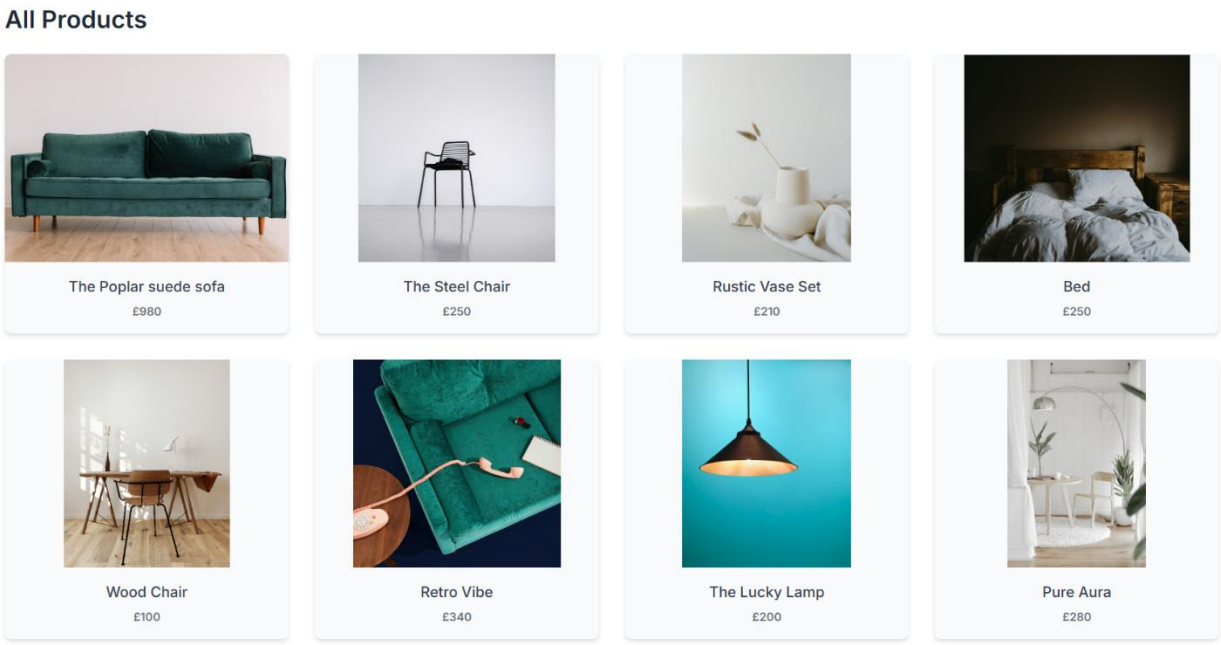
DAY 4 - DYNAMIC FRONTEND COMPONENTS – (FURNITURE-ECOMMERCE)

Here is a proper description on what I have done in my day 4 task of hackathon.

Key Components built:

1. **Product Listing Component:**

2. **Dynamic Grid Layout**: Implemented a dynamic product listing component that renders product data in a grid layout.
3. **Displayed Fields:**
    a. **Product Name**: Displays the name of the product for users.
    b. **Price**: Shows the price of each product.
    c. **Image**: Includes a product image to effectively showcase the collection in an organized manner.



Created By: Shoaib Munir

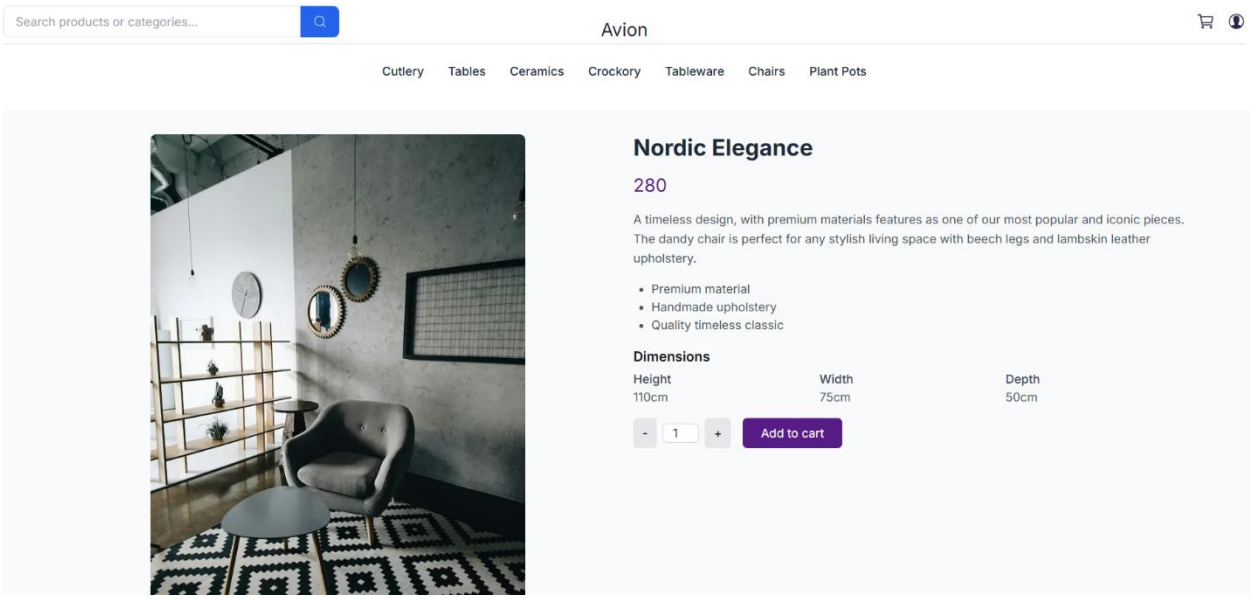
4. **Product Detail Component:**

   a. Each product page provides detailed information about a specific product. The details include:

      - **Dynamic Data Fetching:** Fetched product data from Sanity CMS using its API.
      - **Unique Identifier:** Utilized the product's slug as a unique identifier to dynamically display the correct details on the corresponding page.
      - **Displayed Fields:**

        1. **Product Description:** Detailed information explaining the product's features and dimensions.
        2. **Price:** The product's price for purchasing consideration.

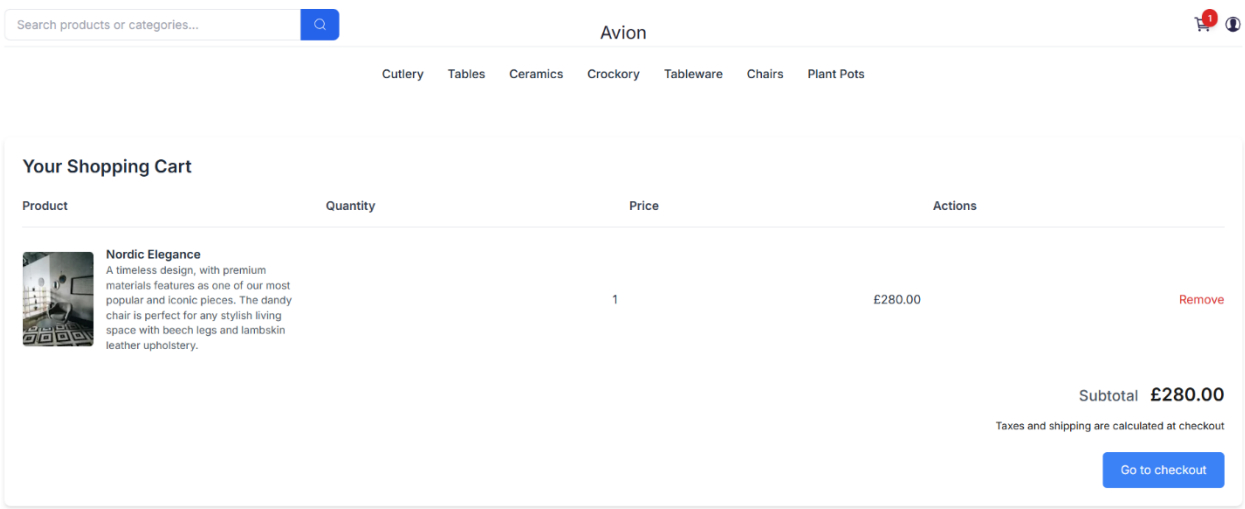
   b. This component allows users to view in-depth information about products they are interested in.



5. **Cart Component:**

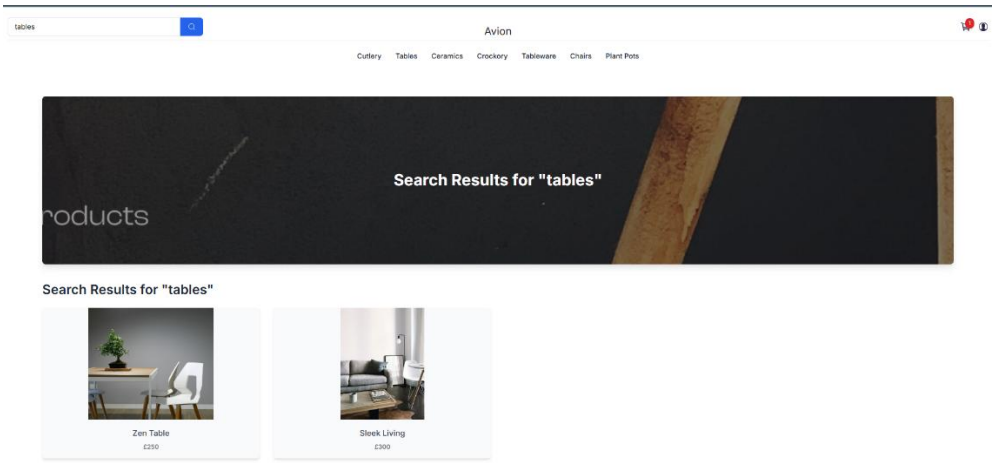
- Developed a cart component to display products added by the user.
- **Displayed Fields:**
    - **Items Added:** Lists the products added to the cart.
    - **Quantity:** Shows the number of units for each product.
    - **Total Price:** Calculates and displays the total cost of items in the cart, dynamically updated with quantity changes.

- Implemented state management to dynamically track and update cart items, providing a seamless shopping experience.



6. **Search Component:**

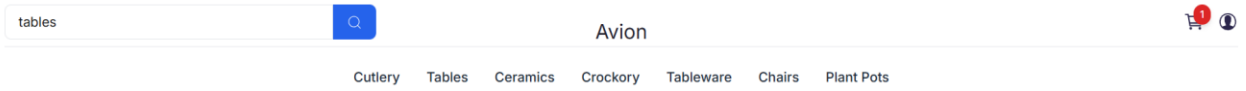
- Designed a search component to enhance the user experience.
- **Functionality:** Enables users to search for products by name or category, ensuring efficient and seamless browsing.



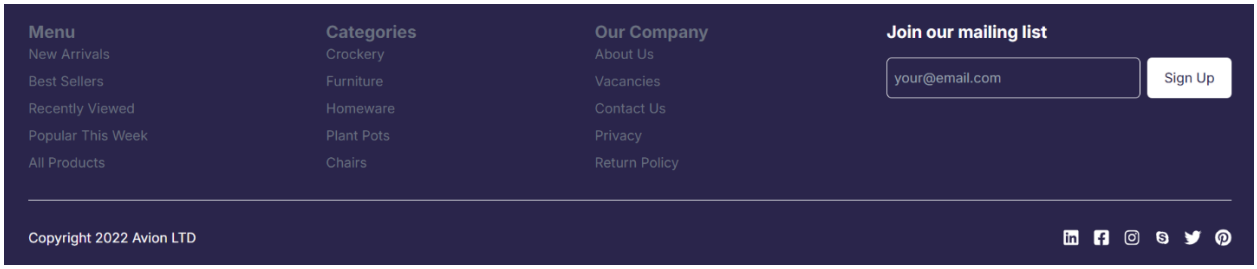
7. **Footer and Header Components:**

- **Consistency:** Created consistent footer and header components for all pages.
- **Header Features:**
  - Search box for product or category search.
  - Cart view.
  - User profile access.
  - Categories of products.
- **Footer Features:** Includes additional site information and relevant links to ensure a consistent user experience.
- **Responsiveness:** Both components are fully responsive, ensuring smooth performance on various screen sizes and devices.
- **Accessibility:** Designed following best practices in web accessibility.

Screenshot of Header Component:

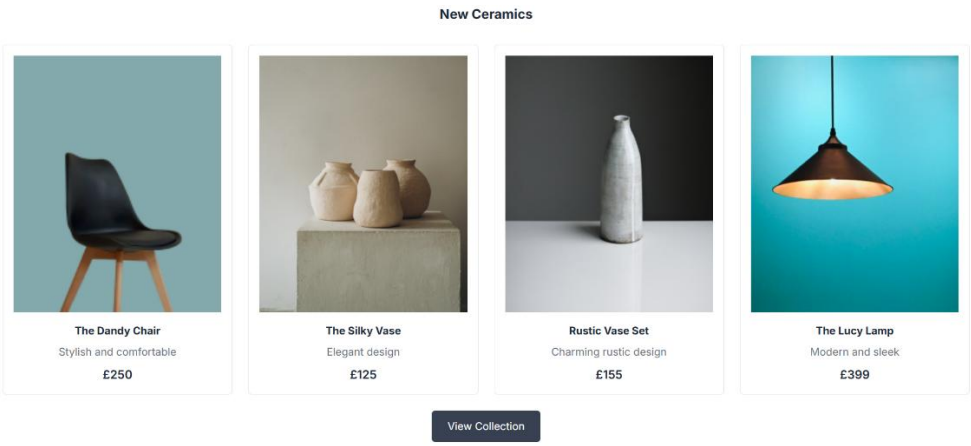


Screenshot of Footer Component:



8. Related Products Component:

- Displayed related products below the product details page to help users easily find similar items.
- Users can click on a product to view its detailed page.



9. Checkout Flow Component:

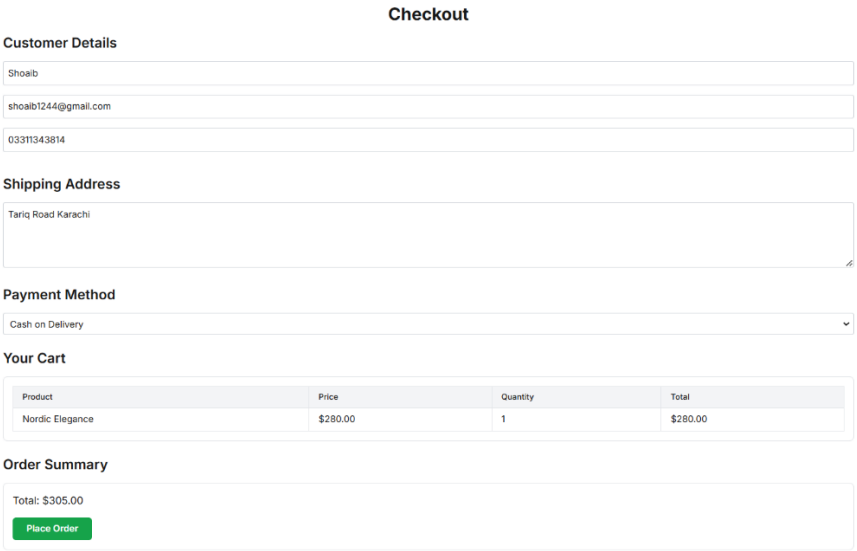
a. Designed a multi-step form to collect:

i. Customer details.
ii. Billing and shipping addresses.
iii. Payment information.

b. Features:

i. Conditional navigation between steps for ease of use.
ii. Order summary page displaying customer details, addresses, and payment method after submission.
iii. Cart validation to ensure items are added before proceeding.

Screenshot of Checkout:



10. **Technical Report Summary:**

- **Development Steps:**
  - Designed components based on project requirements.
  - Integrated components following the specified layout structure.
  - Tested each component's functionality before moving to the next phase.
- **Data Integration:**
  - Fetched dynamic data from Sanity CMS using its API.
  - Ensured real-time updates across components.
- **Testing:**
  - Verified functionality and responsiveness of each component before final integration.

11. **Challenges faced and solutions Implemented:**

- **Challenge:** Ensuring responsiveness on different screen sizes.
  - **Solution:** Utilized Tailwind CSS media queries to handle responsiveness.
- **Challenge:** Managing state effectively across components, particularly for the cart feature.
  - **Solution:** Used React state management tools such as useState and useContext to efficiently track and manage data.
- **Challenge:** Fetching and displaying dynamic product data while maintaining performance.
  - **Solution:** Integrated Sanity CMS to fetch real-time data using queries and optimized API usage for dynamic routing in Next.js.

12. **Best practices followed during development:**

- Maintained modular and reusable code.
- Added clear and concise comments for better code understanding.
- Followed web accessibility standards to create an inclusive experience.

13. **Checklist of day 4:**

- Self-validation checklist.
- Verified styling and responsiveness.
- Ensured code quality.
- Documented tasks and submitted deliverables.