**DYNAMIC FRONTEND COMPONENTS**

**OF**

**MARKETPLACE BUILDER**

**BUSINESS TYPE: GENERAL E-COMMERCE**

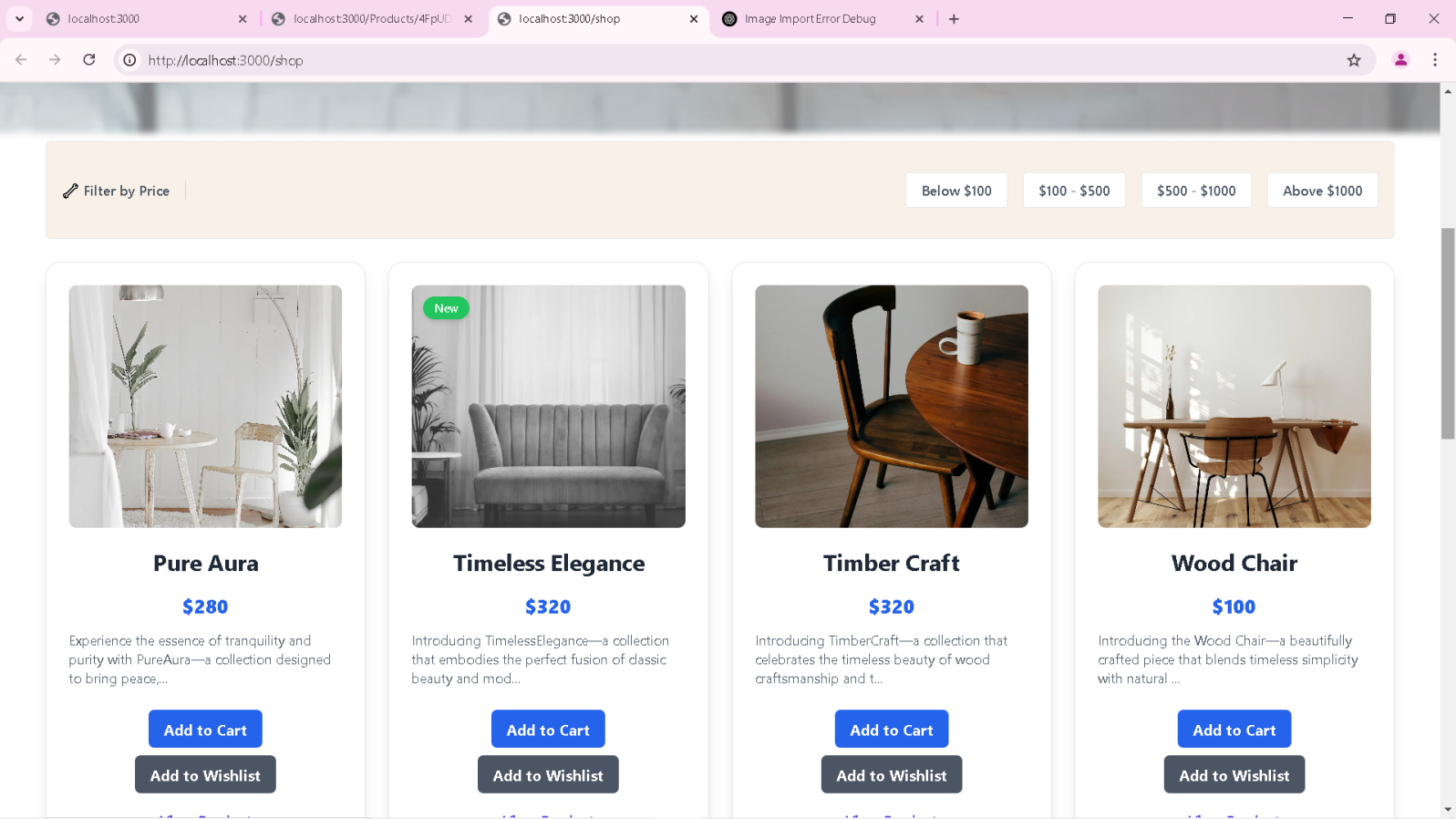
**HACKATHON TASK # DAY 4**

|  |  |  |
| --- | --- | --- |
| **Authour: Anam Shergill** | **Roll No: 472926** | **Day&Time: Sat 2-5** |

**Functional Deliverables**

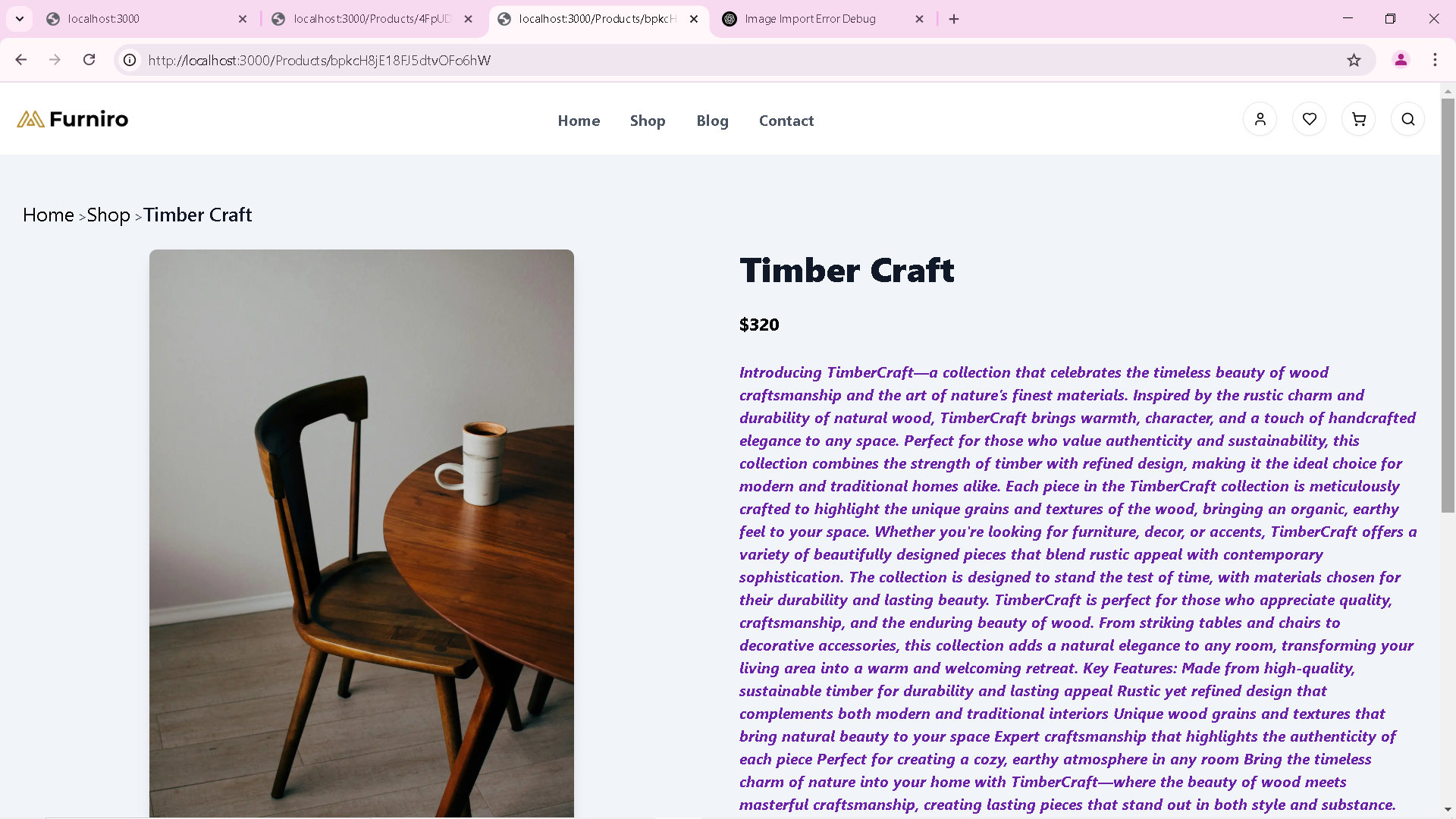
**Product listening page with dynamic data:**

* The product listing page is fully functional & dynamically fetches product data from sanity ( cms ) or APIs.
* Products are displayed with essential details such as image, title, price & rating.
* Data is rendered dynamically to ensure the page updates, whenever new products are added to the CMS.



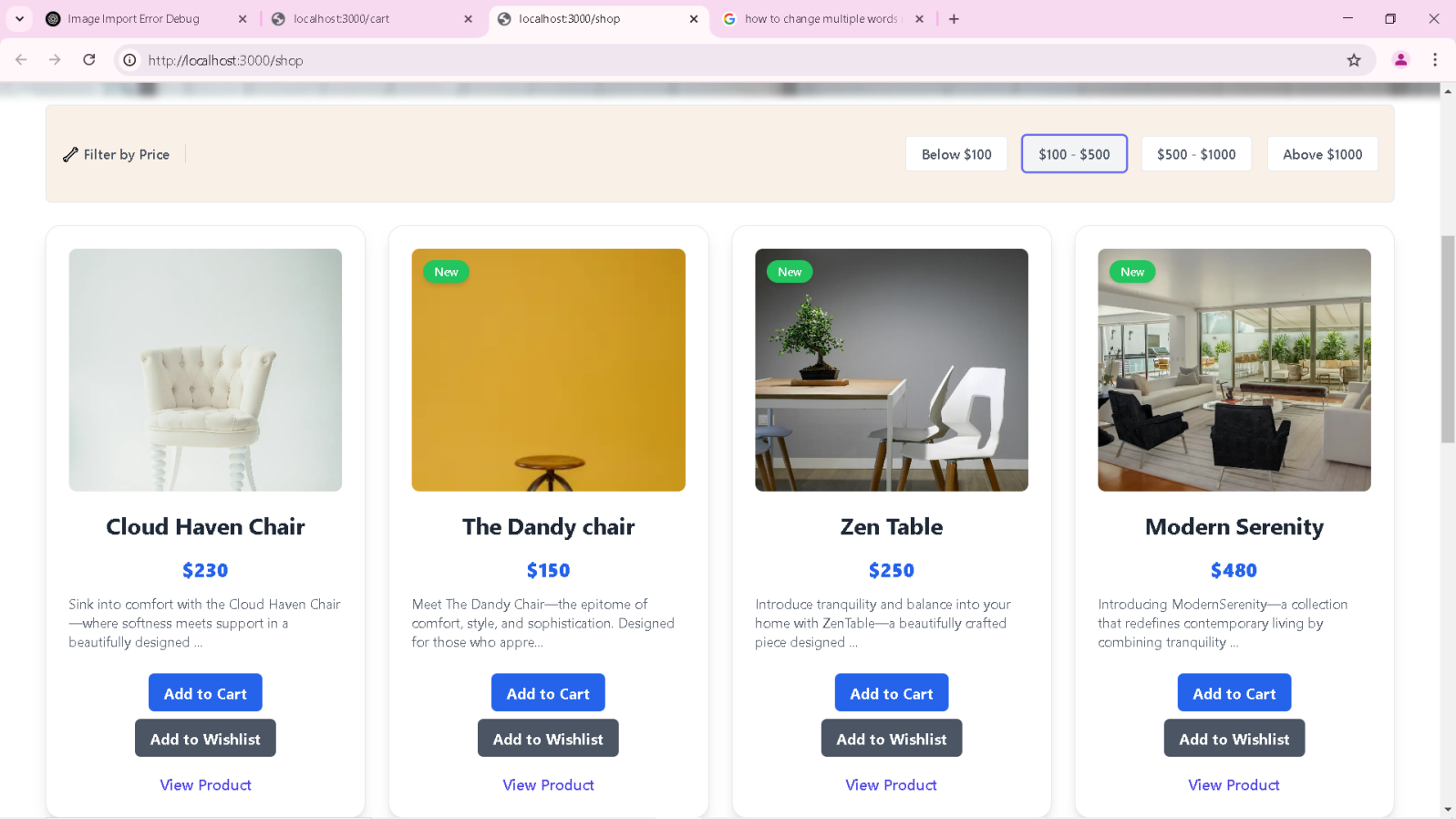
**Individual Product Detail Pages :**

* Each product detail page is implemented using dynamic routing. (products/[id])
* Display accurate product details fetched dynamically based on the product id.
* Features like : high quality product images, title, description, price, rating & status.
* Related products section to suggest similar items.



**Advanced Price Filters :**

* Implemented price filters to refine product viwers dynamically.
* Users can filter products by price range.

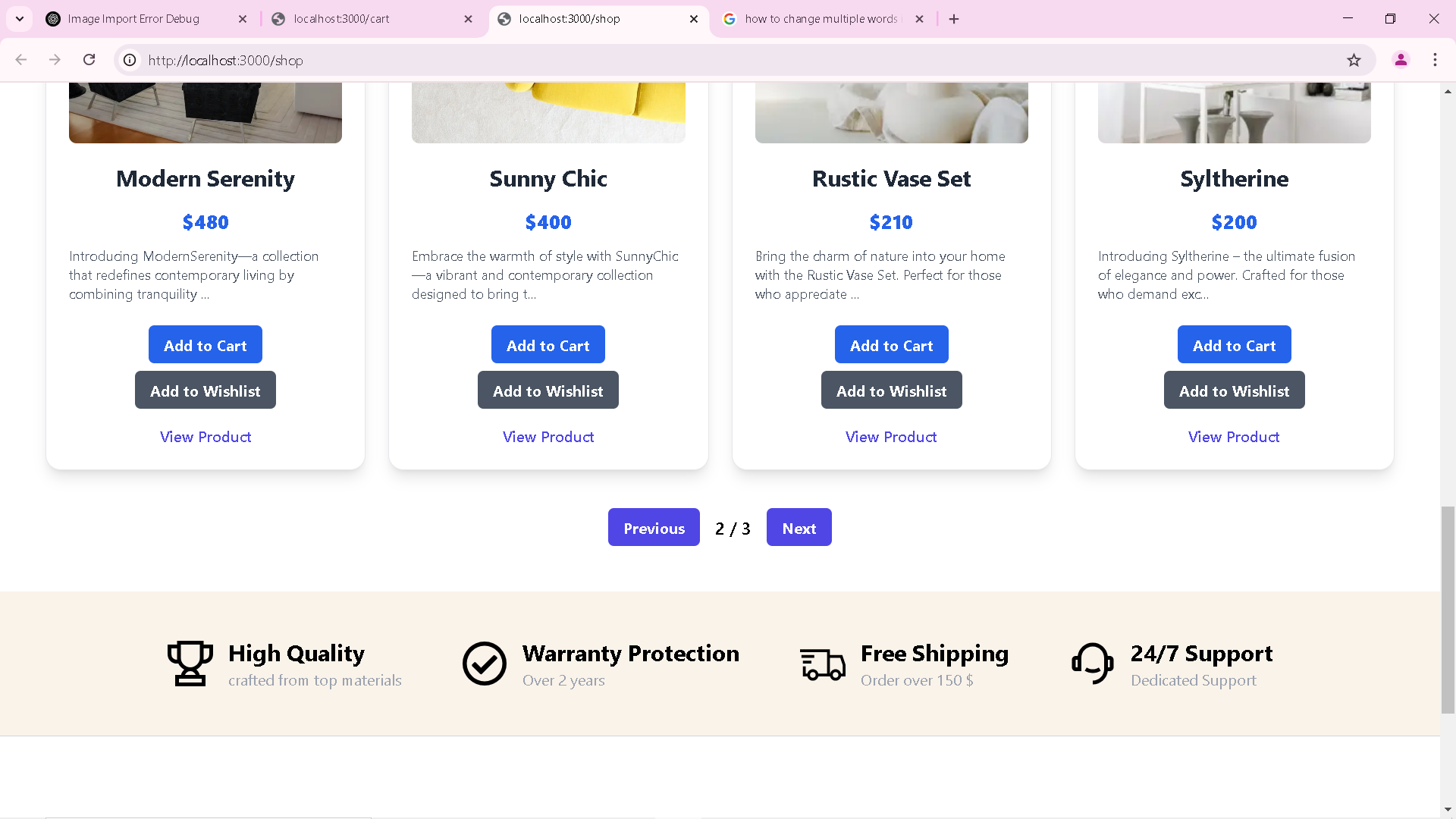


**SEARCH BAR :**

* A searchbar us integrated to filter products by name or tags.
* Features like; real-time search result & case-insensitive search functionality.

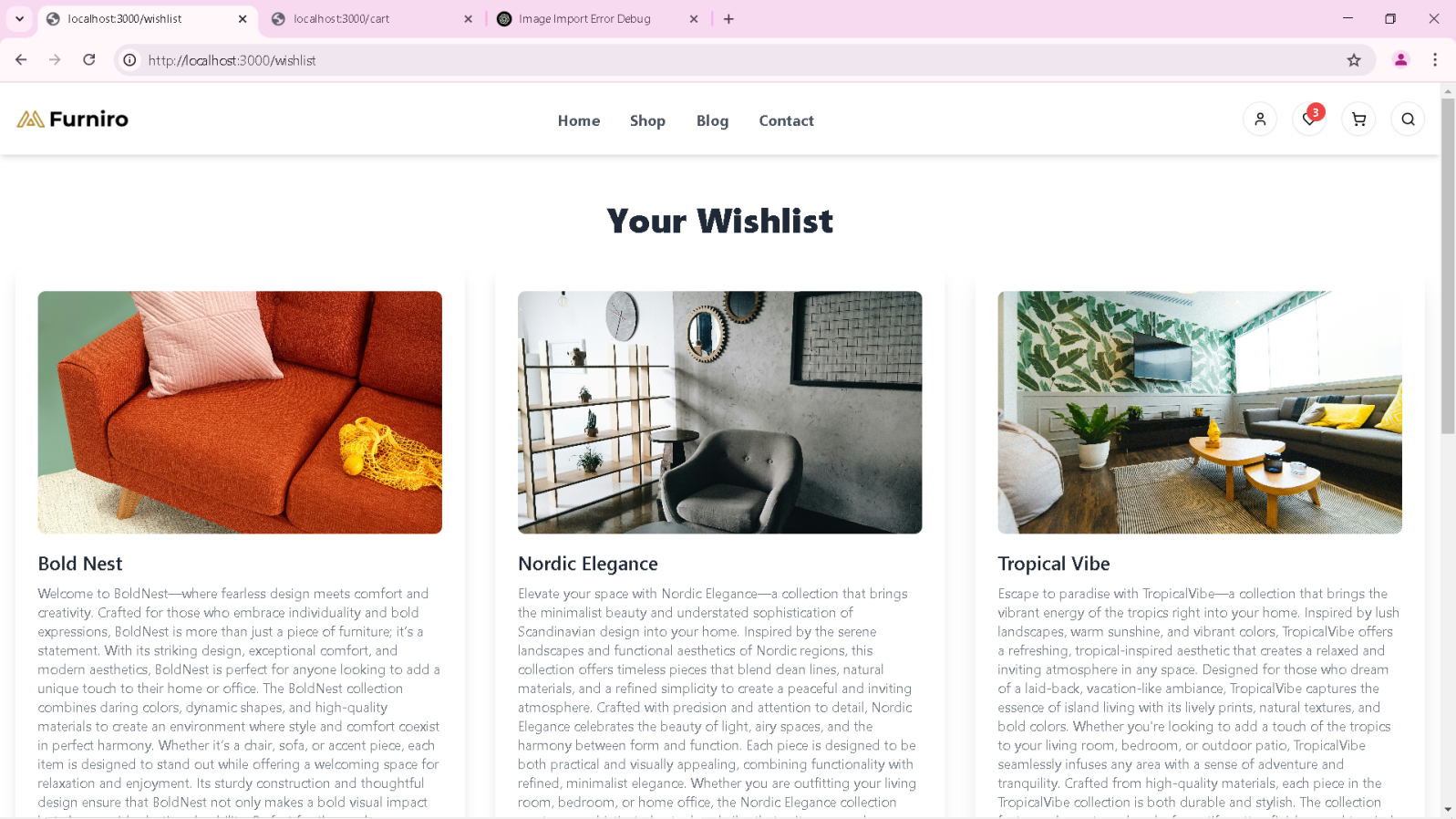
**Additional Features :**

* Implemented pagination to load products efficiently, reducing load timer for large details.
* Display on the products detail page to encourage further exploration & purchases.



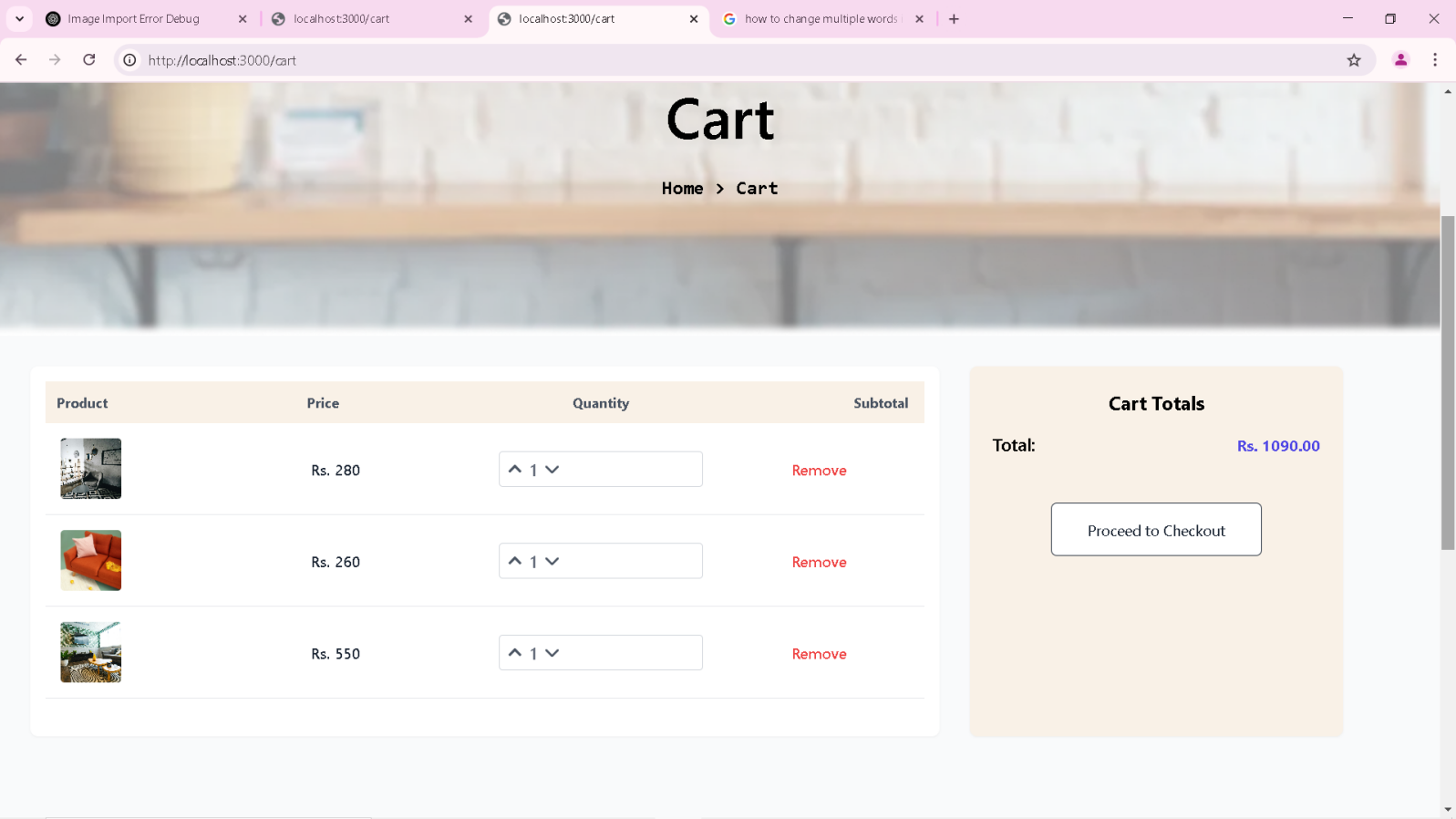
**Add To Wishlist :**

* Products are adding to Wish list.

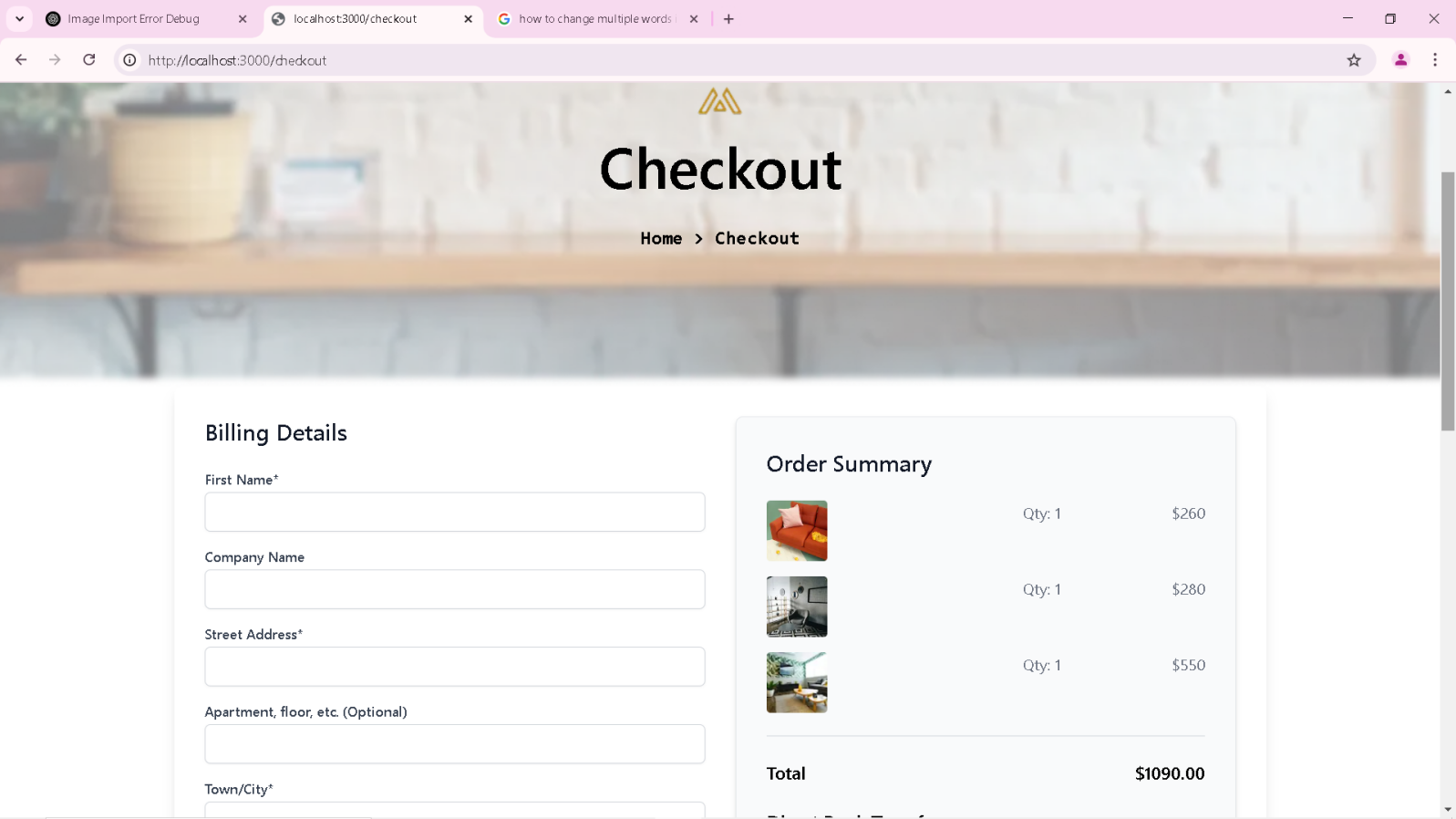


**Add To Cart :**

* Products are adding to Cart.



**Checkout Page :**



**Responsive Styling :**

* All components are styled for responsiveness, ensuring a professional look across devices.
* Used css frameworks & custom classes for a polished UI.

**DOCUMENTATION**

**Steps Taken To Build & Integrate Components**

**1 Dynamic products Listing :**

* Created a AllProducts.tsx component to fetch & render products dynamically from sanity CMS.

**2 Dynamic Routing for Products detail:**

* Used next.js dynamic routing [Id].tsx to implement individual product page.

**3 Category Filter :**

* Build reuseable components ( CategoryFilter ) with state management.

**4 Pagination :**

* Implementd logic to load products in chunks & display related products based on categories OR tags.

**5 Styling & Responsiveness :**

* Ensure UI components are responsive using Tailwing css & media queries.

**FACED CHALLENGES & IMPLEMENT SOLUTIONS**

1. **Data Fetching Dynamically :**

* Challang : ensuring smooth data fetching without delay.
* Solution : used async functions & proper error handling.

1. **Optimization :**

* Change : real-time filtering without performance lag.
* Solution : use debouncing to optimize search option.

1. **Dynamic Routing :**

* Challenge : handling non-existence product Ids.
* Solution : added fallback UI for error handling.

**BEST PRACTICES FOLLOWED**

* Modular & unuseable component design.
* Clean & maintainable code with proper comments.
* Efficient data fetching with error handling.
* Responsive design principles for cross-device compatability.

**\*..\*..\*..\*..\*..\***