**DAY 1: BUILDING THE FOUNDATION OF YOUR ONLINE MARKETPLACE**

**Step 1: Selecting Your Marketplace Model**

**Marketplace Type: General E-Commerce** A versatile e-commerce platform that enables sellers to list products across multiple categories, catering to a broad audience. Key features include:

* Clearly defined product categories.
* Seamless seller registration process.
* Secure payment gateways.
* User-generated reviews and ratings.
* Optimized for both mobile and desktop users.

Prioritize scalability, competitive positioning, and streamlined logistics to ensure long-term success.

**Step 2: Defining Business Objectives**

Establishing clear business goals is crucial for creating a strong foundation. Here’s the initial roadmap:

* **User Growth:** Onboard 1,000 active users within the first six months.
* **Seller Acquisition:** Partner with 50 sellers by the end of the first quarter.
* **Revenue Target:** Achieve $10,000 in monthly revenue within the first year.
* **Brand Reputation:** Build credibility through outstanding customer support.

**Step 3: Structuring Your Data Schema**

A well-defined data schema ensures efficient management of products, orders, customers, and logistics.

**Product Schema**

* **ID:** Unique identifier.
* **Name:** Product title.
* **Price:** Cost per unit.
* **Stock:** Available quantity.
* **Category:** Product classification (e.g., Electronics, Apparel).
* **Tags:** Searchable keywords (e.g., "Bestseller").

**Order Schema**

* **Order ID:** Unique identifier.
* **Customer Details:** Name, contact info, and delivery address.
* **Product List:** Items, quantities, and prices.
* **Status:** Order progress (e.g., Pending, Shipped, Delivered).
* **Timestamp:** Date and time of order placement.

**Customer Schema**

* **Customer ID:** Unique identifier.
* **Name:** Full name.
* **Contact Information:** Phone number and email.
* **Address:** Shipping details.
* **Order History:** List of past purchases.

**Delivery Zones Schema**

* **Zone Name:** Identifier for the delivery area.
* **Coverage:** List of locations served.
* **Assigned Drivers:** Details of courier personnel.

**Shipment Schema**

* **Shipment ID:** Unique tracking number.
* **Order ID:** Linked to the relevant order.
* **Status:** Current delivery state (e.g., In Transit, Delivered).
* **Delivery Date:** Estimated or actual date of arrival.

**Code Implementation: Sanity Data Schemas**

**Category Schema**

import { TagIcon } from '@sanity/icons';

import { defineField, defineType } from 'sanity';

export const categoryType = defineType({

name: 'category',

title: 'Category',

type: 'document',

icon: TagIcon,

fields: [

defineField({ name: 'title', type: 'string' }),

defineField({ name: 'slug', type: 'slug', options: { source: 'title' } }),

defineField({ name: 'description', type: 'text' }),

],

preview: {

select: { title: 'title', subtitle: 'description' },

},

});

**Product Schema**

import { TrolleyIcon } from '@sanity/icons';

import { defineField, defineType } from 'sanity';

export const productType = defineType({

name: 'product',

title: 'Products',

type: 'document',

icon: TrolleyIcon,

fields: [

defineField({ name: 'name', title: 'Product Name', type: 'string', validation: (Rule) => Rule.required() }),

defineField({ name: 'slug', title: 'Slug', type: 'slug', options: { source: 'name', maxLength: 96 }, validation: (Rule) => Rule.required() }),

defineField({ name: 'image', title: 'Image', type: 'image', options: { hotspot: true } }),

defineField({ name: 'description', title: 'Description', type: 'text', validation: (Rule) => Rule.min(20).max(2000) }),

defineField({ name: 'price', title: 'Price', type: 'number', validation: (Rule) => Rule.required().min(0) }),

defineField({ name: 'categories', title: 'Categories', type: 'array', of: [{ type: 'reference', to: { type: 'category' } }] }),

defineField({ name: 'stock', title: 'Stock', type: 'number', validation: (Rule) => Rule.min(0) }),

],

preview: {

select: { title: 'name', media: 'image', price: 'price' },

prepare({ title, media, price }) {

return { title, media, subtitle: `$${price}` };

},

},

});

**Order Schema**

import { BasketIcon } from '@sanity/icons';

import { defineArrayMember, defineField, defineType } from 'sanity';

export const orderType = defineType({

name: 'order',

title: 'Order',

type: 'document',

icon: BasketIcon,

fields: [

defineField({ name: 'orderNumber', title: 'Order Number', type: 'string', validation: (Rule) => Rule.required() }),

defineField({ name: 'stripeCheckoutSessionId', title: 'Stripe Checkout Session ID', type: 'string' }),

defineField({ name: 'stripeCustomerId', title: 'Stripe Customer ID', type: 'string', validation: (Rule) => Rule.required() }),

defineField({ name: 'customerName', title: 'Customer Name', type: 'string', validation: (Rule) => Rule.required() }),

defineField({ name: 'customerEmail', title: 'Customer Email', type: 'string', validation: (Rule) => Rule.required().email() }),

defineField({ name: 'products', title: 'Products', type: 'array', of: [

defineArrayMember({

type: 'object',

fields: [

defineField({ name: 'product', title: 'Product', type: 'reference', to: [{ type: 'product' }] }),

defineField({ name: 'quantity', title: 'Quantity', type: 'number', validation: (Rule) => Rule.required().min(1) }),

],

}),

]}),

defineField({ name: 'totalPrice', title: 'Total Price', type: 'number', validation: (Rule) => Rule.required().min(0) }),

defineField({ name: 'currency', title: 'Currency', type: 'string', validation: (Rule) => Rule.required() }),

defineField({ name: 'status', title: 'Order Status', type: 'string', options: { list: [

{ title: 'Pending', value: 'pending' },

{ title: 'Paid', value: 'paid' },

{ title: 'Shipped', value: 'shipped' },

{ title: 'Delivered', value: 'delivered' },

{ title: 'Canceled', value: 'canceled' },

] }, validation: (Rule) => Rule.required() }),

],

});

This structured approach ensures an efficient marketplace foundation with a scalable and robust data model.