Detail document

Outline of the Document

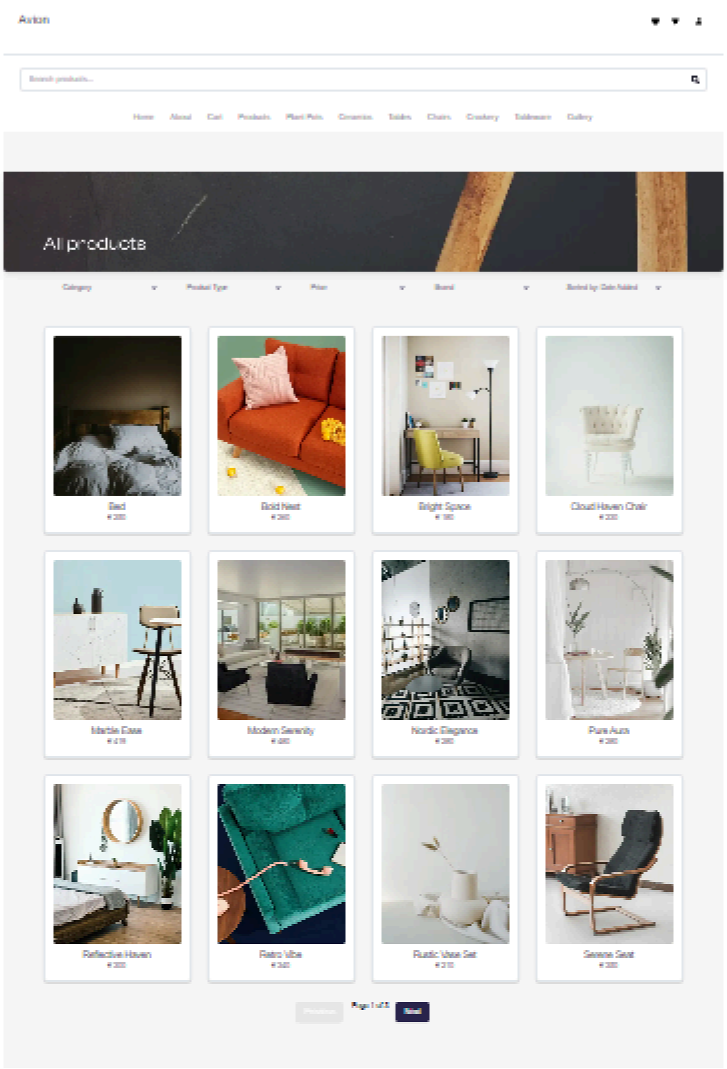
1. Introduction
2. Dynamic Components and Functionalities
   - Cart Functionality
   - Product Amount Totals
   - Product Details and Specifications
3. Sanity Integration
   - Setting up Sanity CMS
   - Fetching Products for the Listing Page
4. Product Listing Page
   - Design Overview
   - Dynamic Features
5. Conclusion

Detailed Document

1. Introduction

This document outlines the architecture, dynamic components, and functionalities for building a product listing page integrated with Sanity CMS. Key features include a dynamic cart system, real-time product amount calculations, and a detailed product specification section. The document also explains the setup process for Sanity CMS and its integration with the frontend to

ensure seamless data management.



2. Dynamic Components and Functionalities

2.1 Cart Functionality

- **The cart functionality will allow users to:**
  - **Add products to the cart dynamically.**
  - **Increase or decrease the quantity of items.**
  - **Remove items from the cart.**
  - **Real-time updates to reflect cart totals.**
- **Implementation Details:**
  - **Use React's state management with `useContext` or `Redux` for global cart state.**
  - **Dynamic rendering of cart items using React components.**
  - **LocalStorage or session management to preserve cart state on page reloads.**

2.2 Product Amount Totals

- **Real-time calculations for the total amount in the cart.**
  - **Total = (Quantity × Product Price) for all products.**
  - **Include options for shipping and tax calculations.**
- **Implementation Details:**
  - **Write utility functions to calculate subtotal, taxes, and total.**
  - **Recalculate totals on cart updates using `useEffect` or Redux selectors.**

2.3 Product Details and Specifications

- **Display detailed product information on the product detail page, including:**

- **Product name, price, description, and specifications.**
- **High-quality product images.**
- **Ratings and reviews (optional).**

- **Implementation Details:**

    - **Fetch product data dynamically from Sanity CMS using GraphQL or GROQ.**
    - **Design reusable components for displaying specifications and details.**

---

3. Sanity Integration

3.1 Setting up Sanity CMS

Create a new Sanity project using the Sanity CLI:

npx sanity init

-

Define schemas for products, categories, and other necessary entities. Example:

```
 export default {
  name: "product",
  type: "document",
  title: "Product",
  fields: [
    { name: "name", type: "string", title: "Name" },
    { name: "price", type: "number", title: "Price" },
    { name: "description", type: "text", title: "Description" },
    { name: "image", type: "image", title: "Product Image" },
  ],
};
```

-

3.2 Fetching Products for the Listing Page

Fetch products using Sanity's GROQ queries.

Example query:

```
 const query = `*[_type == "product"]{
  name, price, description, "imageUrl":
image.asset->url
}`;
```

-

Use `@sanity/client` for data fetching:

```
 import sanityClient from "@sanity/client";


const client = sanityClient({
  projectId: "<your-project-id>",
  dataset: "production",
  useCdn: true,
});


export const fetchProducts = async () => {
  return await client.fetch(query);
};
```

-

---

4. Product Listing Page

4.1 Design Overview

- A grid-based layout for product cards.
- Each card displays the product image, name, price, and a button to add the product to the cart.

- A search bar and filters for categories or price range.

4.2 Dynamic Features

- Hover Effects: Zoom on product images for a closer look.

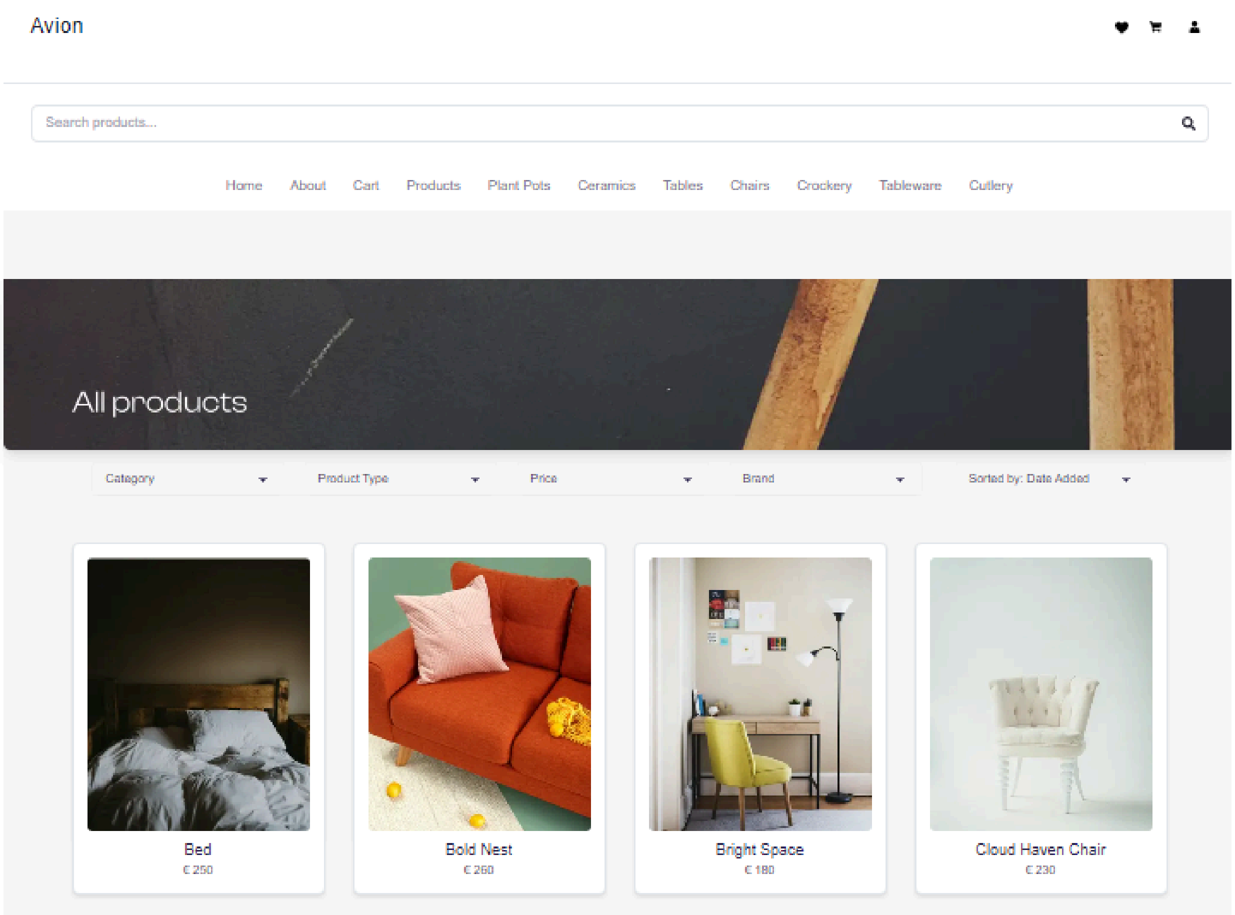
- Pagination: Load products in batches for better performance.

- Sorting and Filtering: Allow users to sort by price, popularity, etc.

- Implementation Details:
  - Use Tailwind CSS for responsive design.
  - Implement search, filter, and pagination logic dynamically.

5. Conclusion

By combining React's dynamic capabilities with Sanity CMS, this architecture ensures a seamless user experience for both administrators managing products and customers exploring the product listing page. The outlined features, from cart functionality to product details, establish a foundation for scalable and user-friendly



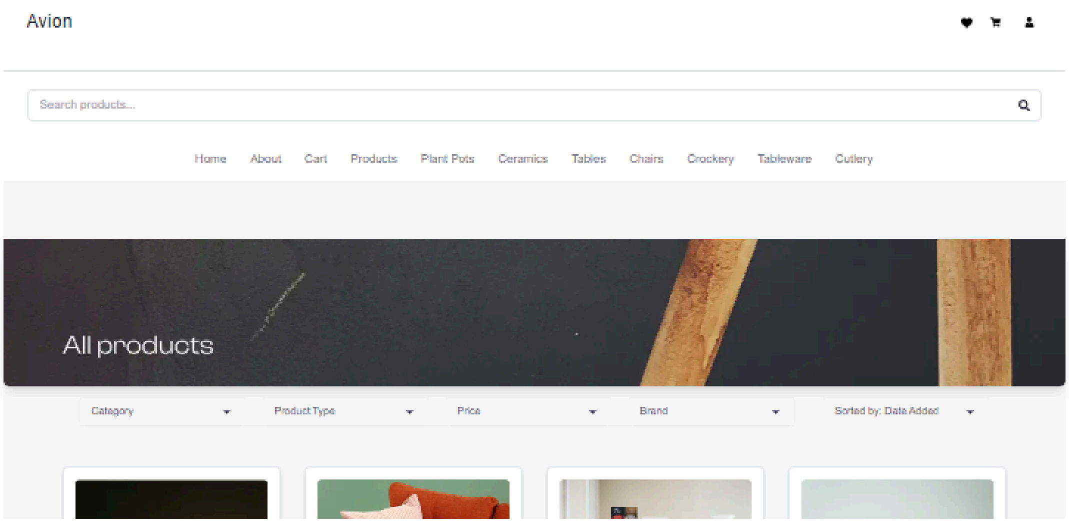
All products

Bed
€ 250

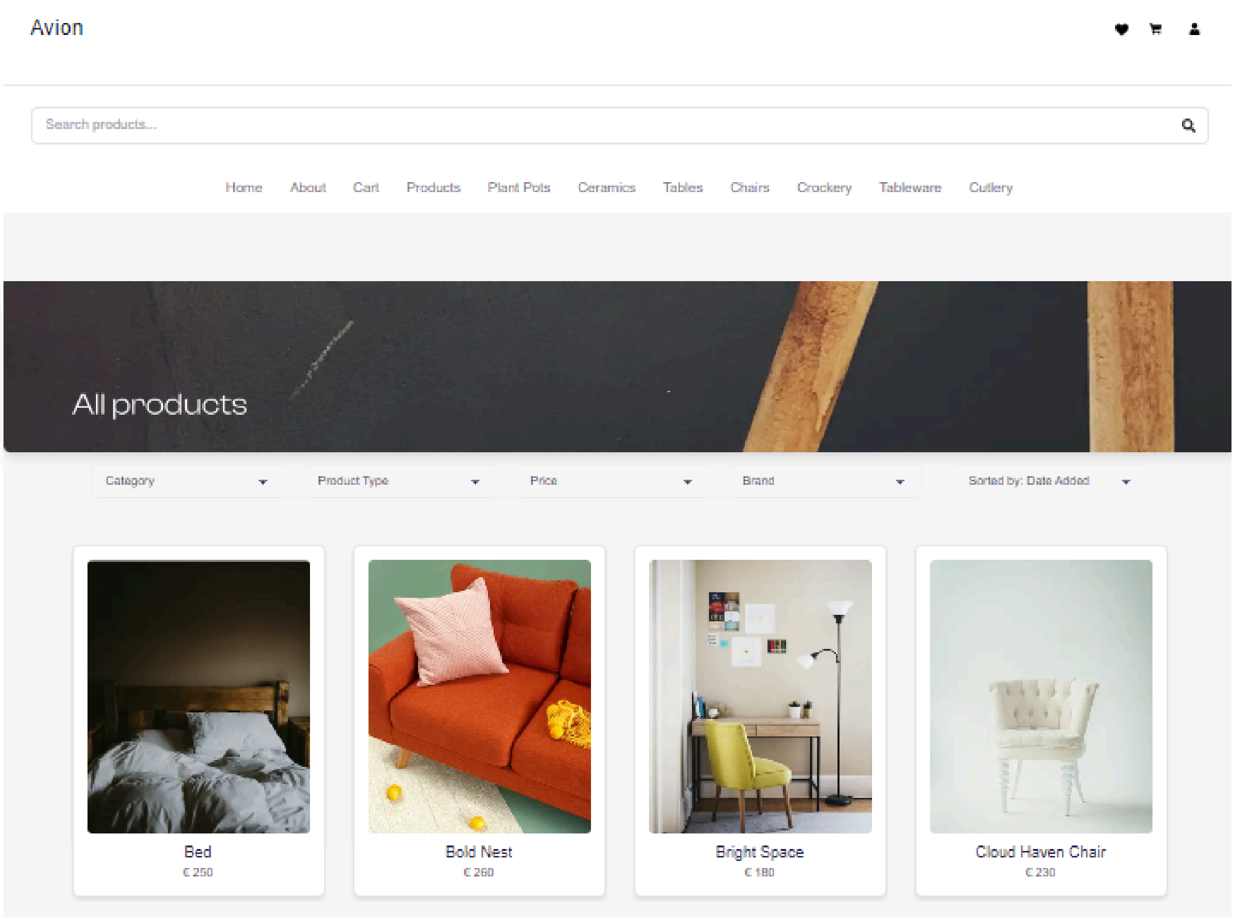
Bold Nest
€ 260

Bright Space
€ 180

Cloud Haven Chair
€ 230


Avion
Search products...
Home
About
Cart
Products
Plant Pots
Ceramics
Tables
Chairs
Crockery
Tableware
Cutlery
All products
Category
Product Type
Price
Brand
Sorted by: Date Added

eCommerce applications.



Bed
€ 250

Bold Nest
€ 260

Bright Space
€ 180

Cloud Haven Chair
€ 230

PREPARED BY:KANWAL SAMUEL