Hina Alvi

# Hackathon Day 2

# TECHNICAL FOUNDATION

## Technical Requirements:

### 1 User-Friendly Interface

- **Intuitive Design:** A sleek, modern layout with easy-to-navigate menus for effortless browsing.
- **Category Organization:** Clear sections like Living Room, Bedroom, Office, etc., make finding products simple.
- **Advanced Filtering Options:** Sort products by price, material, size, color, and popularity for a personalized shopping experience.
- **Quick Access Features**: Sticky navigation bars and footers provide instant access to essential pages**.**

### 2. Responsive Design

- Device Adaptability: Fully responsive design ensures seamless performance across mobiles, tablets, desktops, and laptops.
- Mobile Optimization: Easy-to-tap buttons, legible fonts, and spacious layouts ensure a smooth mobile experience.
- Dynamic Scaling: Images and content adjust automatically to fit the user’s device perfectly.

### 3. Essential Pages

- Home Page: Highlights featured products, ongoing promotions, and quick links to categories.
- Product Listing Page: Displays all items with filters, images, and pricing details.
- Product Details Page: Offers in-depth product information, including images, dimensions, and prices.
- Cart Page: Allows users to review selected items, update quantities, and prepare for checkout.
- Checkout Page: Guides customers through a step-by-step checkout process.
- Order Confirmation Page: Confirms orders with detailed summaries and tracking information.

### 4. Technical Implementation

- Framework: Built using Next.js for fast and efficient performance.
- CMS: Sanity is used to dynamically manage content.
- UI Framework: Tailwind CSS ensures a consistent, visually appealing design.
- Component Library: Shadcn components add functionality and quality.
- Cart Management: Combines local storage for offline use and server-side integration for scalability.

## 5. Responsive Cart and Checkout Flow

- Mobile-Friendly Checkout: Optimized for smaller screens with an intuitive design.
- User-Focused Steps: Simplified, step-by-step checkout minimizes effort and confusion.
- Smooth Payment Process: Payment integration works seamlessly across all devices.
- Progress Indicators: Visual progress bars guide users from the cart to order confirmation.

## 6. Secure Payment Integration

- Trusted Gateways: Integrates with leading payment platforms like Stripe and PayPal.
- SSL Encryption: Secures all transactions with robust encryption to protect user data.
- Flexible Payment Options: Supports multiple payment methods for convenience.
- Reliable Processing: Ensures hassle-free and secure transactions for every purchase

# Backend Requirements for Comforty

Powered by Sanity CMS for Seamless Data Management

## Sanity CMS for Data Management

**Purpose:**

To provide a centralized database for managing all aspects of the Comforty marketplace.

## Key Features:

## Furniture Items:

**Store details such as:**

Name

Description

Price

Category (e.g., Chairs, Sofas, Tables)

Material

Dimensions

High-quality images

Ensure easy updates to product information.

User Information:

## Manage customer data, including:

Full names

Email addresses

Delivery addresses

Order histories for personalized user experience.

Orders:

## Track order statuses efficiently:

Pending

Shipped

Delivered

Support order management for smooth customer service.

Categories:

## Organize furniture into intuitive categories like:

Living Room

Bedroom

Office

Outdoor

Simplify navigation and filtering for users.

Sanity Schemas:

## Design clear and well-structured schemas to ensure efficient data management.

Allow easy addition, updating, or removal of data without disrupting other parts of the system.

Provide flexibility for future updates or feature enhancements.

# *Third-Party API Integrations*

To ensure dynamic functionality and exceptional customer experience, Comforty integrates essential external APIs:

1. Payment Gateways

Secure Transactions:

Integration with trusted platforms like Stripe or PayPal to handle payments.

## Quick and Reliable:

Ensure smooth transaction processing with robust security protocols.

## Multiple Payment Options:

Enable customers to choose from debit/credit cards, wallets, or online banking.

2. Shipment Tracking APIs

### Real-Time Tracking:

Provide customers with up-to-date shipment statuses.

### Transparency:

Keep users informed about order locations and estimated delivery times.

Reliable Logistics:

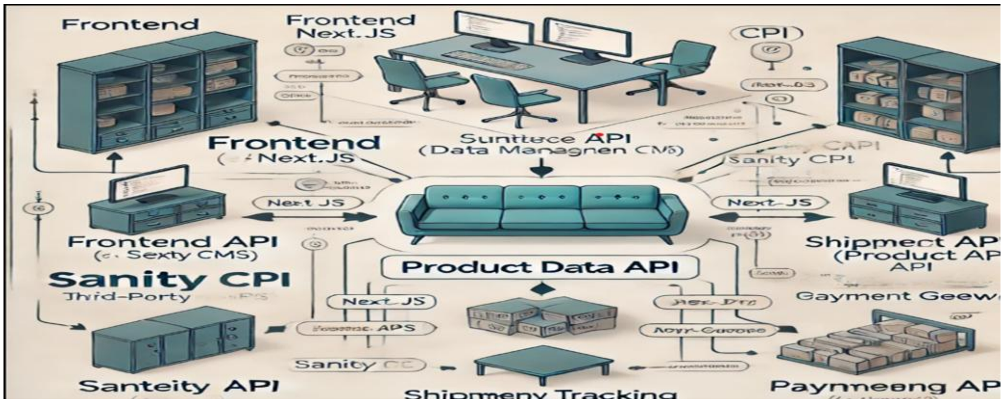
Partner with reputable shipping providers for efficient deliveries.

□

### *1. User Registration:*

- The user signs up on the frontend.

# 2)Design System Architecture



## Data Flow and Workflows

Comforty Backend Processes and Functionality

### *1. User Registration*

### Data Storage:

User registration details (e.g., name, email, and password) are securely stored in Sanity CMS.

### Confirmation:

A success message is displayed to the user upon successful registration, ensuring a smooth onboarding experience.

### 2. Product Browsing

When a user visits the Furniture Listing Page:

The frontend sends a request to fetch furniture data via APIs connected to Sanity CMS.

Sanity CMS responds with updated furniture information, which is dynamically displayed on the website for users to browse.

### 3. Order Placement

Adding Items to the Cart:

Users add desired furniture items to the cart.

### Placing the Order:

Upon checkout, order details (products, user info, and payment status) are sent to Sanity CMS.

The order is securely recorded in the database, and a confirmation is provided to the user.

### 4. Shipment Tracking

Real-Time Tracking:

Shipment details are fetched from a Third-Party API connected to the logistics provider.

These details are displayed on the user’s "Order Status" page, keeping them informed about their delivery in real time.

### 5. Payment Processing

Secure Checkout:

Payment information is processed via trusted gateways like Stripe or PayPal.

Once payment is completed:

A confirmation message is displayed to the user.

Payment status is updated in Sanity CMS for backend order tracking.

### 6. Adding Furniture to the Cart

**User Action:**

Users select furniture items and click "Add to Cart."

**Temporary Storage:**

Cart details are stored locally (e.g., in local storage or state management systems).

**Checkout Process:**

During checkout, cart data is sent to the backend for order finalization.

### 7. Tracking an Order

User Interaction:

Users click the "Track Order" button.

**API Call:**

The frontend calls the /shipment/{orderId} API.

**Display Details:**

Shipment information from the Third-Party Tracking API is fetched and displayed on the user's "Order Status" page.

## 8. Payment Integration

Checkout Process:

During checkout, secure payment details are sent to the integrated Payment Gateway.

**Confirmation:**

Once payment is verified, the order status is updated in the backend, and the user is notified.

## 9. Finalizing Orders

Review and Confirm:

Users review their cart and confirm the order.

Backend Communication:

The frontend sends order details (furniture items, user information, and payment status) to the /orders API.

**Order Management:**

The order is stored in Sanity CMS, and a confirmation email or notification is sent to the customer.

## 3) API ENDPOINTS

| ENDPOINT | METHOD | PURPOSE | RESPONSE EXAMPLE |
|---|---|---|---|
| /products | GET | **Fetches all available products.** | [{"id":1,"name":"Sofa","price":15999,"stock":20}] |
| /products/{id} | GET | **Fetches details for a specific product**. | {"id":1,"name":"Sofa","price":15999,"description":"Comfortable 3-seater sofa"} |
| /orders | POST | **Creates a new order.** | {"orderId":101,"status":"Order Placed","ETA":"3-5 days"} |
| /shipment/{orderId} | GET | **Fetches real-time order tracking.** | {"shipmentId":456,"status":"In Transit","expectedDelivery":"2 days"} |

# Entities

### 1. Products

- **Fields**: `id`, `name`, `description`, `price`, `stock`, `category`, `image`

### 2. Orders

- **Fields**: `orderId`, `customerName`, `contactInfo`, `address`, `items`, `totalAmount`, `paymentStatus`

### 3. Shipment

- **Fields**: `shipmentId`, `orderId`, `status`, `expectedDelivery`

### 4. Customers

- **Fields**: `customerId`, `name`, `contactInfo`, `address`, `orderHistory`