**HACKATHON 3**

**DAY 1 TASK : Laying The Foundation For Your Market Place Journey**

**What type o market place are you building?**

General E-commerce (selling various products online) General E-commerce Market place

**• Products Categories**

1) Furniture (chairs)

2) Home Decoration (vase)

3) Lighting (Lamp)

4) Living Room (Sofa)

• **Target Audience**

1. Office Managers
2. Hospitality Industry
3. Home decorator
4. Interior designers and decorator

**What problem does you market place aim to solve?**

• An online market place provides access to a diverse selection of products regardless of consumer location

. • Enhanced product visualization tools such as high-quality images, detailed description allowing for efficient organization and presentation aim of products in order to meet diverse consumers

**Aim**: These websites focus on products that enhance interior design and provide functional furniture for homes or offices, interior designers, hospitality industry .

**Unique Feature:**

• Fast delivery

• Discounts

• Authentic Products

**What are main entities and their relationship?**

**[Product]**

.ID .Name

.Price

.Dimension

!

!

**[Order]** -- **> [Customer]**

-Order ID -Name

-Customer ID -Quality

-Product ID -Contact Info

!

!

**[Payment]**

-Payment Id

-Payment Date

-Payment Method

**Products Schema: [**

“Product name”: Product name;

“Type “ : string;

“price” : number;

“Dimension” : Text ;

]

**Order Data** [

“Order ID”: order Id;

“ Customer ID”: customer Id;

“Total Amount “: Total price of order;

“order status”: current state;

“Product ID : product Id ;

“Quantity” ; Quantity ;

]

**Customer Data** [

“Customer Name” : Customer Name ;

Customer ID”: customer Id;

“Email “: email;

“Mobile no”: Mobile No ;

“Address”: Address ;

]

**Payment Data** [

“Payment ID”: Payment Id;

“Payment Date”: Payment date ;

“Payment Method “: Cash /credit card ;

]

**Sanity:**

**Product Data 1**{

“Product Id”: 01 ;

“Product Name: Sofa;

“Price”: $ 980

“Dimension”: Height, Width , Depth

}

**Product Data 2**{

“Product Id”: 02 ;

“Product Name: The Dandy Chairs;

“Price”: $ 250

“Dimension”: Height(110cm), Width(75cm) , Depth(50cm)

}

**Product Data 3**{

“Product Id”: 03 ;

“Product Name: The silky Vase ;

“Price”: $ 125

“Dimension”: Height, Width , Depth

}

**Product Data 4**{

“Product Id”: 04 ;

“Product Name: Rustic vase set;

“Price”: $ 155

“Dimension”: Height, Width , Depth

}

**Order Data** [

“Order ID”: order ID;

“Customer ID” : rabia Khan;

“Total Amount “: $ 980;

“Product ID” : Sofa ;

“Quantity” ; 01 ;

]

**Customer Data** [

“Customer Name”: rabia Khan;

“ Customer ID”:rabia Khan;

“Email “: [RabiaKhan@gmail.com](mailto:RabiaKhan@gmail.com);

“Mobile no”: 0345-2429028;

“Address”: Johar Town;

]

**Payment Data** [

“Payment ID”: #407;

“Payment Date”: 30-01-2025 ;

“Payment Method “: Online;

“Transfer Payment”: Bank

]

**2 TASK TECHNICAL FOUNDATION FURNITURE E COMMERCE WEBSITE**

**Technical Feature**

**1 FRONTEND** : Design the following interactive pages Next.js with typescript ,tailwind:

▪ **Home**: Products and Categories

▪ **Shop** (new ceramics /popular )

**products** :Displays all furniture products with filters

▪ **About** :Describe the brand and its story

▪ **Contact** : Includes a contact form ,location map 2 Live u

▪ Inventory changes for products availability

▪ On order status 3 Integration of APIs

▪ Payment gateway

▪ Shipping/Tracking API

**FRONTENED**

**THIRD API**

**SANITY**

**PAYMENT GATEWAY**

**SHIPPING TRACKING API**

**ORDER DATA**

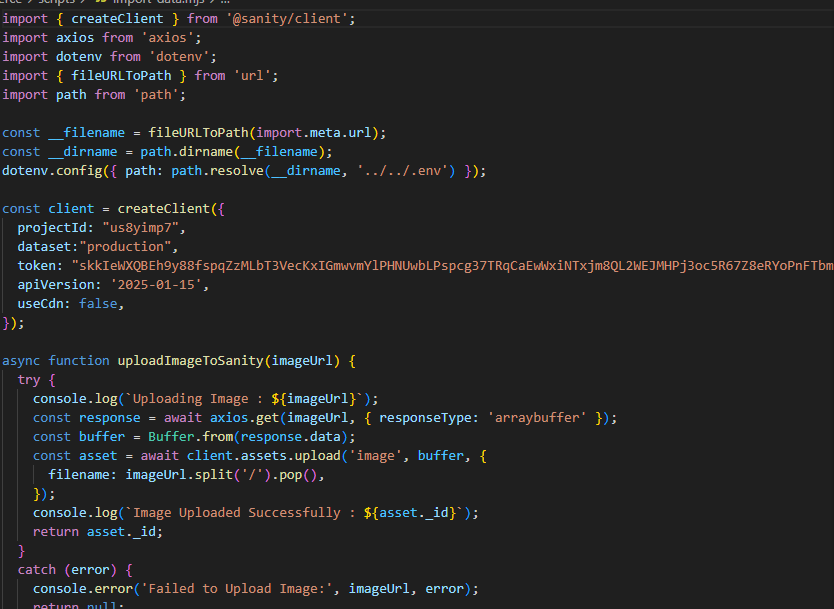


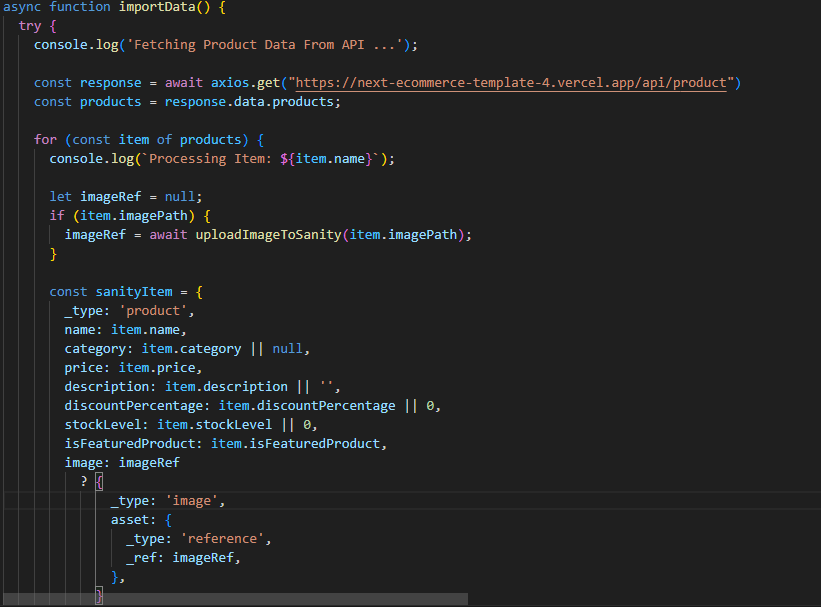


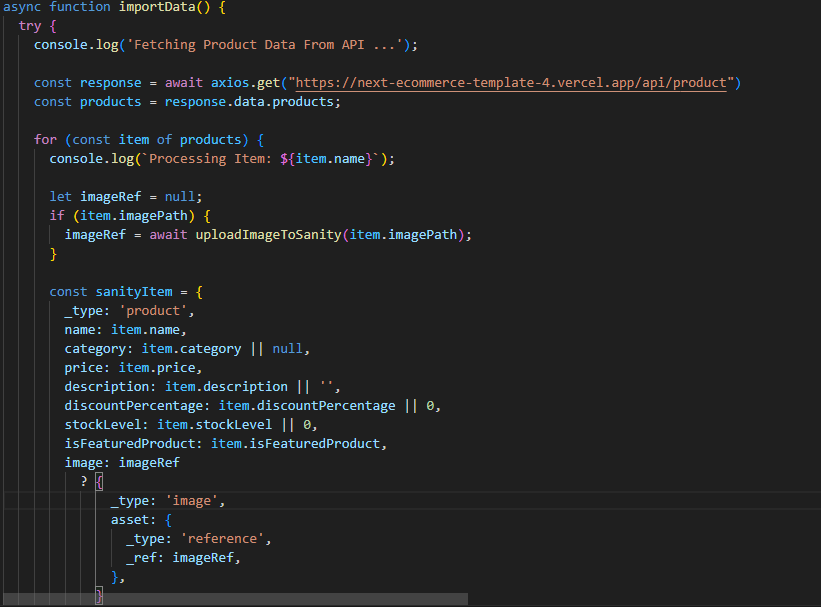
**PRODUCT SCHEMA**

**Day 3 –API Integration and Data Migration Furniture E-commerce Website**

**Overview**: The primary focus of Day 3 was on integration APIs, specifically with Sanity CMS, using GROQ, queries and updating the schema to support dynamic data requirements







Here’s a summary of the tasks accomplished:

**Key Objectives**:

1. Establish a connection with sanity CMS for managing product data efficiently
2. Implement GROQ queries to fetch structured data dynamically from Sanity.
3. Update the Sanity schema to align with the evolving requirements of the API.
4. Migrate product data from an external API to Sanity CMS, including image uploads.
5. Test and validate the integration to ensure smooth data flow between the backend and frontend

**Objective**: Migrate product data, including images, from an external API to Sanity CMS Steps taken to migrate API into sanity

1. **Fetch Product Data** :

• Used Anxios to retrieve product data from an external API.

1. **Upload Images to Sanity:**

• Uploaded product images using the client. Assets.upload method

• Ensured that each image was linked to its respective product.

**iii. Create Product Document in Sanity:**

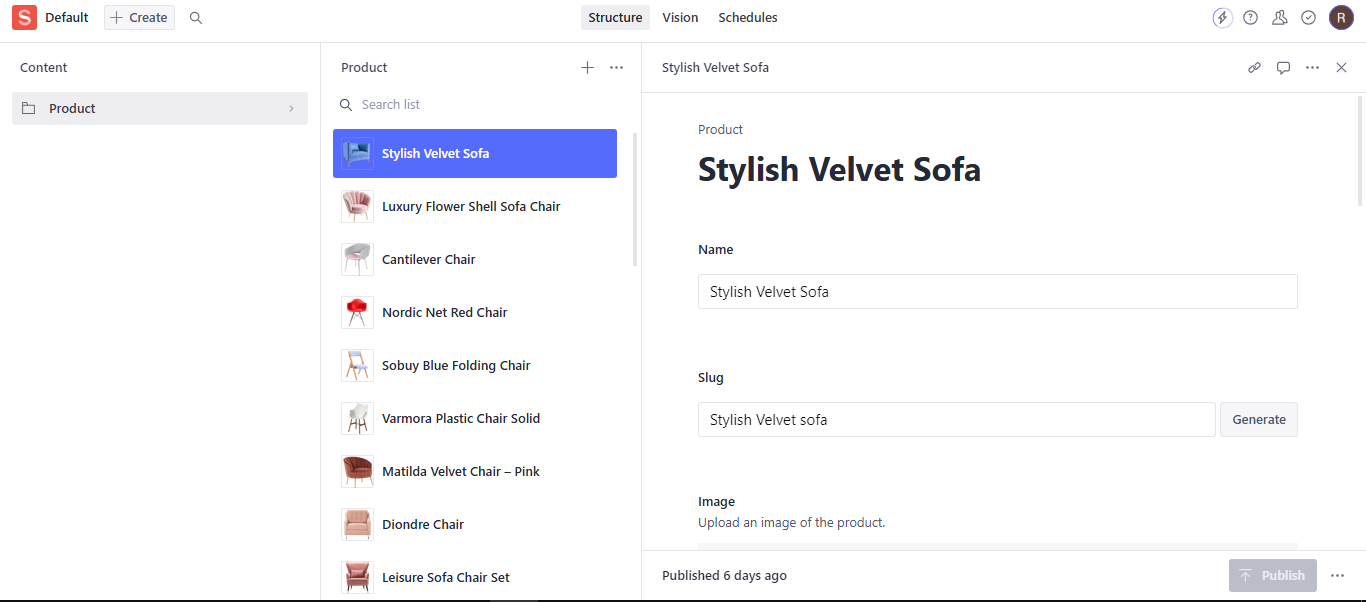
Iterated over the fetched data and created a corresponding product document in sanity using the client.create () method

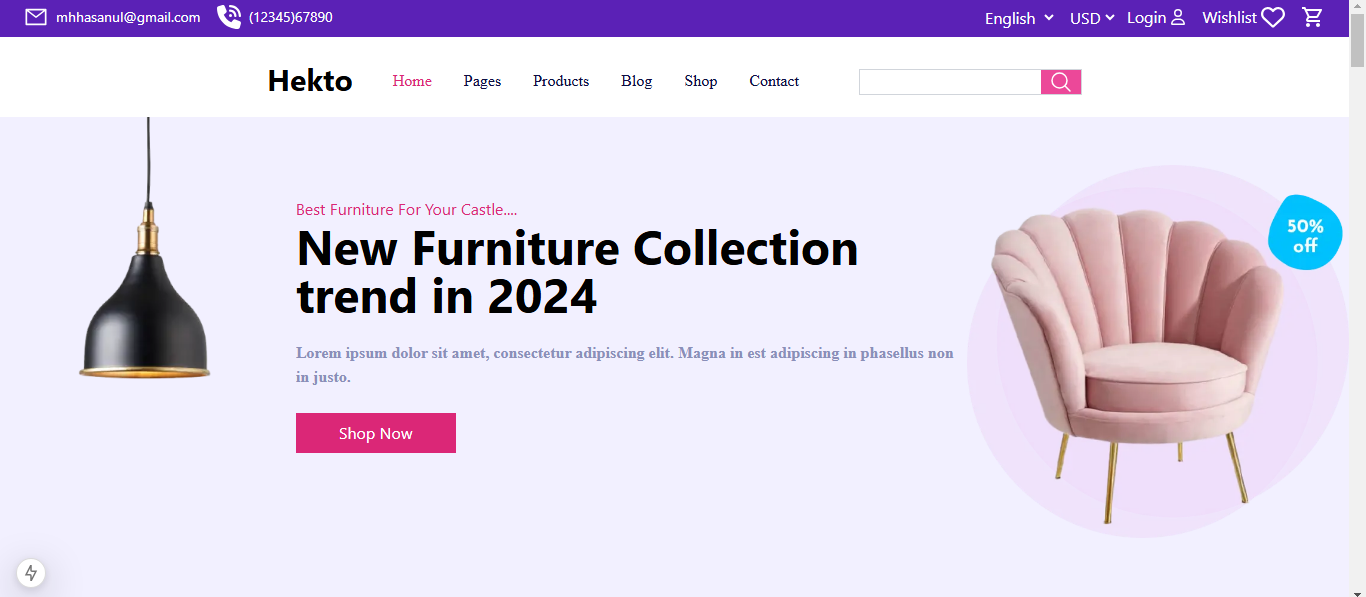
**2. Sanity CMS Integration:**

• Connection Setup

• Integrated the @Sanity/client library into the project to enable communication with the Sanity CMS.

• Configured the client using projects-specific credentials, including the project Id, dataset and API Version





**Day 4–Dynamic Routing and Displaying sanity Data :**

**•** Dynamic Routing

• Filter Functionality Implementation

• Top Picks Functionality

**Individual product detail pages with accurate routing and data rendering**

Day 5 –Testing and Backend Refinement Report :

**Furniture E-commerce Web site Overview:**

On Day 5,the focus was on testing core functionalities, refining the backend, and optimizing the performance of the e-commerce website.

Key tasks included testing features like cart functionality, checkout and responsiveness, alongside implementing error handling and performance improvements. **Functional Deliverables:**

Screenshots: Logs or Reports from Testing Tools: Lighthouse Report: Evaluate performance, accessibility and SEO

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CASE ID | Test Scenario | Test Steps | Expected Result | Actual Result | Status | Severity Level |
| TC 001 | Login Functionality | Enter valid credentials and click login | User should be logged in success fully | User logged in | Pass | Medium |
| TC 002 | Add to Cart | Select a product and click add to cart | Product should be added to the cart | Error message displayed | Pass | Medium |
| TC 003 | Checkout Process | Complete checkout with valid payment details | Order should be placed success fully | Product added | Pass | High |
| TC004 | Search Functionality | Enter a product name in search bar | Relevant products should be displayed | Order placed | Pass | Low |
| TC 005 | Validate add to cart functionality | Add product to cart | Product is added to the cart and cart content updates | Products displayed | Pass | Medium |
| TC 006 | Test Cart display | Cart should display all added products | Cart should display all added products | Product is added to the cart | Pass | Low |
| TC 007 | Test cart item quantity functionality | Increase  /decrease product quantity | Cart should update the quantity of the product | Cart displays products | Pass | Medium |
| TC 008 | Test product removal from cart | Remove product from cart | Product is removed from cart | Quantity updated | Pass | Medium |
| TC 09 | Test error handling during checkout | Enter invalid payment details | Error message for invalid payment should appear | Product removed success fully | Pass | High |
| TC 010 | Test error handling during checkout | Open product page | Page should load within three seconds | Error message displayed | Pass | Low |

CSV FORMATS :