Furniture E-Commerce Marketplace

The Technical Documentation

Hackathon 3: Day 2

Introduction

This document provides a detailed overview of the technical structure for a furniture e-commerce marketplace. The primary objective is to develop a user-friendly platform where customers can explore furniture, place orders, and track deliveries seamlessly. Leveraging technologies like Next.js, Sanity CMS, and third-party APIs, this project focuses on scalability, security, and an exceptional user experience.

Prepared By: Muhammad Yousuf

Table of Contents

- Checkout
- Order Confirmation
- Shipment Tracking
- Order History

2. System Workflow

- Workflow Steps
- Workflow Diagram

3. Unique Features

- Category-Based Filtering
- Real-Time Shipment Tracking
- Secure Payments
- Customizable Furniture Options
- Mobile Optimization
- Order History

4. API Specifications

- User Registration
- Fetch Categories
- Fetch Products by Category
- Place Order
- Track Shipment
- Process Payment

5. Sanity CMS Schemas

Product Schema

Order Schema

User Schema

6. FAQs

Overview of the Marketplace

The Furniture E-Commerce Marketplace is a comprehensive platform designed to revolutionize the way users purchase furniture online.

It provides a seamless shopping experience, enabling customers to browse through an extensive range of furniture, customize products, and securely place orders.

This marketplace leverages cutting-edge technologies like **Next.js**, **Sanity CMS**, and various third-party APIs to ensure scalability, reliability, and an exceptional user experience.

With features like category-based filtering, real-time shipment tracking, secure payment gateways, and mobile optimization, the platform covers to a wide audience, including individuals and businesses.

The system architecture is built to handle high traffic, with a robust backend that manages user data, product information, and order processing.

The integration of modern tools like Tailwind CSS for styling and TypeScript for type safety ensures that the development process is efficient and the codebase is maintainable.

By combining advanced technology with user-friendly design, the Furniture E-Commerce Marketplace aims to set a new standard for online furniture shopping.

Tech Stack

Front-End

- **Framework**: Next.js ensures fast performance with server-side rendering.
- **Styling**: Tailwind CSS for creating modern and responsive designs.
- **TypeScript**: Provides type safety and enhances developer productivity.

Key Features:

  - Dynamic routing for category and product pages.
  - Cart functionality with state management.
  - Fully responsive design for optimal viewing on all devices.

Back-End

- **Sanity CMS**: Manages data for products, users, and orders.

Third-Party APIs:

  - **Shipment Tracking**: ShipEngine API for real-time updates.
  - **Payment Processing**: Supports multiple gateways like 2Checkout, PayFast, and QuickPay.

Third-Party APIs

- **ShipEngine API**: Enables real-time shipment tracking.
- **Payment Gateways**: Ensures secure and flexible payment options.

Development Tools

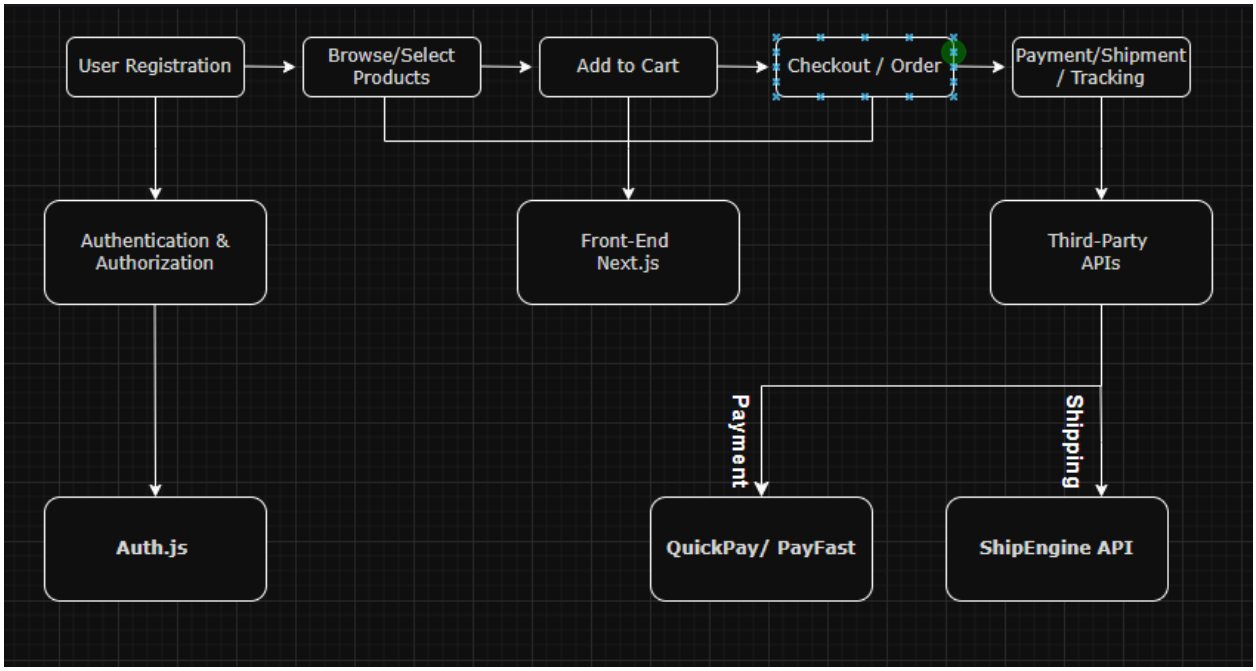
- **GitHub**: Version control for team collaboration.
- **Postman**: API testing and validation.
- **Lucidchart/Excalidraw**: Visual tools for workflow and architecture diagrams.

Essential Pages

1. **Home Page**: Highlights categories and featured products.
2. **Product Listing**: Displays products by category or all products.
3. **Product Details**: Shows detailed product information, including images and pricing.
4. **Cart**: Allows users to review and modify their selected items.
5. **Checkout**: Collects user information and processes payments.
6. **Order Confirmation**: Provides a summary of the order and confirmation message.
7. **Shipment Tracking**: Displays real-time shipment status.
8. **Order History**: Allows users to view past orders by entering their phone number or orderId, for those who did not want to signup. Registered user will have the professional dashboard, that displays the previous, current orders, wishlist etc.

System Workflow

1. **User Registration**: Captures user details in Sanity CMS and sends a confirmation response.
2. **Browsing Products**: Retrieves categories and products from Sanity CMS.
3. **Placing Orders**: Saves order details in Sanity CMS and processes payments securely.
4. **Shipment Tracking**: Fetches tracking updates via ShipEngine API.
5. **Payment Processing**: Manages secure transactions through integrated gateways.



Unique Features

Category-Based Filtering: Users can explore products by specific categories like Living Room or Bedroom.

Real-Time Shipment Tracking: Provides up-to-date order status through ShipEngine API.

Secure Payments: Multiple gateways ensure safe and reliable transactions.

Customizable Furniture Options: Certain products allow customization of colors, materials, and sizes.

Mobile Optimization: Fully responsive design ensures a seamless experience across devices.

Order History: Simplifies tracking of past orders with comprehensive details.

API Specifications

User Registration

- **Endpoint**: `/register`
- **Method**: POST
- **Description**: Captures user details and saves them in Sanity CMS.

Payload Example:

```
{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "password": "securepassword",
}
```

Response Example:

```
{

  "message": "Registration successful",

  "userId": "12345"

}
```

Fetch Categories

- **Endpoint**: `/categories`
- **Method**: GET

- **Description**: Retrieves all available product categories from Sanity CMS.

Response Example:

```
[
  { "id": "1", "name": "Living Room" },
  { "id": "2", "name": "Bedroom" }

]
```

Place Order

- **Endpoint**: `/orders`
- **Method**: POST
- **Description**: Saves order details in Sanity CMS.

Payload Example:

```
{

  "userId": "10011",

  "orderItems": [

    { "productId": "201", "quantity": 1 },

    { "productId": "302", "quantity": 2 }

  ],

  "totalAmount": 2990

}
```

Response Example:

```
{

  "message": "Your Order placed successfully!",

  "orderId": "79801"

}
```

Track Shipment

- **Endpoint**: `/shipment/{orderId}`
- **Method**: GET
- **Description**: Fetches shipment tracking details using ShipEngine API.

Response Example:

```
{  "shipmentId": "SE123",

  "status": "In Transit",

  "estimatedDelivery": "2025-01-18" }
```

Process Payment

- **Endpoint**: `/payment`
- **Method**: POST
- **Description**: Processes payments securely via the selected gateway.

Payload Example:

```
{

  "orderId": "78901",

  "amount": 2100,

  "paymentMethod": "PayFast"

}
```

Response Example:

```
{

  "message": "Payment successful",

  "transactionId": "TX456789"

}
```

Sanity CMS Schemas

Product Schema

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

Category Schema

```
export default {
  name: 'category',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Category Name' },
    { name: 'slug', type: 'slug', title: 'Slug', options: { source: 'name', maxLength: 96 } },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'image', type: 'image', title: 'Category Image' }
  ]
};
```

Order Schema

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'userId', type: 'reference', to: [{ type: 'user' }], title: 'User' },
    { name: 'orderItems', type: 'array', of: [{ type: 'orderItem' }], title: 'Order Items' },
    { name: 'totalAmount', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status', options: { list: ['Pending', 'Completed', 'Shipped'] } }
  ]
};
```

User Schema

```
export default {
  name: 'user',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Full Name' },
    { name: 'email', type: 'string', title: 'Email Address' },
    { name: 'password', type: 'string', title: 'Password' }
  ]
};
```

FAQs

Q1: What is the purpose of this documentation?
A: This documentation serves as a comprehensive guide for developers, stakeholders, and contributors to understand and implement the Furniture Marketplace system, including its architecture, APIs, and workflows.

Q7: What are the core features of the marketplace?
A:

- **User Management**: Role-based access for admins, vendors, and customers.
- **Product Listings**: CRUD operations for furniture items.
- **Search and Filter**: Advanced search with category and price filters.
- **Order Management**: End-to-end order tracking.
- **Payment Integration**: Secure payment gateways.

Q8: How is the category schema structured?
A: The category schema includes:

- `title`: Name of the category.
- `slug`: Unique identifier for routing.
- `description`: A brief overview.
- `image`: Category banner.
- `parentCategory`: Optional field for nested categories.

Q9: How are APIs organized?
A: APIs are RESTful and grouped by functionality:

- **Auth**: Login, registration, and user management.
- **Products**: Endpoints for product CRUD operations.
- **Orders**: Create, update, and track orders.
- **Categories**: Manage and retrieve category data.

Q3: What technologies are used in the Furniture Marketplace?
A: The platform uses [e.g., Next.js, Tailwind CSS, Sanity CMS, Node.js, MongoDB] for a robust, scalable, and user-friendly experience.

Q4: How do I set up the project locally?
A: Follow these steps:

1. Clone the repository: `git clone <repository-url>`
2. Install dependencies: `npm install`
3. Set up environment variables in a `.env` file (refer to the `.env.example` provided).
4. Start the development server: `npm run dev`.

Q5: What is the structure of the project?
A: The project follows a modular structure:

- **Frontend**: Built with Next.js, Typescript and Tailwind CSS.
- **Backend**: Managed via Sanity CMS.
- **APIs**: Using Third-Party APIs for Shipment, Tracking and Payment proceeding.