Technical Report: Furniture E-Commerce Website Development

1. Introduction

This report provides a comprehensive summary of the steps taken to build and integrate components for the furniture e-commerce website. It outlines the challenges faced during development and the solutions implemented, along with best practices followed to ensure scalability, security, and performance.

2. Technology Stack

Frontend:

- **Framework:** Next.js (for server-side rendering and performance optimization)
- **UI Components:** ShadCN + Tailwind CSS (for a modern, responsive interface)
- **State Management:** React Context API (for global state handling)
- **Routing:** Next.js Dynamic Routing (for SEO-friendly navigation)

Backend:

- **CMS:** Sanity CMS (for product, user, and order management)
- **API Communication:** RESTful APIs / GraphQL (for efficient data fetching)
- **Authentication:** JWT-based authentication (for secure user logins)

Database & Storage:

- **Sanity CMS Database** (for structured and scalable content management)
- **Cloud Storage:** Cloudinary / AWS S3 (for handling product images)

Security:

- **Authentication:** NextAuth.js (for secure user login/logout)
- **Data Protection:** HTTPS, JWT, and role-based access control

Payment Integration:

- **Payment Gateway:** Stripe / Razorpay (for secure transactions)
- **Order Management:** Webhooks for real-time order status updates

3. Development Steps & Integration

Step 1: Frontend Development

- Implemented product listing, filtering, and search functionality.
- Designed an intuitive cart and wishlist system.
- Integrated a responsive UI using Tailwind CSS.

Step 2: Backend Setup & API Integration

- Configured Sanity CMS for structured product and order data.
- Built API routes in Next.js to handle product retrieval and cart updates.
- Set up authentication mechanisms for user security.

Step 3: Security Measures

- Implemented JWT-based authentication for user sessions.
- Applied CORS policies and rate limiting to prevent security threats.
- Enforced HTTPS and secure cookies for data protection.

Step 4: Payment Gateway Integration

- Integrated Stripe API for handling transactions securely.
- Set up webhooks to update order status in real time after payment confirmation.

Step 5: Testing & Deployment

- Conducted unit and integration testing to identify and fix bugs.
- Deployed the frontend on Vercel and backend on a cloud server.
- Optimized performance by implementing lazy loading and caching.

4. Challenges & Solutions

Challenge	Solution Implemented
Managing product and user data efficiently	Used Sanity CMS to structure and manage content
Handling secure user authentication	Implemented JWT authentication with NextAuth.js
Ensuring smooth API communication	Used RESTful APIs with caching mechanisms
Payment gateway security	Implemented Stripe with secure tokenized transactions
Performance optimization	Used server-side rendering (SSR) and image optimization

5. Best Practices Followed

- **Code Modularity:** Separated concerns between frontend, backend, and APIs.
- **Security First:** Applied encryption, role-based access, and secure authentication.
- **Scalability:** Chose a CMS (Sanity) that supports business growth.
- **SEO Optimization:** Implemented Next.js SSR and dynamic meta tags.
- **Performance Optimization:** Used lazy loading, image compression, and caching.
- **User Experience Focus:** Ensured smooth navigation, quick checkout, and real-time updates.

6. Conclusion

The furniture e-commerce website was built with a focus on scalability, security, and performance. By leveraging Next.js, Sanity CMS, and Stripe integration, the platform ensures a seamless shopping experience. Challenges such as authentication, API management, and payment security were successfully addressed through best practices and modern technologies. The result is a high-performing, user-friendly, and secure e-commerce platform ready for scaling and future enhancements.

BY: AQSA MEMON