

# HACKATHON DAY 4 REPORT

## DYNAMIC FRONTEND COMPONENT

**Marketplace Name:** Avion Furniture Store

### Introduction

This report outlines the work accomplished during the hackathon, focusing on building a dynamic and interactive web application. The project's primary objective was to fetch data from Sanity, implement dynamic routing, and develop an 'Add to Cart' functionality using modern web technologies.

### Technologies Used

The project utilized the following technologies:

- ✓ **Next.js:** For server-side rendering and efficient routing.
- ✓ **Sanity:** As a headless CMS to manage and fetch content.
- ✓ **Shadcn:** For UI components and styling.
- ✓ **React Icons:** To enhance UI with various icon elements.

### Implementation Steps

#### 1. Data Fetching from Sanity:

- ✓ Connected the application with Sanity CMS.
- ✓ Retrieved product data dynamically using GROQ queries.
- ✓ Displayed the fetched data on the website with responsive UI components.

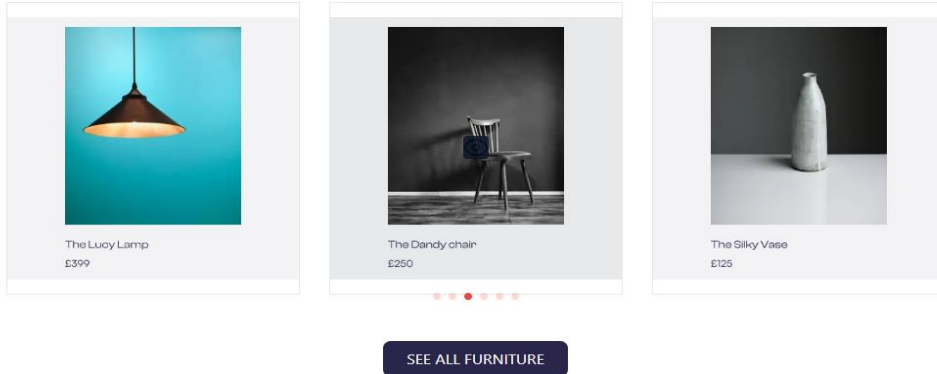
```
✓  
const getData = async () => {  
  const query = `*[_type == 'product'] {  
    _id,  
    product_id,  
    name,  
    description,  
    images,  
    price_id,  
    price  
  }`;  
}
```

## 2. Dynamic Routing:

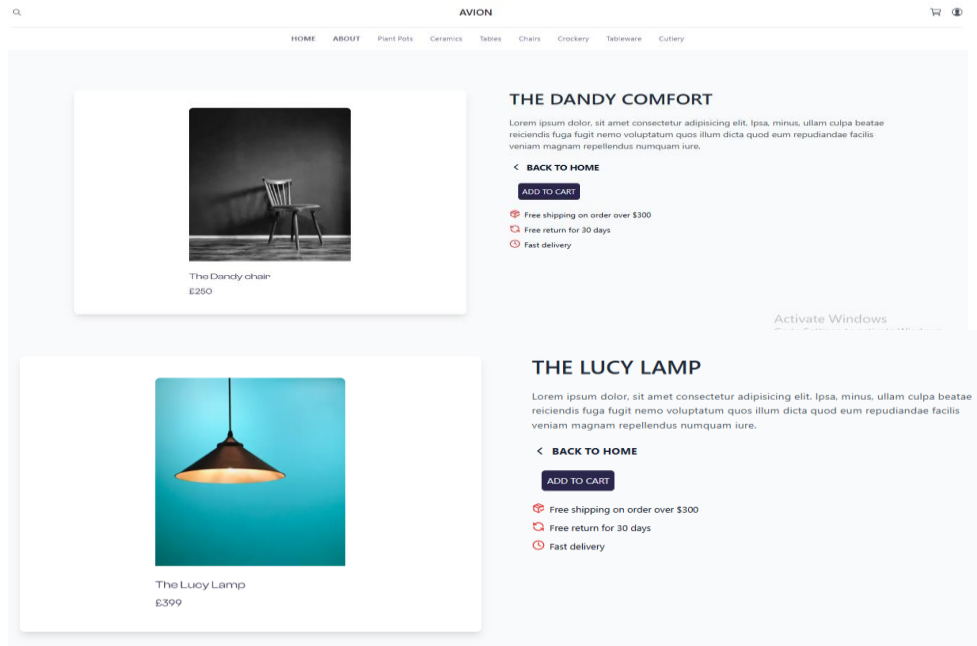
- ✓ Implemented dynamic routing for the "Most Popular Products" section.
- ✓ Ensured that each product has its dedicated page generated dynamically based on data from Sanity.

### MOST POPULAR PRODUCTS

The World's Premium Brand In One Destination.



### Individual Detail Page for Each Product



## ✓ Responsive Design



## Challenges Faced

During the development process, the following challenges were encountered:

### 1. Add to Cart Functionality:

- Managing the state efficiently to store and update cart items dynamically.
- Ensuring persistent cart data across page reloads.

### 2. Dynamic Routes from Sanity:

- Structuring the routing dynamically based on the product data fetched from Sanity.
- Handling route generation and ensuring smooth navigation.

## Future Improvements

- Enhance the cart functionality by adding quantity management and checkout features.
- Implement client-side caching to optimize data loading.
- Improve UI/UX for better user engagement.

## Conclusion

The hackathon project successfully implemented dynamic data fetching, despite facing challenges, effective solutions were applied to achieve the project goals. Further improvements can be made to enhance the user experience and performance.