**Day 4 - Dynamic Front-End Components - Furniro**

**Introduction:**

The *Dynamic FrontEnd Components* project focuses on creating a responsive, feature-rich marketplace interface for *Furniro*, a furniture e-commerce platform. The goal was to implement dynamic, reusable components using modern frontend frameworks and tools like Next.js, React.js, and Sanity CMS. This assignment allowed us to explore advanced concepts such as dynamic routing, state management, and API integrations.

**Technologies Used:**

* **Next.js**: For server-side rendering, dynamic routing, and overall frontend framework.
* **React.js**: For creating reusable components and managing UI state.
* **Sanity CMS**: To store and manage dynamic content like product details.
* **Lucide React**: For adding scalable vector icons.
* **Recharts**: For creating interactive charts in the analytics dashboard.
* **React Toastify**: For toast notifications to enhance user feedback.

**Dynamic FrontEnd Components:**

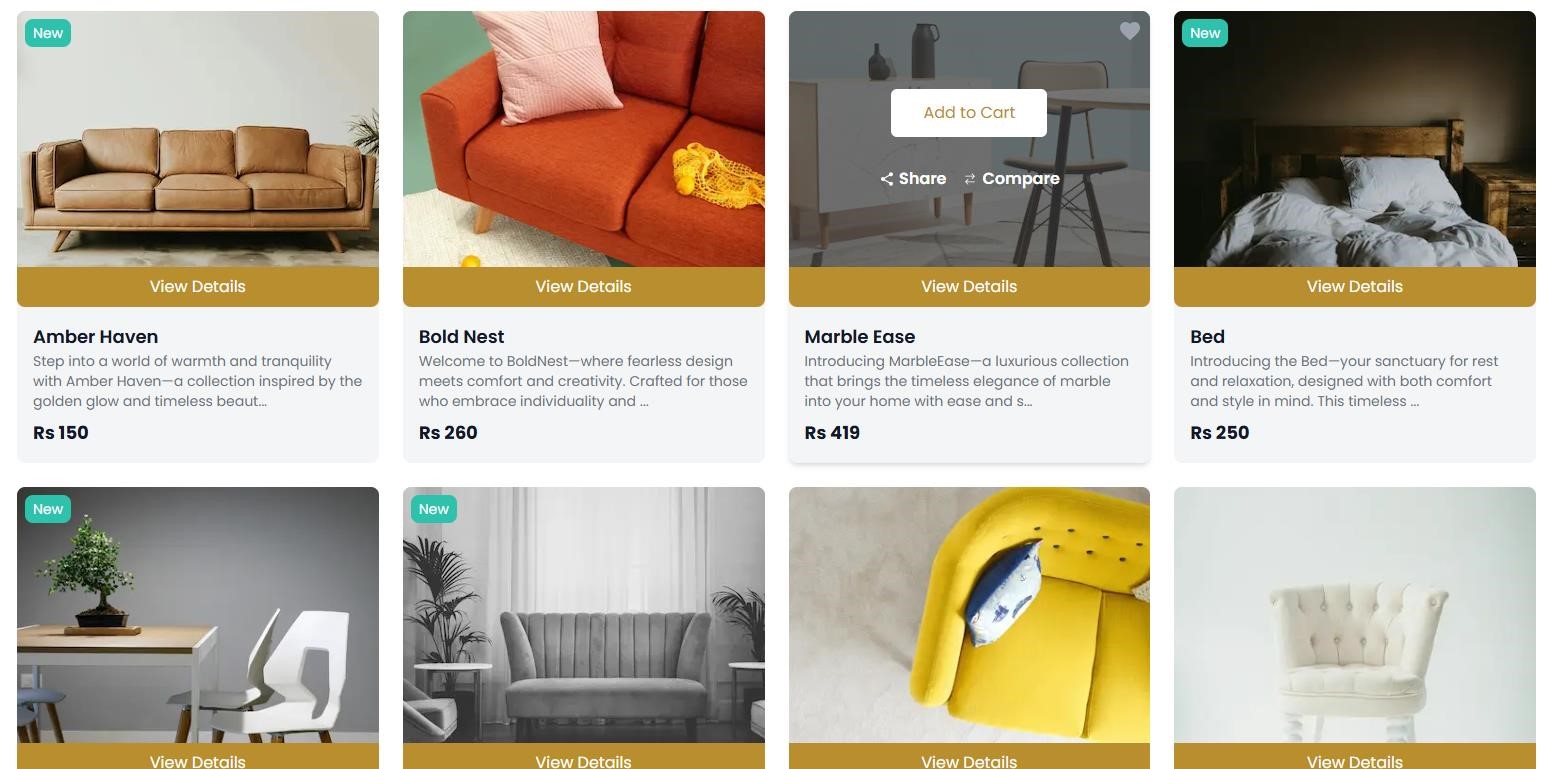
1. **Product Listing Page:**

We implemented a dynamic product listing page by fetching product data from

*Sanity CMS*. Each product was displayed in a grid layout, showing the following fields:

* + Product title
  + Description (shortened for the grid view)
  + Product image
  + Price

The page is fully responsive and adapts to different screen sizes. The grid layout ensures that users can browse multiple products conveniently.

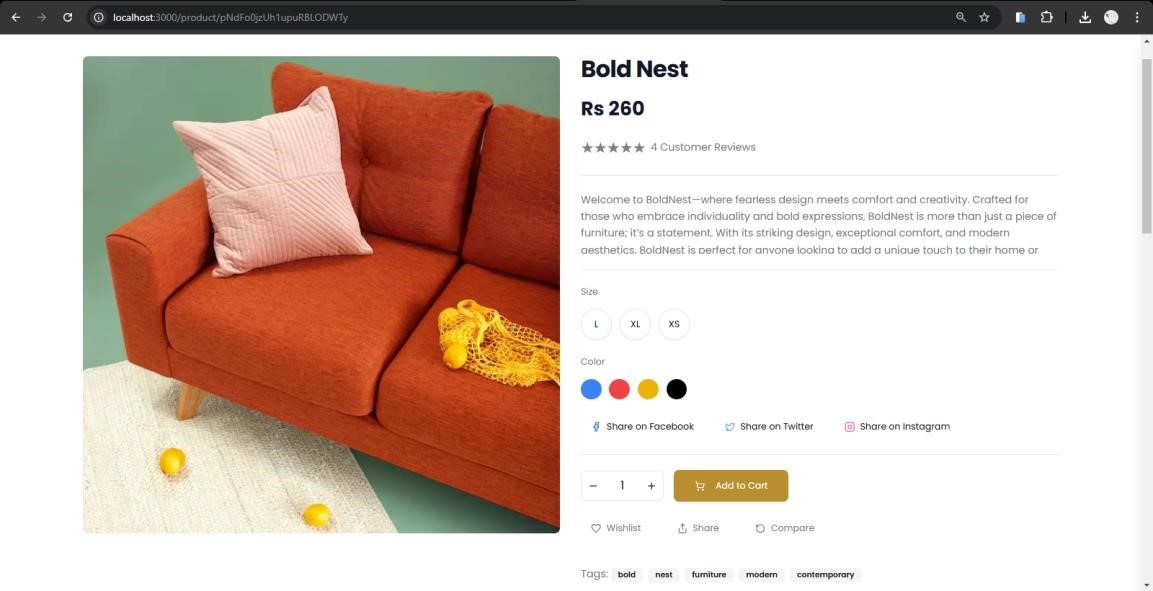


1. **Product Details Page:**

The product details page provides comprehensive information about each product. Using Next.js dynamic routing, we created individual pages for each product based on its unique ID.

**Implementation Details:**

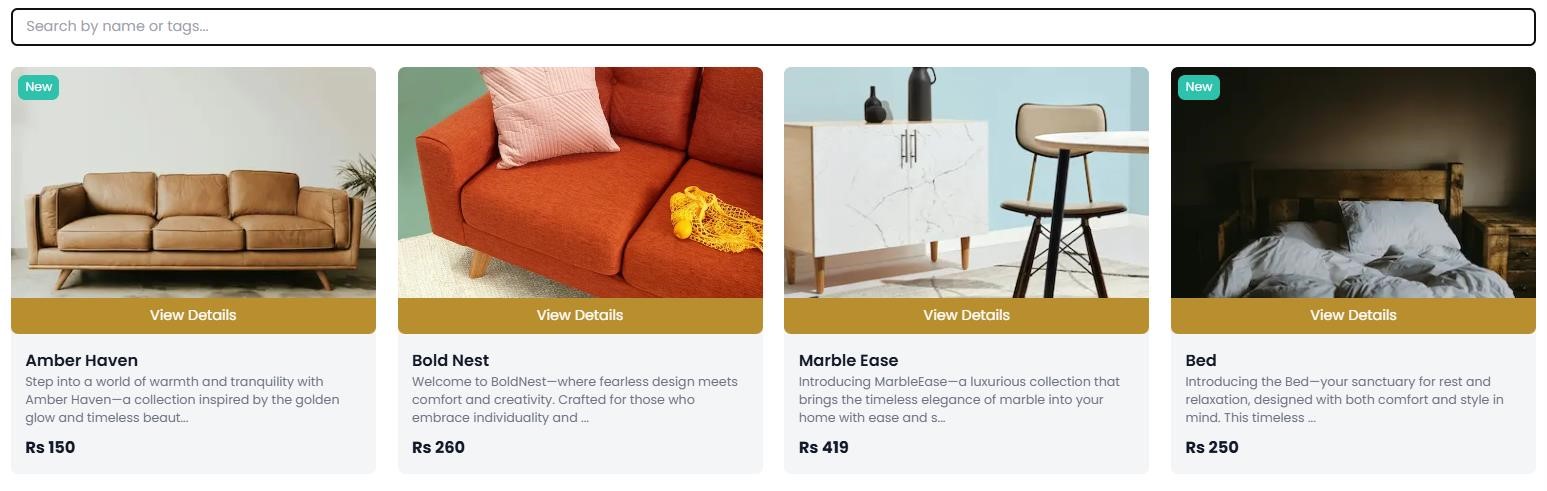
* + **Dynamic Routing**: Leveraged the [id].js file in Next.js to generate pages dynamically based on product IDs.
  + **Reusable UI Elements**: Components like buttons, rating stars, and badges were reused here.



1. **Search Bar:**

The search bar was implemented to enable users to filter products by name or tags dynamically. It is available on the product listing page and updates the displayed products in real-time as the user types.

BEFORE SEARCH:



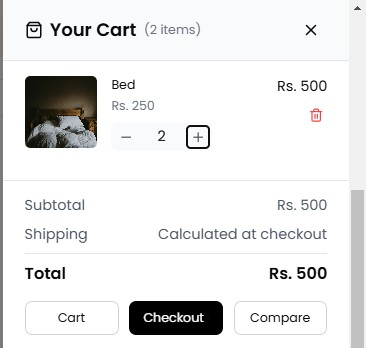
AFTER SEARCH:

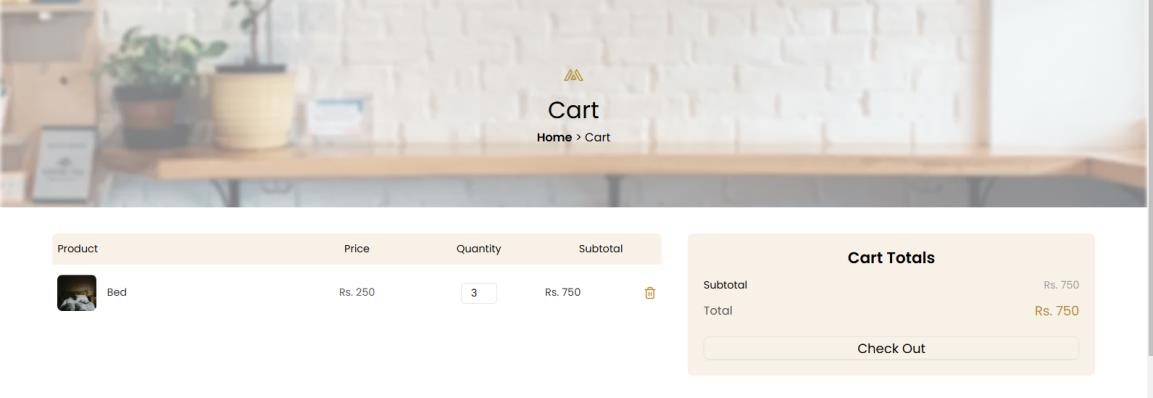


1. **Cart Component:**

The **Cart Component** enables users to view and manage the products they have added to their cart.

**Implementation Details:**

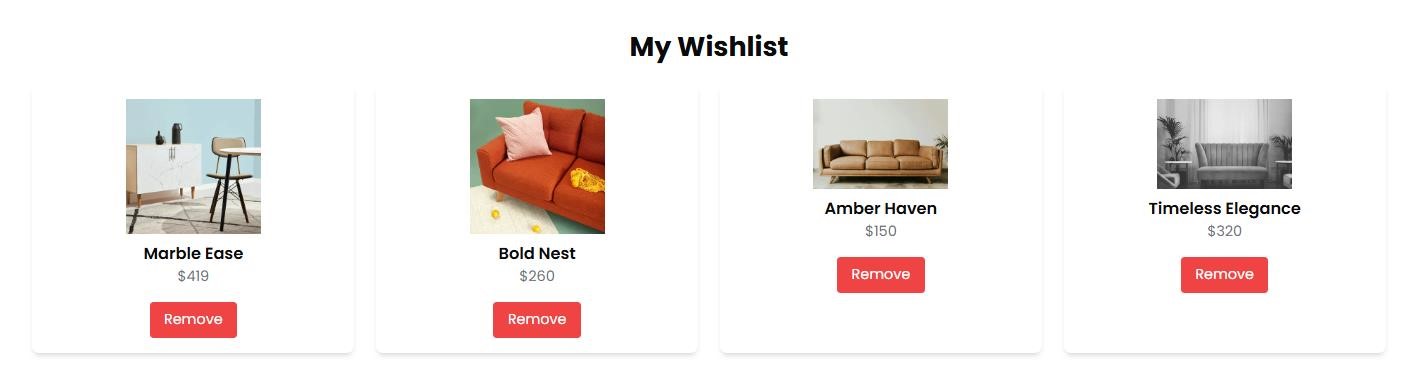
* + **State Management:** We utilized the useState hook in React to track cart items and their quantities dynamically.
  + **Features:**
    - Displays the list of added products with their name, price, and quantity.
    - Allows users to update the quantity of each product. o Automatically calculates and displays the total price. o Removes items from the cart when requested.
    - 



1. **Wishlist Component:**

The **Wishlist Component** allows users to save products for future consideration. **Implementation Details:**

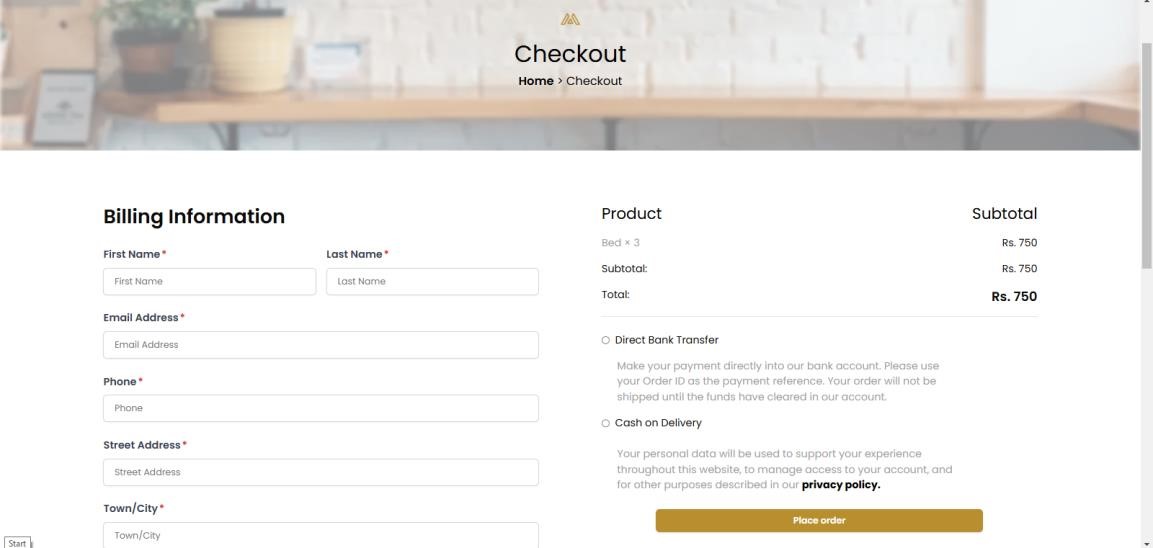
* + **Local Storage:** We used localStorage to persist wishlist data so users can revisit it even after refreshing or closing the browser.



1. **Checkout Flow Component:**

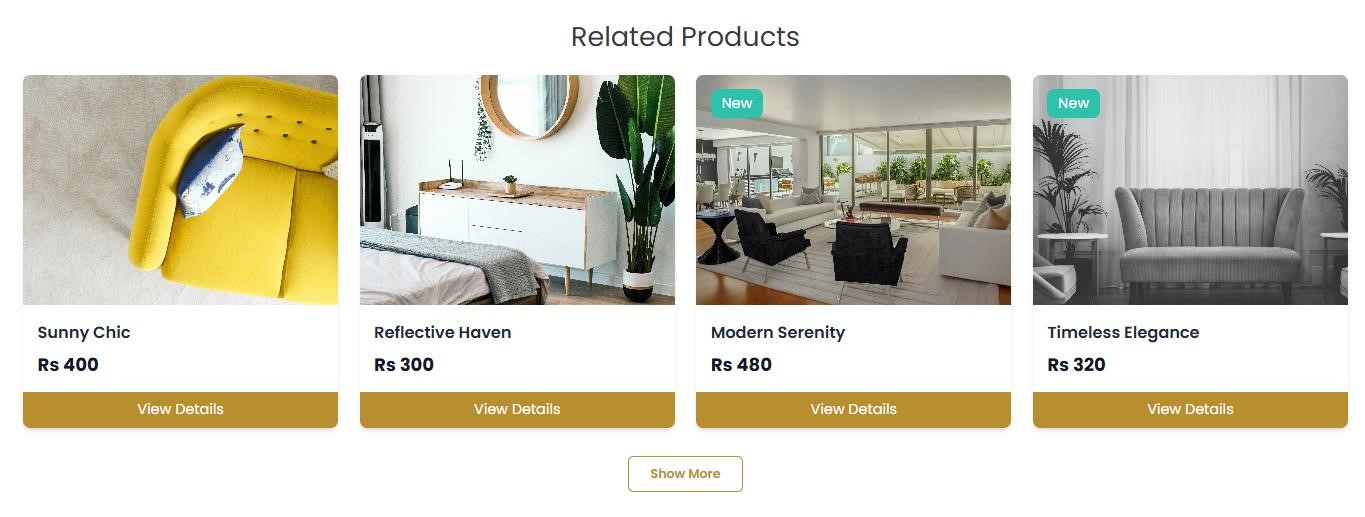
The **Checkout Flow Component** simplifies the process of placing an order.

* + **Form Creation:**
    - Includes fields for billing and shipping information such as name, address, city, and country.
    - Collects payment details for order processing.  **Validation:**
    - Ensures all required fields are filled before allowing the order to proceed. o Displays error messages for missing or invalid inputs.



1. **Related Products Component:**

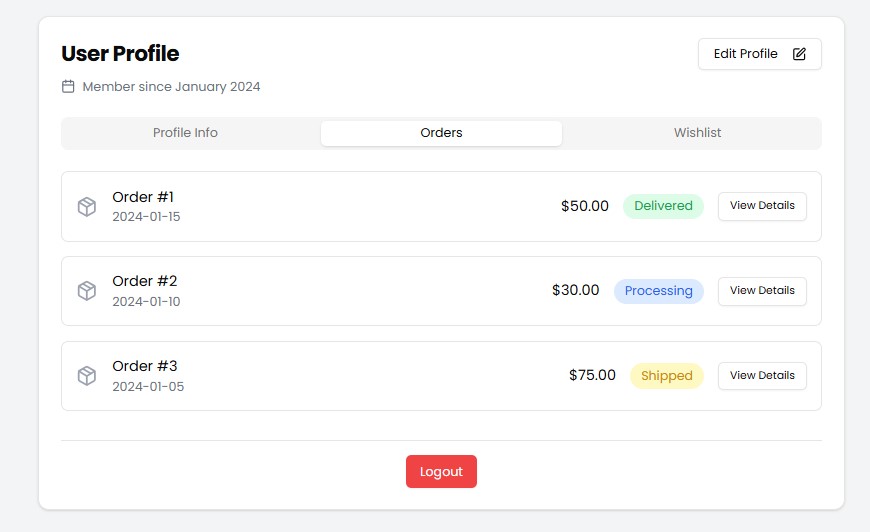
The **Related Products Component** enhances user experience by suggesting products related to the one currently being viewed.



1. **User Profile Component:**

Displays user-specific information and order history.

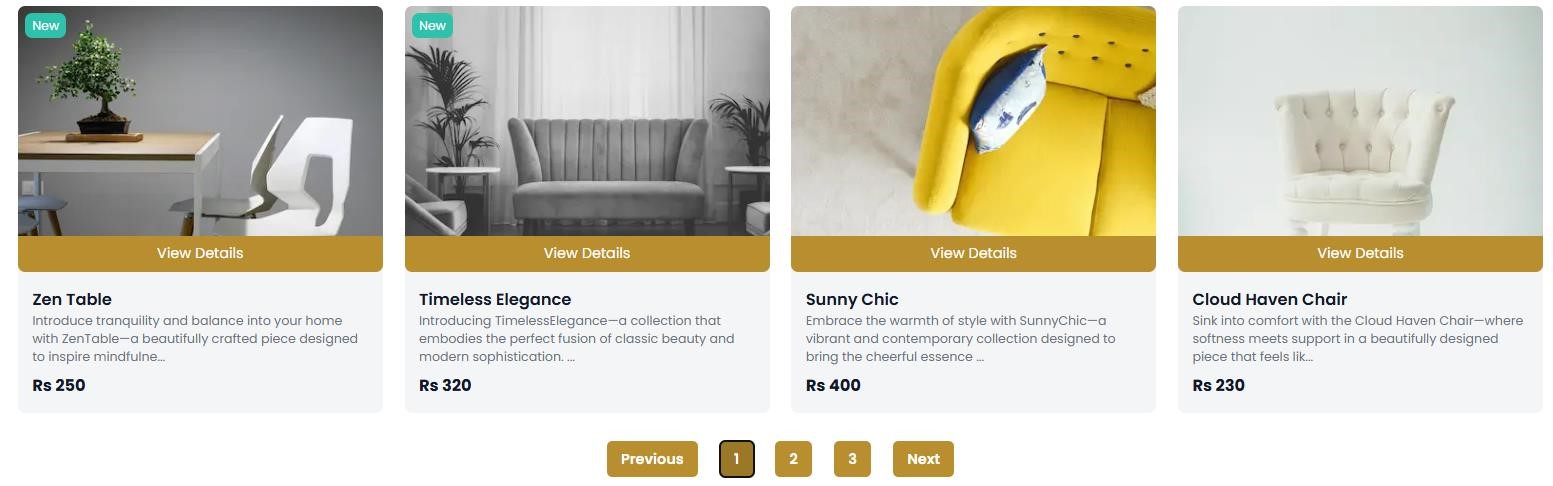
* + **Features:**
    - User details: name, email, saved addresses.
    - Order history with links to individual orders.
  + **Authentication:** Redirects non-logged-in users to the login page.



1. **Pagination Component:**

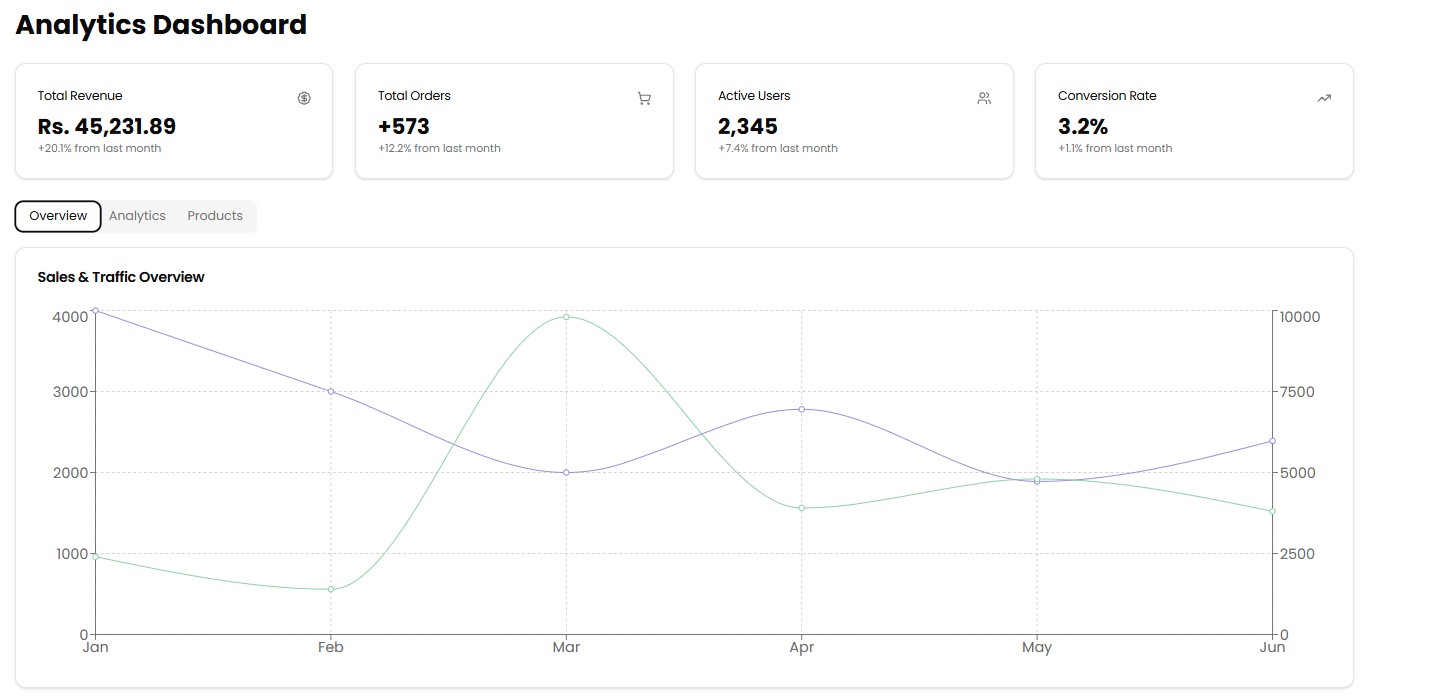
Organizes large product lists into pages for easier navigation.

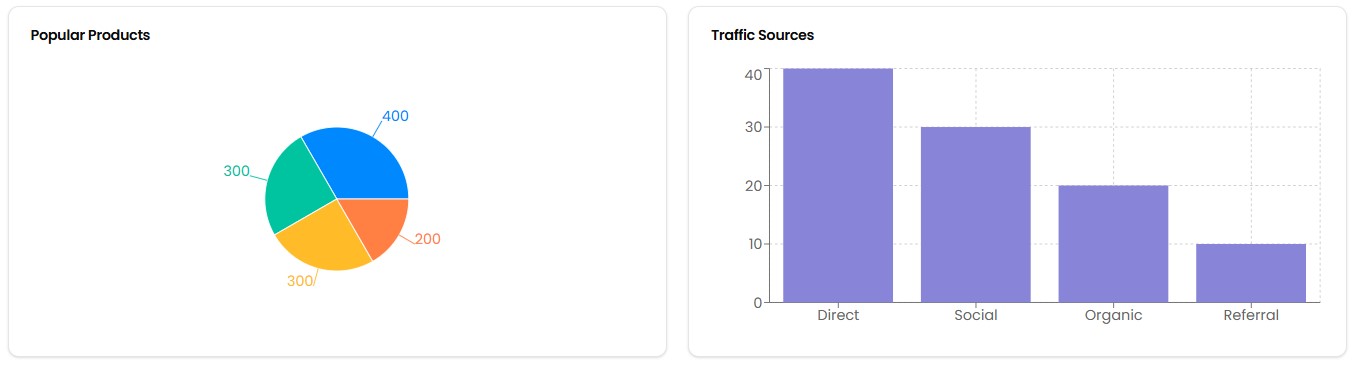
* + **Navigation:** Includes previous, next, and numbered buttons.



1. **Analytics Dashboard Component:**

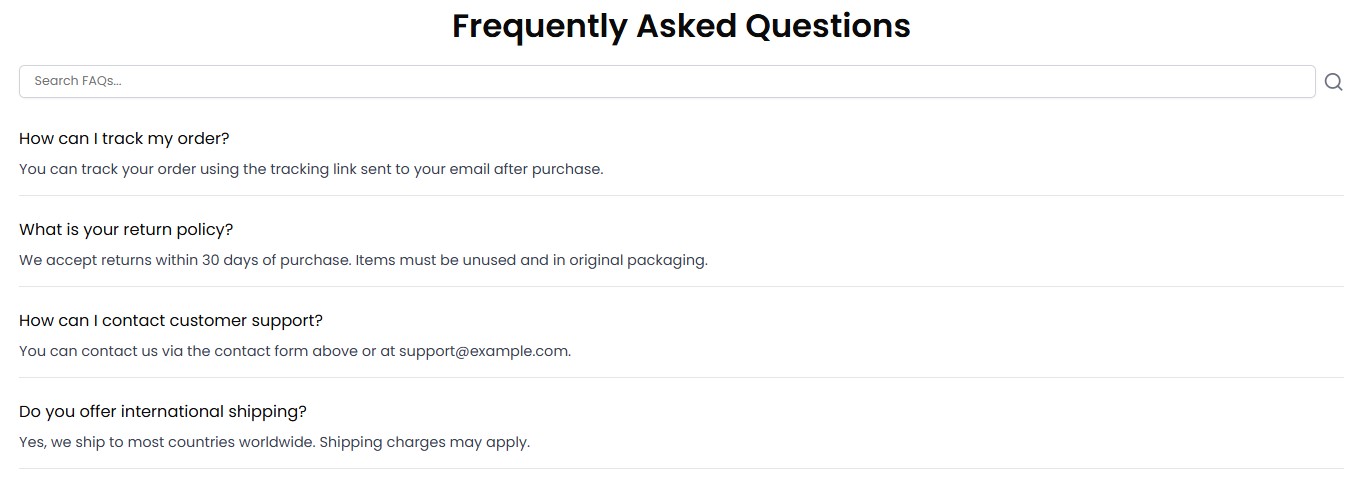
Displays key performance indicators (KPIs) such as sales, traffic, and popular products.





**11.FAQ and Help Center Component:**

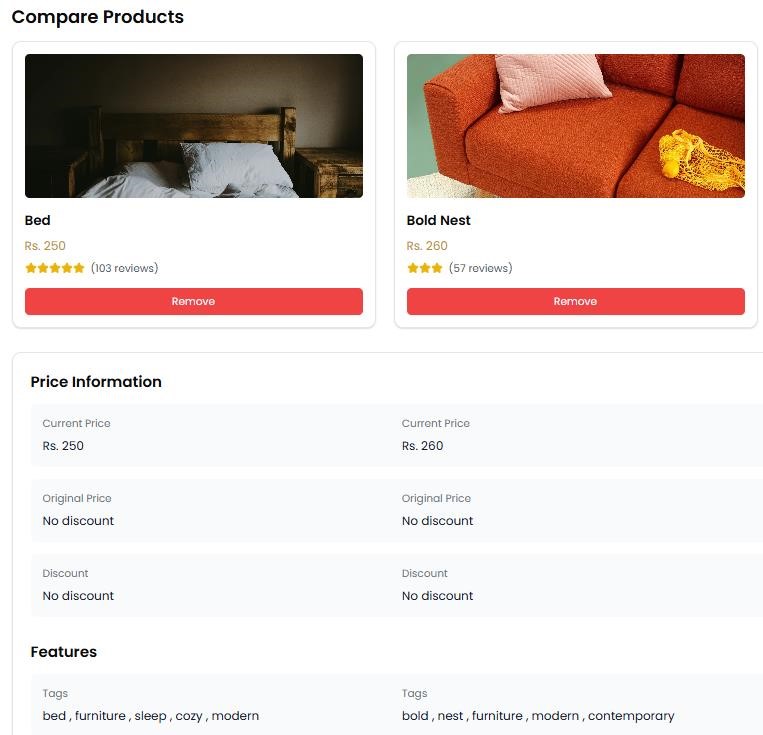
Provides a searchable FAQ section and support options.



1. **Product Comparison Component:**

Allows users to compare multiple products side by side.

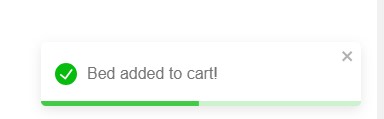
* + **Details:** Compares product features, price, and ratings.

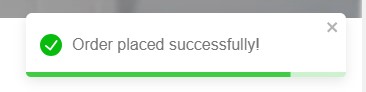
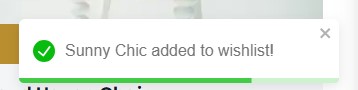


1. **Notifications Component**

Displays real-time alerts for actions like adding to the cart, errors, or successful purchases.

* + **Toast Notifications:** For instant feedback.





**Self Validation Checklist:**

**Task Status**

FrontEnd components ✔

Styling and responsiveness ✔

Code Quality ✔

Documentation and submission ✔

Final Review ✔

-Made by Nabeel Ali