MARKETPLACE BUILDER HACKATHON 2025

Define Technical Requirements (Frontend):

Responsive Design: Templates should adapt seamlessly to desktop and mobile views.

Home Page: Displays categories, banners, and featured products.

Product Details Page: Includes product image, description, price, and "Add to Cart" button.

Sanity CMS (Backend):

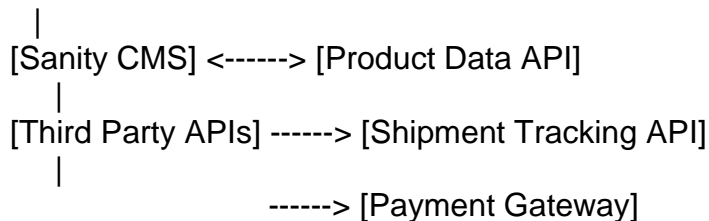**Product management:** Schema includes name, price, image, description, and stock.

Customer management: Fields for name, email, and order history.

Integrate APIs for:

- Payment.
- Shipment tracking.
- Handling Cart

Design System Architecture

```
[Frontend (Next.js)]

     |
   [Sanity CMS] <------> [Product Data API]
      |
   [Third Party APIs] ------> [Shipment Tracking API]
     |
                    ------> [Payment Gateway]
```

Workflow Description:

- o   A user browses products → Fetches data via Sanity CMS API.
- o   User places an order → Order stored in Sanity CMS → Payment processed via Gateway.
- o   Order system is updated using the Shipment Tracking API.
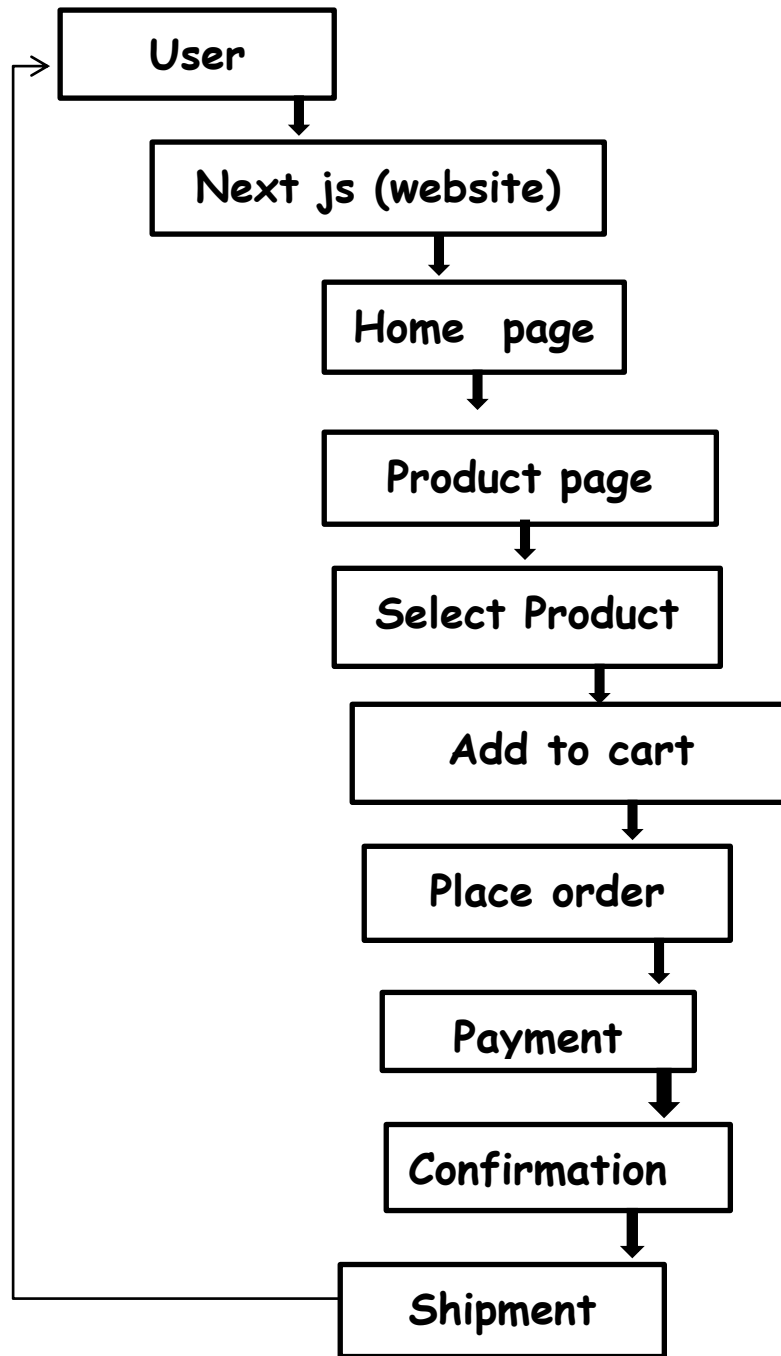
Order Schema

```
export default {
  name: "order",
  type: "document",
  title: "Order",
  fields: [
    {
      name: "orderID",
      type: "string",
      title: "Order ID",
    },
    {
      name: "cartItems",
      type: "array",
      title: "Cart Items",
      of: [
        {
          type: "object",
          fields: [
            { name: "productID", type: "reference", to: [{ type: "product" }], title: "Product
ID" },
            { name: "name", type: "string", title: "Product Name" },
            { name: "price", type: "number", title: "Price" },
            { name: "quantity", type: "number", title: "Quantity" },
          ],
        },
      ],
        },
    {
      name: "status",
      type: "string",
      title: "Order Status",
    },
  ],
};
```

Product Schema

```
export default {
  name: "product",
  type: "document",
  title: "Product",
  fields: [
    {
      name: "name",
      type: "string",
      title: "Product Name",
    },
    {
      name: "price",
      type: "number",
      title: "Price",
    },
    {
      name: "stock",
      type: "number",
      title: "Stock",
    },
    {
      name: "image",
      type: "image",
      title: "Product Image",
      description: "Image of the product",
      },
  ],
};
```

System Architecture diagram

```
         ┌───────────────────┐
    ┌───→│       User         │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │  Next js (website) │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │     Home  page     │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │   Product page     │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │   Select Product   │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │    Add to cart     │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │    Place order     │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │     Payment        │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    │    │   Confirmation     │
    │    └───────────────────┘
    │              ↓
    │    ┌───────────────────┐
    └────│     Shipment       │
         └───────────────────┘
```

API Plan Requirements

End Points	Method	Description	Response
/products	Get	fetch product listings	{id, name, price,stock,image,etc}
/products/:id	Get	fetch details of a specific product by its ID	{id, name, price,stock,image,etc}
/cart	Post	add a product to the cart.	{id, name, price,quantity}
/cart/:userid	Get	get cart details for a specific user	{id, name, price,quantity}
/user/order	Post	place a new order	{orderID, status, 'Success'}
/shipment	Get	track status of a shipment using its ID	{shipmentID, status, deliveryDate}
/payment	Post	process a payment for an order	{paymentID, status, 'Processed'}