

Technical Requirements Overview

Frontend Development

For the frontend of the marketplace, I have utilized Next.js to create a dynamic and responsive user interface. The key features include:

User -Friendly Interface: The design prioritizes ease of navigation, allowing users to browse products effortlessly.

Responsive Design: The application is optimized for both mobile and desktop users, ensuring a seamless experience across devices.

Essential Pages:

Home Page: A welcoming landing page showcasing featured products and promotions. Product Listing Page: A comprehensive view of all available products with filtering and sorting options. Product Details Page: Detailed information about individual products, including images, descriptions, and reviews. Cart Page: A user-friendly cart interface for reviewing selected items before checkout. Checkout Page: A streamlined process for completing purchases. Order Confirmation Page: A confirmation screen that summarizes the order details post-purchase. About Page: Information about the marketplace and its mission. Contact Page: A section for users to reach out with inquiries or support requests. Account Page: A user profile area for managing personal information and order history.

2. Backend Management with Sanity CMS

For backend management, I have implemented Sanity CMS to effectively handle product data, customer information, and order records. Key aspects include:

Data Management: Sanity serves as the primary database for the marketplace, allowing for easy updates and management of product listings and customer details.

Schema Design: I have focused on designing schemas within Sanity that align with our business goals, ensuring that all necessary data points are captured and easily accessible.

3. Third-Party API Integrations

To enhance functionality and streamline operations, I have integrated the following third-party APIs:

Shipment Tracking: Integrated ShipEngine for real-time shipment tracking, providing users with updates on their order status.

Payment Gateway: Utilized Stripe as the payment gateway, ensuring secure and efficient processing of transactions.

System Architecture and Data Flow

Key Components

Sanity CMS for Data Management:

Sanity CMS serves as the backend database, managing all product data, customer information, and order records. It allows for dynamic content updates and efficient data retrieval.

ShipEngine for Shipment Process:

ShipEngine is integrated to handle the shipment process, providing real-time tracking information for orders. This ensures that users can monitor their shipments seamlessly.

Stripe for Payment Processing:

Stripe is utilized as the payment gateway, ensuring secure and efficient processing of transactions. It handles payment details and provides confirmation back to the user.

Interaction of Sanity with Frontend:

The frontend, built with Next.js, interacts with Sanity CMS through a dedicated Product Data API. This API fetches product listings and details, which are dynamically displayed on the site. When users place orders, the order details are sent back to Sanity for record-keeping.

High-Level System Architecture Diagram:

System Architecture Diagram



API Requirements

General eCommerce Example:

1. Endpoint: /products

Method: GET

Description: Fetch all available products from Sanity to display on the marketplace.

Response Example:

```
[ { "id": 1, "name": "Product A", "price": 100, "tags": ["electronics", "gadgets"], "image":  
"https://example.com/images/product-a.jpg", "width": 500, "height": 500, "rating": 4.5,  
"stockQuantity": 50 }, { "id": 2, "name": "Product B", "price": 150, "tags": ["home",  
"appliances"], "image": "https://example.com/images/product-b.jpg", "width": 600,  
"height": 400, "rating": 4.0, "stockQuantity": 30 } ]
```

2. Endpoint: /products/{id}

Method: GET

Description: Fetch detailed information for a specific product based on its ID.

Response Example:

```
{ "id": 1, "name": "Product A", "price": 100, "tags": ["electronics", "gadgets"], "image":  
"https://example.com/images/product-a.jpg", "width": 500, "height": 500, "rating": 4.5,  
"stockQuantity": 50, "description": "A high-quality electronic gadget." }
```

3. Endpoint: /customers

Method: POST

Description: Create a new customer record in Sanity.

Payload Example:

```
{ "customerId": 1, "firstName": "John", "lastName": "Doe", "companyName": "Doe  
Enterprises", "country": "USA", "address": "123 Main St", "city": "New York", "province":  
"NY", "zipCode": 10001, "phone": 1234567890, "email": "john.doe@example.com",  
"additionalInformation": "Preferred contact via email." }
```

```
{ "message": "Customer record created successfully.", "customerId": 1 }
```

4. Endpoint: /orders

Method: POST

Description: Create a new order in Sanity.

Payload Example:

```
{ "customerId": 1, "products": [ { "productId": 1, "quantity": 2 }, { "productId": 2,  
"quantity": 1 } ], "paymentStatus": "Paid" }
```

```
{ "orderId": 12345, "status": "Order Created", "message": "Your order has been successfully placed." }
```

5. Endpoint: /shipment

Method: GET

Description: Track order status via third-party API.

Response Example:

```
{ "shipmentId": "SHIP123456", "orderId": 12345, "status": "In Transit",  
  "expectedDeliveryDate": "2023-10-15" }
```

Technical Requirements Overview

Frontend Development

For the frontend of the marketplace, I have utilized Next.js to create a dynamic and responsive user interface. The key features include:

User -Friendly Interface: The design prioritizes ease of navigation, allowing users to browse products effortlessly.

Responsive Design: The application is optimized for both mobile and desktop users, ensuring a seamless experience across devices.

Essential Pages:

Home Page: A welcoming landing page showcasing featured products and promotions. Product Listing Page: A comprehensive view of all available products with filtering and sorting options. Product Details Page: Detailed information about individual products, including images, descriptions, and reviews. Cart Page: A user-friendly cart interface for reviewing selected items before checkout. Checkout Page: A streamlined process for completing purchases. Order Confirmation Page: A confirmation screen that summarizes the order details post-purchase. About Page: Information about the marketplace and its mission. Contact Page: A section for users to reach out with inquiries or support requests. Account Page: A user profile area for managing personal information and order history.

2. Backend Management with Sanity CMS

For backend management, I have implemented Sanity CMS to effectively handle product data, customer information, and order records. Key aspects include:

Data Management: Sanity serves as the primary database for the marketplace, allowing for easy updates and management of product listings and customer details.

Schema Design: I have focused on designing schemas within Sanity that align with our business goals, ensuring that all necessary data points are captured and easily accessible.

3. Third-Party API Integrations

To enhance functionality and streamline operations, I have integrated the following third-party APIs:

Shipment Tracking: Integrated ShipEngine for real-time shipment tracking, providing users with updates on their order status.

Payment Gateway: Utilized Stripe as the payment gateway, ensuring secure and efficient processing of transactions.

Data Schema Design:

