

AURA

ELEVATE YOUR FASHION

1) Technical Requirements

The technical requirements for the Aura e-commerce platform are divided into the following categories:

1. Frontend Requirements

Key Features of Our E-Commerce Platform:

- **User-Friendly Interface:** Effortlessly browse products and manage orders with a simple and intuitive design.
-
- **Responsive Design:** Enjoy a seamless experience across devices, whether you're using a mobile phone or desktop computer.

Essential Pages:

Home Page: A dynamic showcase of featured products, promotions, and special offers to capture customer interest.

Product Listing: Browse through a well-organized catalog of products with filtering options and sorting tools for easy navigation.

Product Details: In-depth product descriptions, customization options, and customer reviews to aid in informed purchasing decisions.

Shopping Cart: A convenient space to manage items, update quantities, and review your selections before checkout.

Checkout Process: A streamlined, multi-step journey that guides users through payment and order confirmation with ease.

Order Confirmation: A detailed summary including order ID, payment details, and a tracking link for real-time updates on your purchase.

Backend Requirements (Sanity CMS):

Sanity CMS Integration:

Sanity CMS will serve as the central content management system to efficiently store and manage essential data, including:

- **Product Information:** Product name, price, stock levels, images, and available customization options.
- **Customer Data:** Personal details such as name, shipping address, and contact information.
- **Order Records:** Comprehensive order details including order ID, order status, payment information, and shipment tracking.

Schema Design: Tailor custom schemas within Sanity CMS to align with business processes, ensuring flexibility and scalability to accommodate future growth.

Third-Party API Integrations:

Payment Gateways: To Seamlessly integrate secure and trusted payment solutions, such as **Stripe**, to facilitate smooth and safe transactions.

Shipment Tracking API: Utilize services like **Ship Engine** to provide real-time shipment tracking, ensuring customers are always updated on the status of their orders.

Real-Time Delivery API: Offer customers live updates on express deliveries, providing precise delivery times and status through integrations with reliable **Real-Time Delivery APIs**.

Product API: Implement a robust **Product API** to sync and manage product data between the CMS and the front-end. This API will handle product catalog updates, stock management, and pricing, ensuring accurate and real-time information.

2) Design System Architecture

High-Level Overview:

Frontend (Next.js):

Leverages dynamic rendering to create responsive pages and components that adapt based on user interactions.

Seamlessly integrates with the backend to retrieve product data, manage orders, and present real-time shipment status updates.

Backend (Sanity CMS):

Centralized content management system that efficiently handles all eCommerce-related data.

Serves as the primary data source for managing product details, customer information, and order records, ensuring consistent updates across platforms.

Authentication:

We will be using Clerk or next auth for our project

Third-Party Integrations:

Payment Gateway: Securely processes customer transactions, offering popular solutions such as Stripe or PayPal.

Shipment Tracking API: Delivers real-time tracking updates through integrations with services like Ship Engine keeping customers informed about their shipments.

Here's a more concise version of the user workflow:

3) Key Workflows

3.1 User Registration

- **Process:**
 1. User signs up via the frontend using **Clerk** for a seamless registration process.
 2. Registration details, including user information, are securely stored in **Sanity CMS**.

3. Backend handles login and session management, ensuring secure access for the user.

3.2 Product Browsing

- **Process:**
 1. User navigates through product categories displayed on the frontend.
 2. The **Sanity CMS API** retrieves essential product data such as name, price, stock, description, and images.
 3. Dynamic and real-time product listings are displayed on the frontend for an optimal browsing experience.

3.3 Order Placement

- **Process:**
 1. User adds desired products to the cart and proceeds to checkout.
 2. Order details, including product information, quantities, and shipping address, are sent to **Sanity CMS**.
 3. Payment is processed securely via **Stripe**.
 4. After payment confirmation, the order is recorded in **Sanity CMS**, and a confirmation email is sent to the user.
 5. **ShipEngine** updates shipment details, and tracking information is generated.

3.4 Shipment Tracking

- **Process:**
 1. **ShipEngine** API is used to fetch real-time shipping updates and generate tracking details.
 2. Shipment tracking information is displayed on the frontend, allowing users to track their orders.

3.5 Inventory Management

- **Process:**
 1. Product stock levels are managed within **Sanity CMS**, with real-time updates.
 2. When a product is added to the cart, the current stock level is checked via the **Sanity CMS API**.
 3. **Out-of-stock** items are moved to the wishlist rather than the cart, while **in-stock** items can proceed to checkout.

3.6 Order Delivery & Feedback

The user receives their order and is prompted to provide feedback or a product review.

API Requirements

Endpoint	Method	Purpose	Response Example
/products	GET	Fetch all product details	[{ "name": "Product Name", "slug": "product-slug", "price": 100 }]
/order	POST	Submit new order details	{ "orderId": 123, "status": "success" }
/shipment-tracking	GET	Fetch real-time tracking updates	{ "trackingId": "AB123", "status": "In Transit" }
/delivery-status	GET	Fetch express delivery tracking information	{ "orderId": 456, "deliveryTime": "30 mins" }
/inventory	GET	Fetch real-time stock levels	{ "productId": 789, "stock": 50 }
/cart	POST	Add product to cart	{ "cartId": 101, "items": [...] }
/wishlist	POST	Add product to wishlist	{ "wishlistId": 202, "items": [...] }

SANITY SCHEMA

```
import { defineType } from 'sanity';

export default defineType({
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'title',
      type: 'string',
      title: 'Title',
      validation: (Rule) => Rule.required().max(100),
    },
    {
      name: 'rating',
      type: 'number',
      title: 'Rating',
      validation: (Rule) => Rule.min(0).max(5),
      description: 'Rating of the product (out of 5)',
    },
    {
      name: 'slug',
      type: 'slug',
      title: 'Slug',
      description: 'Automatically generated slug based on product ID',
      options: {
        source: (doc) => doc._id,
        slugify: (input) => input.toLowerCase().replace(/\s+/g, '-'),
        maxLength: 128,
      },
    },
  ],
});
```

```
    },
  },
  {
    name: 'price',
    type: 'number',
    title: 'Price',
    validation: (Rule) => Rule.required().min(0),
  },
  {
    name: 'discountedprice',
    type: 'number',
    title: 'Discounted Price',
    validation: (Rule) => Rule.required().min(0),
  },
  {
    name: 'availability',
    type: 'string',
    title: 'Availability',
    options: {
      list: [
        { title: 'In Stock', value: 'in-stock' },
        { title: 'Out of Stock', value: 'out-of-stock' },
      ],
    },
    validation: (Rule) => Rule.required(),
  },
  {
    name: 'description',
    type: 'text',
    title: 'Description',
    validation: (Rule) => Rule.max(500),
  },
  {
    name: 'availableColors',
    type: 'array',
    title: 'Available Colors',
    of: [{ type: 'string' }],
    options: {
      layout: 'tags',
    },
  },
],
preview: {
  select: {
    title: 'title',
    media: 'images.0',
    subtitle: 'price',
  },
  prepare({ title, media, subtitle }) {
    return {
      title,
      media,
      subtitle: `${title}${subtitle}`,
    };
  },
},
});
```

Technical Roadmap

Phase 1: Development

- **Landing Page Development:**
Design and launch a compelling, responsive landing page optimized for conversions, featuring key product highlights, unique value propositions, and seamless navigation for users.
- **Product Management:**
Use Sanity CMS to store, manage, and display real-time product data, ensuring seamless dynamic product pages and content updates.
- **User Authentication:**
Implement Clerk for user sign-up and login, integrated with Sanity CMS for secure management of user data.
- **Shopping Cart & Wishlist:**
Develop a smart cart system with real-time stock checks and the ability to save out-of-stock items to a wishlist, with a dynamic total for checkout.
- **Payment Gateway Integration:**
Integrate Stripe for secure payment processing, with clear handling of success and failure scenarios.
- **Real-Time Shipment Tracking:**
Integrate ShipEngine to provide live shipment tracking, ensuring customers can follow their deliveries from start to finish.
- **Inventory & Stock Control:**
Implement an inventory system within Sanity CMS for real-time stock level updates, preventing out-of-stock products from being added to the cart.
-

Phase 2: Testing & Security

- **Comprehensive Testing:**
Perform end-to-end testing of all user flows—registration, product browsing, cart management, checkout, and tracking—to ensure a smooth user experience.

- **Security & Compliance Checks:**
Conduct security audits, particularly around user authentication, payment processing, and data protection, ensuring full compliance with privacy standards.
-

Phase 3: Marketing & Promotion

- **SEO & Content Strategy:**
Optimize the website for search engines with targeted SEO strategies to increase organic visibility. Publish high-quality content to engage potential customers.
 - **Email Marketing:**
Implement email marketing campaigns to target users, with automated welcome emails, order updates, and special promotions. Build and maintain a customer email list.
 - **Paid Advertising & Social Media:**
Run paid ad campaigns across Google, Facebook, and Instagram to drive traffic and boost conversions. Leverage influencer marketing and organic social content to expand brand presence.
 - **Referral Programs & Discounts:**
Launch referral programs and offer exclusive discounts to encourage new users and repeat business, fostering brand loyalty and customer retention.
-

Phase 4: Deployment & Post-Launch

- **Cloud Deployment:**
Deploy the application on a reliable cloud platform (e.g., Vercel, Netlify) to ensure fast, scalable performance under varying traffic levels.
- **Post-Launch Marketing & User Feedback:**
Post-launch, monitor user behavior, gather feedback, and optimize based on performance and customer insights. Adapt marketing strategies to scale the business.
- **Ongoing Optimization:**
Continuously improve the site by enhancing user experience, optimizing loading times, and updating product information. Scale infrastructure to meet growing demand.

AURA

Prepared By Hafeez Siddiqui



