Market Blace Builder Hackathon 2025 (Day 2)

Marketplace Technical Foundation

“Avion: A General-Ecommerce Marketplace”

1. Define Technical Requirements:

Frontend Requirements:

1) **User-Friendly Interface:**
   - User-friendly interface for browsing products.
2) **Responsive Design:**
   - Responsive design for mobile and desktop users.
3) **Essential Pages:**
   - **Home Page:** Displays featured products and categories.
   - **Product Listing:** Showcases all products based on category.
   - **Product Details**: Displays details like: price, availability, etc.
   - **Cart:** Summarizes selected products for checkout.
   - **Checkout, & Order Confirmation:** Handles the payment and shows order confirmation.

Sanity CMS as Backend:

1) Use **Sanity CMS** to manage:
   - Product data.
   - Customer Details.
   - Order Records.
2) Design schemas in Sanity CMS for:
   - **Products:** Includes title, description, price, image, and availability status.
   - **Orders:** Tracks customer details, product IDs, quantities, and timestamps.
   - **Customers:** Includes name, contact info, and address.

Third-Party APIs:

- Integrate APIs for:
  - Shipment tracking.
  - Payment gateways, etc.

2. Design System Architecture

System Components:

1) **Frontend (Next.js):**
   - Manages UI interactions and sends requests to the backend.
2) **Sanity CMS:**
   - Acts as the database for products, orders, and customer data.
3) **Third-Party APIs:**
   - Handles shipment tracking, payment gateways.

Data Flow:

1. A user browses products → Frontend requests data via Sanity CMS API → Dynamic display of product listings.
2. User places an order → Order details sent to Sanity CMS → Recorded via API.
3. Shipment info fetched via Third-Party API → Real-time updates to the user.
4. Payment processed via Payment Gateway → Confirmation recorded in Sanity CMS.

System Architecture Diagram

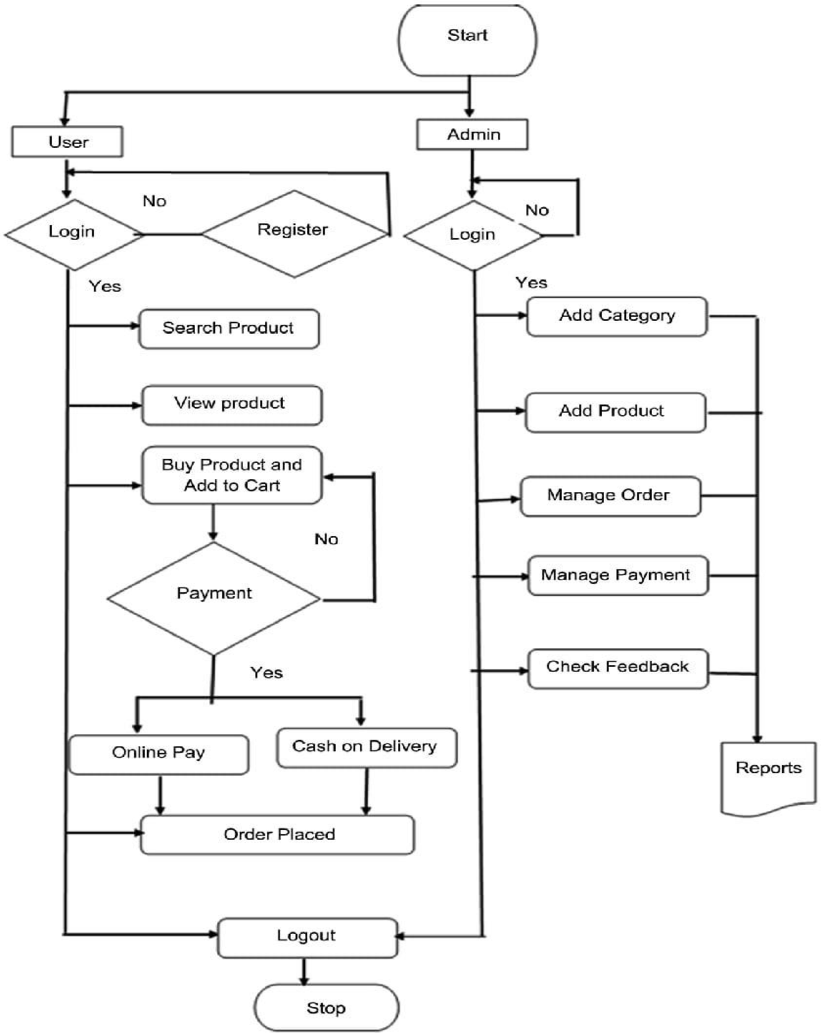
- **Example Architecture**:

  [Frontend (Next.js)] → [Sanity CMS] → [Third-Party APIs for Shipment & Payment].

Key Workflows:

1) **User Registration:**
   - Sign up → Data stored in Sanity → Confirmation sent.
2) **Product Browsing:**
   - Fetch product data via API → Displayed dynamically.
3) **Order Placement:**
   - Cart checkout → Order saved in Sanity.
4) **Shipment Tracking:**
   - Updates fetched via Third-Party API → Displayed to the user.

Workflow Diagram



4. Plan API Requirements

API Endpoints Based on Data Schema

1) **Products**
   - **Endpoint Name:** /products
   - **Method:** GET
   - **Description:** Fetch all available products from Sanity CMS.
   - **Response:** [{ "id": 1, "name": "Product A", "price": 100, "stock": 50, "image": "url_to_image", "material": "Wood", "eco_certification": true }]

2) Orders

- **Endpoint Name:** /orders
- **Method:** POST
- **Description:** Create a new order in Sanity CMS.
- **Payload:** {"customerInfo": {"name": "John Doe", "contact": "123456789", "address": "123 Street Name, City"}, "products": [{ "productId": 1, "quantity": 2 }], "paymentStatus": "Paid"}
- **Response:** {"orderId": 101, "status": "Success", "message": "Order has been placed successfully."}

3) Shipment

- **Endpoint Name:** /shipment
- **Method:** GET
- **Description:** Track order status via a third-party API
- **Response:** {"shipmentId": 201, "orderId": 101, "status": "In Transit", "expectedDeliveryDate": "2025-01-20"}

5. Sanity Schemas for Products and Orders

- **Product Schema**

```
import { defineType, defineField } from "sanity";
export default defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    defineField({
      name: "name",
      title: "Name",
      type: "string",
    }),
    defineField({
      name: "material",
      title: "Material",
      type: "string",
    }),
    defineField({
      name: "price",
      title: "Price",
      type: "number",
    }),
    defineField({
      name: "stock",
      title: "Stock",
      type: "number",
    }),
    defineField({
      name: "ecoCertification",
      title: "Eco Certification",
      type: "boolean",
    }),
    defineField({
      name: "image",
      title: "Image",
      type: "image",
      options: { hotspot: true },
    }),
  ],
});
```

- **Order Schema**

```
import { defineType, defineField } from "sanity";
export default defineType({
  name: "order",
  title: "Order",
  type: "document",
  fields: [
    defineField({
      name: "customerInfo",
      title: "Customer Info",
      type: "object",
      fields: [
        { name: "name", title: "Name", type: "string" },
        { name: "contact", title: "Contact Info", type: "string" },
        { name: "address", title: "Address", type: "string" },
      ],
    }),
    defineField({
      name: "products",
      title: "Products",
      type: "array",
      of: [
        {
          type: "object",
          fields: [
            { name: "productId", title: "Product ID", type: "reference", to: [{ type: "product" }] },
            { name: "quantity", title: "Quantity", type: "number" },
          ],
        },
      ],
    }),
    defineField({
      name: "paymentStatus",
      title: "Payment Status",
      type: "string",
      options: {
        list: ["Pending", "Paid", "Failed"],
      },
    }),
  ],
});
```

Prepared by: Kinza M. Ayub

The End