

Marketplace Technical Foundation - [Your Marketplace Name]

1. Technical Requirements

Frontend Requirements

Responsive Design

The website layout will adapt seamlessly to both mobile and desktop screens.

A grid-based layout will ensure products display properly across all screen sizes.

Essential Pages

Home Page: Includes featured products, search bar, and promotional banners.

Product Listing Page: Displays products with filtering options (price, popularity, category) and pagination.

Product Details Page: Shows detailed product information such as description, price, stock availability, and an "Add to Cart" button.

Cart Page: Summarizes selected products, quantity adjustments, and total price.

Checkout Page: Captures billing and shipping information and includes payment method selection.

Order Confirmation Page: Displays a success message with order ID and summary.

Interactive Search Functionality

Authentication Features: Users can sign up, log in, and manage their profiles.

Sanity CMS Requirements

Schema for Products:

Fields: Name, price, stock, description, and category.

```
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 Availability' },  
    { name: 'description', type: 'text', title: 'Description' },  
    { name: 'category', type: 'string', title: 'Category' }  
  ]  
};
```

Schema for Orders:

Fields: Order ID, customer name, product details, status, and date.

```
export default {  
  name: 'order',  
  type: 'document',  
  fields: [  
    { name: 'orderId', type: 'string', title: 'Order ID' },  
    { name: 'customerName', type: 'string', title: 'Customer Name' },  
    { name: 'productDetails', type: 'array', of: [{ type: 'reference', to: { type: 'product' } }], title: 'Product  
Details' },  
    { name: 'status', type: 'string', title: 'Status' },  
    { name: 'date', type: 'datetime', title: 'Order Date' }  
  ]  
};
```

};

Third-Party APIs

- Shipment Tracking API
- Payment gateways

Product Data API

Endpoint: /products

Method: GET

Purpose: Fetch all product details.

Response: [{ "id": 1, "name": "Product A", "price": 100, "stock": 50 }]

2. System Architecture

Architecture Diagram

[Frontend (Next.js)]

|

|-----> [Sanity CMS] <-----> [Product Data API]

| |---> [Order Data API]

|

|-----> [Payment Gateway] (e.g., Stripe)

|

|-----> [Third-Party APIs] (e.g., Shipment Tracking API)

3. Key Workflows

User Registration:

User submits the registration form -> Data is securely stored in Sanity CMS -> Email verification is sent.

Product Browsing:

User applies filters or searches -> Data fetched from Sanity CMS -> Products are dynamically displayed.

Order Placement:

User adds items to the cart -> Payment is processed -> Order is confirmed and saved in Sanity CMS.

Shipment Tracking:

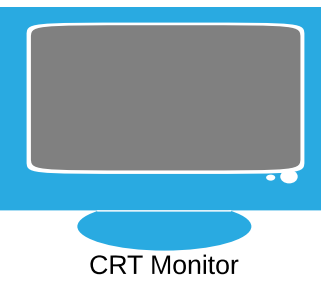
User accesses the tracking page -> Shipment updates are fetched from a third-party API -> Updates are displayed.

Workflow of Eccommerce

ThirdParty

Frontend

Sanity



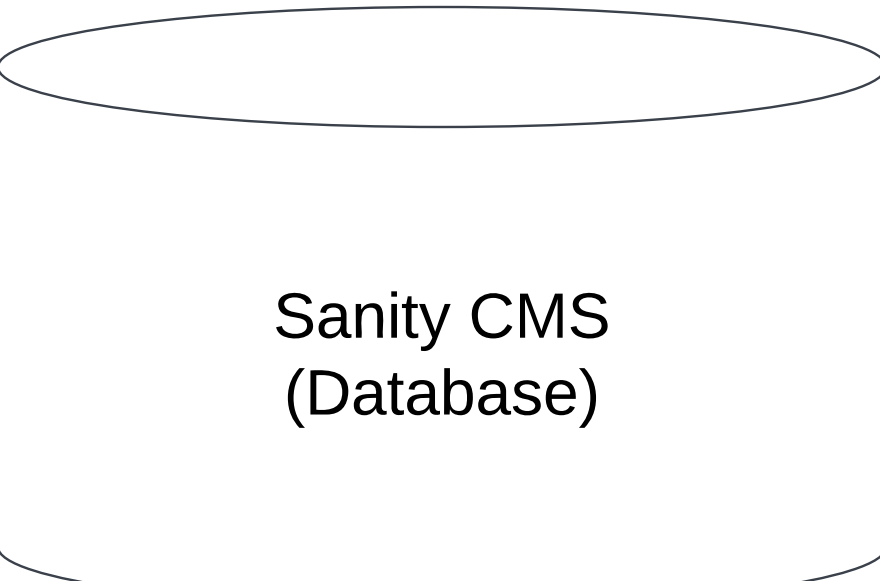
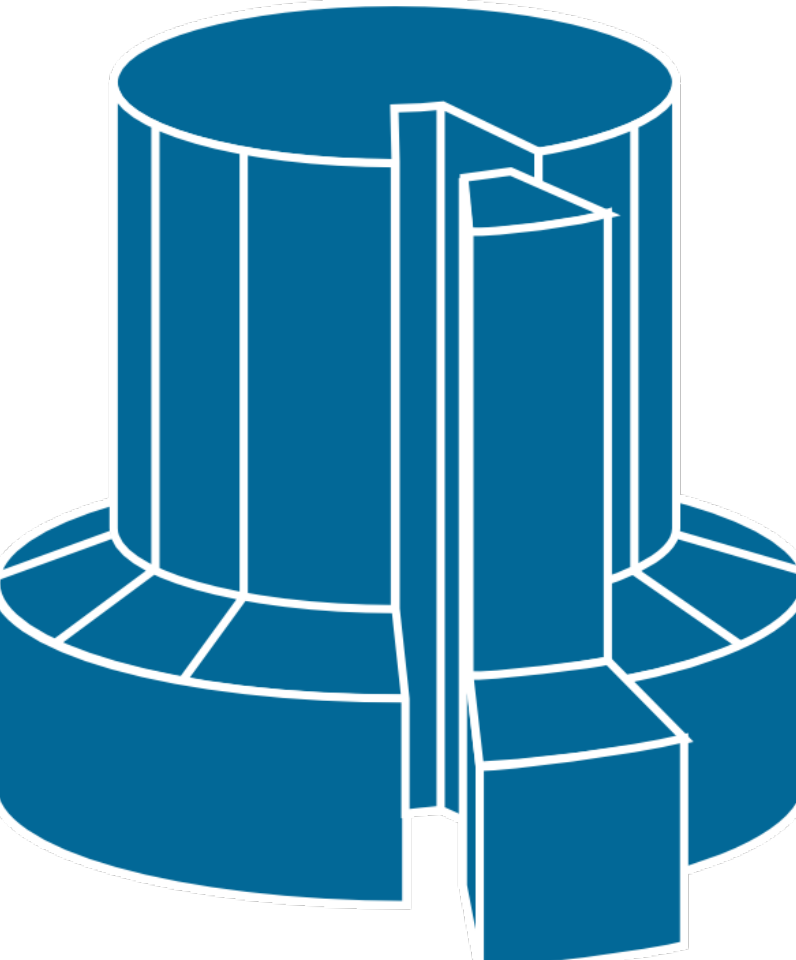
CRT Monitor



CRT Monitor



CRT Monitor



Sanity CMS
(Database)

