

Day 2 Market Place Technical Foundation - (Beaded Jewellery Website)

beadedwear.com

This technical plan outlines the structure, workflow, and integrations required to develop a functional and scalable website for selling beaded jewelry. This plan covers frontend requirements, backend (Sanity CMS), third-party APIs, and workflow integration.

1. Frontend Requirements

The frontend will be developed using **Next.js** with the following key features:

- **Responsive Design:** Tailored for desktops, tablets, and mobile devices.
 - **User-Friendly Navigation:** Categories like Plain bracelets, Design bracelets, Chain bracelets, Stone bracelets, and Custom Designs.
 - **Dynamic Product Pages:** Showcasing images, descriptions, prices, reviews, and stock status.
 - **Shopping Cart:** Allowing users to add/remove items and proceed to checkout.
 - **Customer Dashboard:** Order tracking, purchase history, and saved designs.
 - **Custom Design Request Form:** Users can submit requests for customized jewelry.
-

2. Sanity CMS Integration (Backend)

Role of Sanity

Sanity will act as the content backend, managing product catalogs, user-generated content, and order data.

Sanity Schemas

a. Product Schema

```
export default {  
  name: 'product',  
  title: 'Product',  
  type: 'document',  
  fields: [  
    
```

```

    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
      options: {
        list: ['Plain', 'Design', 'Stone', 'Chain', 'Custom
Designs'],
      },
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'stock',
      title: 'Stock Quantity',
      type: 'number',
    },
    {
      name: 'images',
      title: 'Images',
      type: 'array',
      of: [{ type: 'image' }],
    },
  ],
};

```

b. Order Schema

```
export default {
```

```
name: 'order',
title: 'Order',
type: 'document',
fields: [
  {
    name: 'user',
    title: 'User',
    type: 'reference',
    to: [{ type: 'user' }],
  },
  {
    name: 'products',
    title: 'Products',
    type: 'array',
    of: [{ type: 'reference', to: [{ type: 'product' }] }],
  },
  {
    name: 'totalPrice',
    title: 'Total Price',
    type: 'number',
  },
  {
    name: 'status',
    title: 'Order Status',
    type: 'string',
    options: {
      list: ['Pending', 'Processing', 'Shipped', 'Delivered'],
    },
  },
  {
    name: 'trackingInfo',
    title: 'Tracking Info',
    type: 'string',
  },
  {
    name: 'createdAt',
    title: 'Created At',
    type: 'datetime',
    options: { default: new Date().toISOString() },
  },
],
```

```
};
```

c. User Schema

```
export default {
  name: 'user',
  title: 'User',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'email',
      title: 'Email',
      type: 'string',
    },
    {
      name: 'address',
      title: 'Address',
      type: 'text',
    },
    {
      name: 'phone',
      title: 'Phone Number',
      type: 'string',
    },
    {
      name: 'wishlist',
      title: 'Wishlist',
      type: 'array',
      of: [{ type: 'reference', to: [{ type: 'product' }] }],
    },
  ],
};
```

3. Third-Party APIs

a. Payment Gateway

- **Stripe API:** For secure online payments and refunds.
 - Supports multiple currencies.
 - Handles taxes and discounts.

b. Shipment Tracking

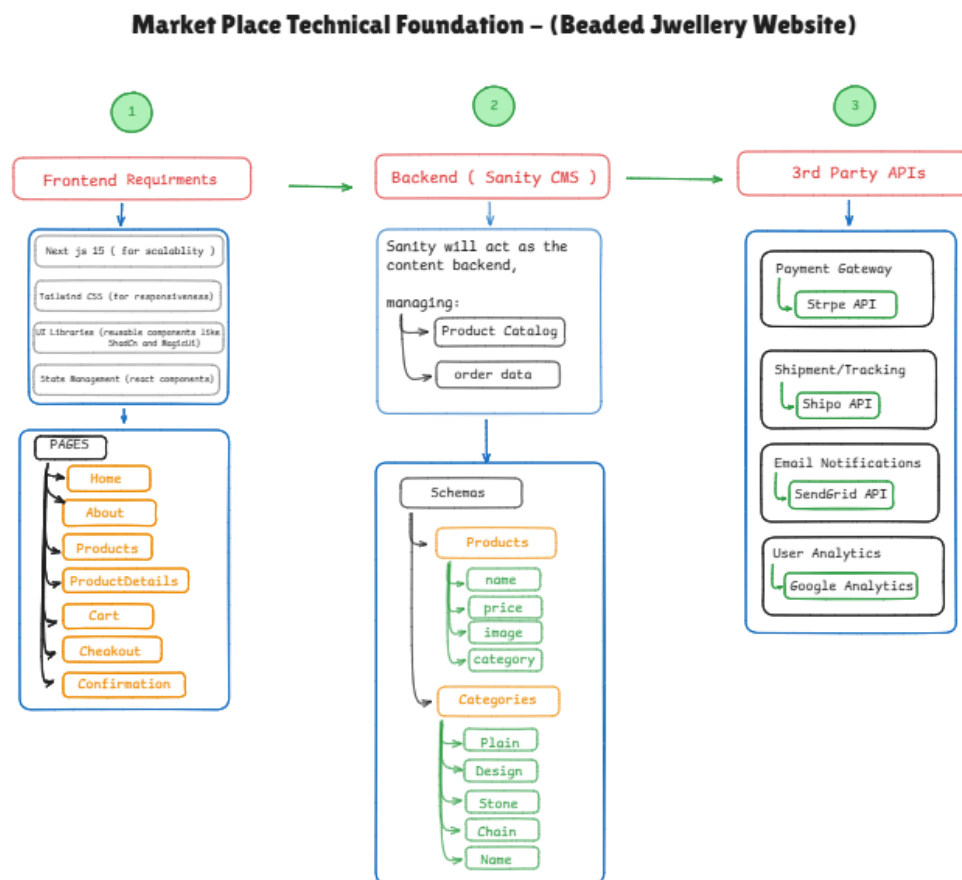
- **Shippo API:** To track shipping statuses in real-time.
 - Fetch tracking numbers.
 - Provide delivery estimates and updates.

c. Email Notifications

- **SendGrid API:** Sends order confirmations, shipping updates, and promotional emails.

d. Analytics

- **Google Analytics API:** For tracking user behavior and optimizing the website.



4. Workflow

Step 1: User Browses the Website

- The user lands on the website and navigates through the product categories.
- The frontend fetches product data dynamically from Sanity CMS.

Step 2: Product Details Page

- The user selects a product and views detailed information, fetched from Sanity, including price, description, images, and stock status.

Step 3: Add to Cart

- The user adds products to the shopping cart, and the cart data is stored in a session or a temporary backend database.

Step 4: Checkout

- The user proceeds to checkout, providing shipping details and making a payment through the Stripe API.
- Once payment is confirmed, an API request sends the order details to Sanity CMS.

Step 5: Order Stored in Sanity

- Sanity CMS records the order, including user details, products, total price, and payment status.

Step 6: Shipment Tracking

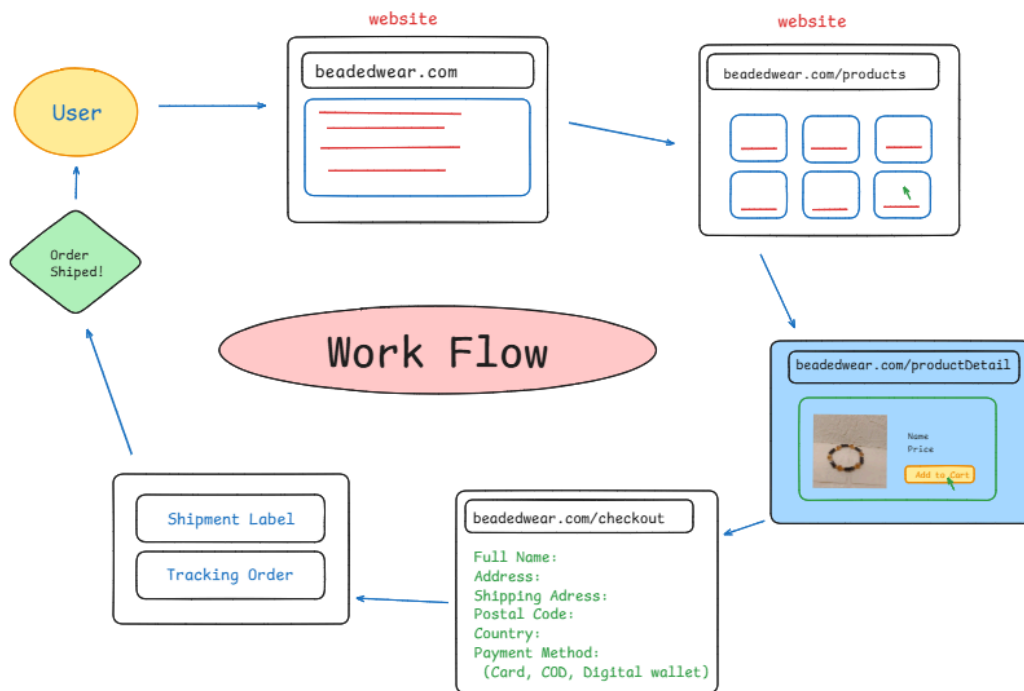
- Shippo API fetches real-time tracking details once the order is shipped.
- Tracking information is updated in Sanity CMS and displayed on the user dashboard.

Step 7: Notifications

- SendGrid API sends an email confirmation of the order and shipping updates.

Step 8: Order Completion

- Once delivered, the order status is updated to "Delivered" in Sanity CMS, and the user is notified via email.



5. Frontend-Backend Integration

- **Frontend Fetching Products:**
 - Query Sanity CMS with GROQ for product data.
- **Order Submission:**
 - POST order details to a backend route that interacts with Sanity CMS.
- **Real-Time Tracking:**
 - Use Shippo API to fetch and update shipment tracking information dynamically.
- **Secure Payments:**
 - Use Stripe SDK for secure payment processing and payment status integration.

This plan ensures the seamless operation of my beaded jewelry website with dynamic content management, secure transactions, and real-time tracking.