Furniture E-Commerce Website System Design

1. Frontend Technologies

Technologies Used:

- Next.js: Enables server-side rendering for faster page loads and better SEO.

- Tailwind CSS: Provides utility-first styling for responsive and modern UI design.

Key Features:

- User-Friendly Interface: Smooth navigation for browsing products and placing orders.

- Responsive Design: Optimized layouts for both desktop and mobile devices.

Implemented Pages:

- Home: Showcases featured products, categories, and ongoing promotions.

- Product Listing: Displays products filtered by categories (e.g., Living Room, Bedroom).

- Product Details: Provides detailed product information with images and specifications.

- Cart: Allows users to view, add, or remove items before purchase.

- Checkout: Collects customer shipping details and payment information.

- Order Confirmation: Displays order summary and confirmation message after successful purchase.

Frontend Functionalities:

- Fetch product and category data from the backend (Sanity CMS) using GROQ (Graph-Relational Object Queries).

- Implement dynamic routing for product details pages (/products/:id).

2. Backend (Sanity CMS)

Furniture E-Commerce Website System Design

Objective:

Serve as the Content Management System (CMS) for managing products, categories, orders, and customer interactions.

Schema Design:

- Products: Fields - name, price, stock, image, category, description.
- Categories: Fields - name, description, associated products.
- Orders: Fields - customer details, product IDs, quantities, payment status, order status.

Key Features:

- Real-time Data Management: Immediate updates reflected on the frontend.
- Efficient Data Handling: Smooth storage and retrieval of data for fast frontend interactions.

3. Interaction Between Frontend and Backend

Fetching Data:

- The frontend (Next.js) uses API requests to retrieve product and category data from Sanity CMS.
- Example GROQ Query:

```
*[_type == "product"] {
  _id,
  name,
  price,
  category->name,
  images
}
```

Submitting Orders:

- When a user places an order, the frontend sends a POST request to Sanity CMS.

- The backend stores order details such as customer information, purchased products, and order status.

4. API Requirements

Objective:

Provide efficient and clear API endpoints for seamless frontend-backend communication.

API Endpoints:

- GET /products: Retrieve all available products. Response: Product ID, name, price, stock, and images.

- GET /categories: Fetch all product categories. Response: Category ID, name, and description.

- POST /orders: Submit a new order. Payload: Customer information, selected product details, and payment status.

- GET /shipment: Track shipment status via a third-party API. Response: Shipment ID, order ID, current status, and estimated delivery date.

5. System Architecture

Objective:

Outline the interaction between system components for efficient workflow.

Workflow:

1. User Interaction: Users browse products, add items to the cart, and proceed to checkout.

2. Order Processing: Order details are sent to the backend (Sanity CMS) through API calls.

3. Payment Verification: Payment methods verify transactions and confirm orders.

4. Shipment Initiation: Orders are passed to the shipment team for dispatch.

5. Delivery Tracking: Shipment status is updated and visible to the user until final delivery.

This architecture ensures a smooth user experience from browsing to final product delivery.