Hackathon Day 2

Marketplace e-commerce website

Frontend Requirements

- User Interface (UI):
- - Modern, minimalistic, and visually appealing
- - Emphasis on product imagery and ease of navigation
- - Responsive for all screen sizes

- Core Pages:
- - Home Page: Highlights featured products, seasonal collections, and promotional offers
- - Category Page: Categorized products (Living Room, Kitchen, Décor Accessories)
- - Product Details Page: Images, descriptions, price, dimensions, and materials
- - Cart Page: Overview of selected products, subtotal, shipping, and total amount
- - Checkout Page: Collect user information and payment
- - Order Confirmation Page: Order details, confirmation, and tracking

- Additional Features:
- - Search with filters (price, material, color, style)
- - User account section for order history and wishlist
- - Live Chat for customer support

Backend Requirements

- Sanity CMS:
- - Centralized product, order, and user data management

- Key Schemas:
- - Product Schema: Stores product details (name, price, category, stock, dimensions)
- - Customer Schema: Stores customer data (name, email, address, orders)
- - Order Schema: Stores order details (products, quantity, total price, status, tracking)

- Features Handled by Sanity:
- - Real-time inventory updates
- - Seamless order management
- - Centralized user and product data

API Requirements

- Endpoints Overview:
- - GET /products: Fetch product list
- - GET /products/:id: Fetch details of a specific product
- - POST /orders: Save a new order
- - GET /orders/:id: Fetch order details
- - GET /shipment/:trackingId: Get shipment tracking info

- API Examples:
- - GET /products: Returns list of products with name, price, and stock
- - POST /orders: Saves an order with customer ID, items, and total price
- - GET /shipment/:trackingId: Returns shipment status and ETA

System Architecture

- Components:
- - Frontend: Built with Next.js, providing a dynamic and responsive interface
- - Backend: Sanity CMS as the content hub
- - Third-party APIs: For payment gateway, shipment tracking, and analytics

- Data Flow:
- - User interacts with the frontend to browse products
- - Frontend fetches data from Sanity CMS via APIs
- - Orders are recorded in Sanity and tracked through APIs

Key User Workflows

- Browsing Products:
- - Products fetched from /products endpoint

- Placing an Order:
- - User adds items to cart → Proceeds to checkout → Order details saved via /orders

- Tracking an Order:
- - User retrieves shipment details via /shipment/:trackingId

Additional Features

- - Wishlist Management: Save favorite items for future purchases
- - Reviews & Ratings: Customers can rate and review products
- - Recommendations: Display related or popular items dynamically

Product Schema (Sanity CMS)

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'category', type: 'string', title: 'Category' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'stock', type: 'number', title: 'Stock Quantity' },
    { name: 'images', type: 'array', of: [{ type: 'image' }], title: 'Images' },
    { name: 'dimensions', type: 'string', title: 'Dimensions' },
    { name: 'material', type: 'string', title: 'Material' },
  ],
};
```

Industry Best Practices

- - Plan before implementation
- - Focus on scalability (modular schemas, optimized APIs)
- - Leverage Sanity CMS for real-time updates
- - Prioritize user experience with intuitive navigation

Customer Schema (Sanity CMS)

- export default {
- name: 'customer',
- type: 'document',
- fields: [
- { name: 'name', type: 'string', title: 'Customer Name' },
- { name: 'email', type: 'string', title: 'Email' },
- { name: 'address', type: 'text', title: 'Address' },
- { name: 'orders', type: 'array', of: [{ type: 'reference', to: [{ type: 'order' }] }] },
-],
- };

API Endpoints Example

- // Fetch all products
- app.get('/products', async (req, res) => {
- const products = await sanityClient.fetch('*[_type == "product"]'
);
- res.json(products);
- });

- // Fetch a single product by ID
- app.get('/products/:id', async (req, res) => {
- const productId = req.params.id;
- const product = await sanityClient.fetch(`*[_type == "product"
&& _id == $id][0]`, { id: productId });
- res.json(product);
- });