Technical Documentation for Furniture Marketplace E-commerce Website

---

1. System Architecture Document

Overview

- **Objective:** Describe the overall design and interaction between components of the Furniture Marketplace E-commerce website.
- **Components:**
    - **Frontend:** Built with **Next.js 15** and **Tailwind CSS** for responsive and modern UI design.
    - **Backend:** Managed with **Sanity CMS**, providing a flexible and scalable backend for content management.
    - **Database:** Utilizes Sanity's built-in data storage capabilities.
    - **Authentication:** Implements [Auth Method, e.g., OAuth2, JWT] for secure user authentication.
    - **Third-party Integrations:** Integrates payment gateways like Stripe and shipping services like FedEx or DHL.

Architecture Diagram

- Diagram showing components such as:
    - Frontend (Next.js 15)
    - Backend (Sanity CMS)
    - Database (Sanity)
    - External APIs (Payment and Shipping)
    - Authentication Service

Interaction Between Components

1. **User Request:** User interacts with the frontend to browse furniture or place orders.
2. **Frontend Logic:** Frontend processes input and sends API requests to the backend managed by Sanity.
3. **Backend Processing:** Sanity handles the request, queries data, and manages content updates.
4. **Response Delivery:** Data is sent back to the frontend for display or user interaction.

---

2. API Specification Document

Base URL:

https://api.furnituremarketplace.com/v1

Endpoints

Authentication

- **POST** `/auth/login`

Request Body:

```
{
 "email": "user@example.com",
 "password": "securepassword"
}
```

-

Response:

```
{
 "token": "jwt_token",
 "user": {
  "id": "123",
  "name": "John Doe"
 }
}
```

-

Products

- **GET** `/products`
  - **Query Parameters:**
    - `category` (optional)
    - `price_min` (optional)
    - `price_max` (optional)

Response:

```
[
 {
  "id": "1",
  "name": "Modern Sofa",
  "price": 599.99,
  "category": "Living Room"
```

```
 }
]
```

-

Orders

- **POST** `/orders`

Request Body:

```
{
 "userId": "123",
 "products": [
  { "productId": "1", "quantity": 2 }
 ],
 "shippingAddress": "123 Main St, Springfield, USA"
}
```

-

Response:

```
{
 "orderId": "456",
 "status": "processing"
}
```

-

---

3. Workflow Diagram

User Journey Workflow

1. User visits the website and browses furniture.
2. User adds items to the cart.
3. User creates an account or logs in.
4. User places an order with a payment and shipping address.
5. Backend processes the order and updates the database via Sanity.
6. User receives order confirmation and tracking details.

Diagram Example:

- Use tools like Lucidchart, Figma, or Draw.io to create a flowchart for this workflow.

---

4. Data Schema Design

Users Table

Field	Type	Description
`id`	UUID	Primary key
`name`	String	Full name of the user
`email`	String	Unique email address
`password`	String	Hashed password
`role`	Enum	User role (`buyer`, `seller`)

Products Table

Field	Type	Description
`id`	UUID	Primary key
`name`	String	Name of the product
`price`	Float	Price of the product
`category`	String	Product category
`image`	String	URL of the product image

---

5. Technical Roadmap

Phase 1: Planning and Setup

- Define requirements and architecture.
- Set up project repository with Next.js 15 and Tailwind CSS.
- Configure Sanity CMS for backend management.

Phase 2: Development

- Develop core features:
  - User authentication using [Auth Library].
  - Product listing and filtering.
  - Cart and checkout functionality.
- Integrate third-party APIs (e.g., Stripe for payments).

Phase 3: Testing and Optimization

- Conduct unit and integration testing.
- Optimize API calls and Tailwind styling for performance.
- Perform cross-browser and device testing.

Phase 4: Deployment and Maintenance

- Deploy the application on Vercel.
- Set up monitoring for performance and error tracking.
- Regularly update content through Sanity CMS.

Prepared By Khalid Hussain
RN: 00372206