E-Commerce Marketplace Technical Document

1. Project Overview

This is a **single-owner e-commerce platform** developed using **Next.js** with **Sanity CMS** as the backend. The platform consists of interfaces for **buyers and an admin**, providing a seamless shopping experience while focusing on **performance, scalability, and user engagement**.

2. Technology Stack

- **Frontend:** Next.js (TypeScript, Tailwind CSS)
- **Backend:** Sanity CMS (Headless CMS for structured content management)
- **Database:** Sanity CMS (GROQ queries for data retrieval)
- **Authentication:** NextAuth.js (Google, Email/Password, JWT-based authentication)
- **API Management:** Next.js API Routes
- **Payment Gateway:** Stripe (Secure transactions)
- **Deployment:** Vercel (Frontend), Sanity Hosting (CMS Backend)

3. User Roles & Features

Buyers:

- Browse and search products
- Add to cart and wishlist
- Secure checkout with Stripe
- Track orders and view order history
- Review and rate products
- Manage account settings

Admin:

- Manage product listings
- View order details and earnings
- Manage inventory and stock
- Monitor transactions and disputes
- Generate reports and analytics

4. Sanity CMS Schema Design

User Schema

```
{

  "name": "user",
  "type": "document",
  "fields": [
   { "name": "name", "type": "string" },
   { "name": "email", "type": "string", "unique": true },
   { "name": "role", "type": "string", "options": ["buyer", "admin"] },
   { "name": "password", "type": "string", "hidden": true }
  ]
}
```

Product Schema

```
{
 "name": "product",
 "type": "document",
 "fields": [
  { "name": "title", "type": "string" },
  { "name": "description", "type": "text" },
  { "name": "price", "type": "number" },
  { "name": "category", "type": "reference", "to": [{ "type": "category" }] },
  { "name": "images", "type": "array", "of": [{ "type": "image" }] },
  { "name": "stock", "type": "number" }
 ]
}
```

Order Schema

```
{
 "name": "order",
 "type": "document",
  "fields": [

   { "name": "buyer", "type": "reference", "to": [{ "type": "user" }] },

   { "name": "items", "type": "array", "of": [{ "type": "reference", "to": [{ "type": "product" }] }] },

   { "name": "total", "type": "number" },

   { "name": "status", "type": "string", "options": ["pending", "shipped", "delivered", "cancelled"]
}

  ]

}
```

5. API Endpoints Plan

Authentication

- POST /api/auth/signup - Register a new user
- POST /api/auth/login - Authenticate user
- GET /api/auth/logout - Logout user

Product Management

- GET /api/products - Retrieve all products
- GET /api/products/:id - Retrieve product details
- POST /api/products - Create a new product (admin only)
- PUT /api/products/:id - Update product details (admin only)
- DELETE /api/products/:id - Remove a product (admin only)

Order Management

- POST /api/orders - Place an order
- GET /api/orders/:id - Retrieve order details
- GET /api/orders/user/:id - Retrieve orders for a specific user
- PUT /api/orders/:id/status - Update order status (admin only)

User Management

- GET /api/users/:id - Retrieve user profile
- PUT /api/users/:id - Update user profile

6. Deployment Strategy

- **Version Control:** GitHub (CI/CD pipeline integration)
- **Hosting:**
  - Frontend: Vercel (Scalable and serverless hosting)

  - CMS Backend: Sanity Hosting (Optimized for structured content management)
- **Database:** Sanity Studio with GROQ-based queries
- **Monitoring & Analytics:** Google Analytics + Sentry for real-time error tracking

7. Security & Performance Considerations

- Secure JWT-based authentication
- Role-based access control (RBAC) implementation
- Data validation and sanitization for protection against vulnerabilities
- Optimized image handling via Sanity CDN
- Performance-optimized server-side rendering (SSR) and static site generation (SSG)

8. Future Enhancements

- AI-driven product recommendations
- Bidding/auction system integration
- Development of a mobile app using React Native