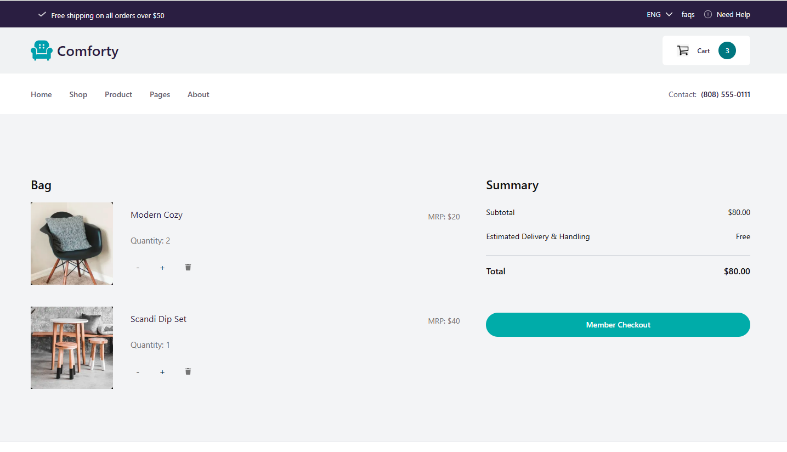
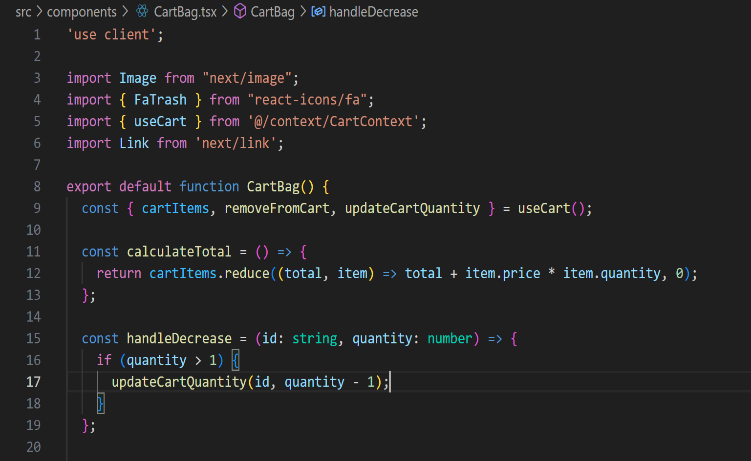
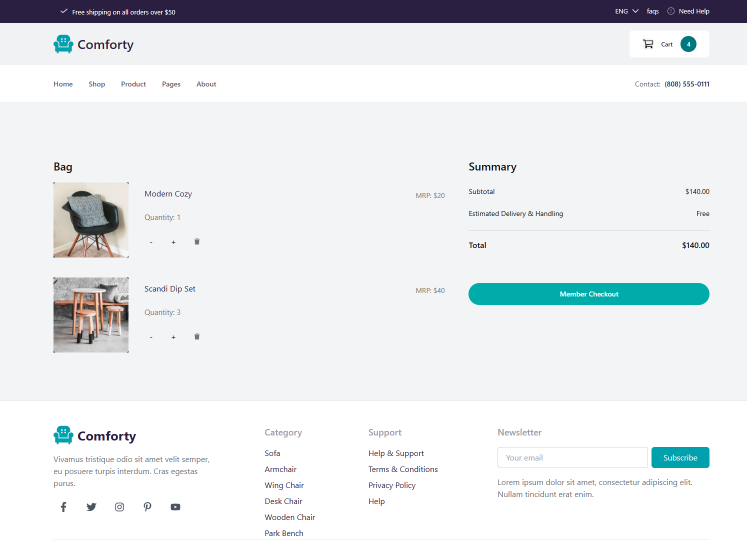
1. **Cart page:**

* **Functionality and Steps:**

When user click on Add to cart button, the product is added to cart immediately, and cart icon on Header shows the number of products in cart, by clicking on that icon user go to cart page, where all products images, names, quantity and prices are shown, user can change quantity and remove products form cart.

Cart page also shows cart summary which includes total amount and a button to go to checkout page.

* **Screenshots:**

**  **

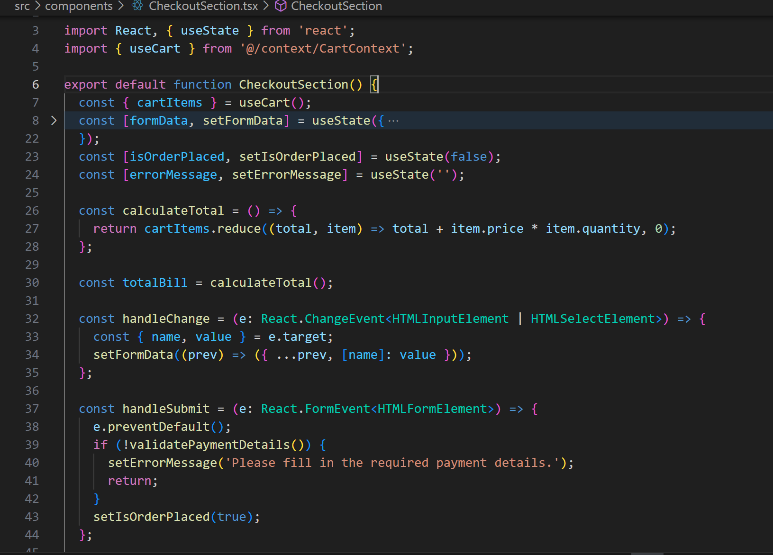
1. **Checkout Page:**

* **Functionality and Steps:**

Users must fill a Form to complete the order process, select payment method, check total amount, after submitting a message shows that order is successfully placed. This user data and customer data is stored in a sanity schema, which can be access through admin Dashboard by Admin.

* **Screenshots:**

A screenshot of a login form

Description automatically generated

**Checklist for Day 4:**

**Self-Validation Checklist:**

**Frontend Component Development:**

**✔ ✘**

**Styling and Responsiveness:**

**✔ ✘**

**Code Quality:**

**✔ ✘**

**Documentation and Submission:**

**✔ ✘**

**Final Review:**

**✔ ✘**