Marketplace Builder Hackathon 2025 (Day-2)
General E-Commerce Marketplace Plan

System Architecture Overview

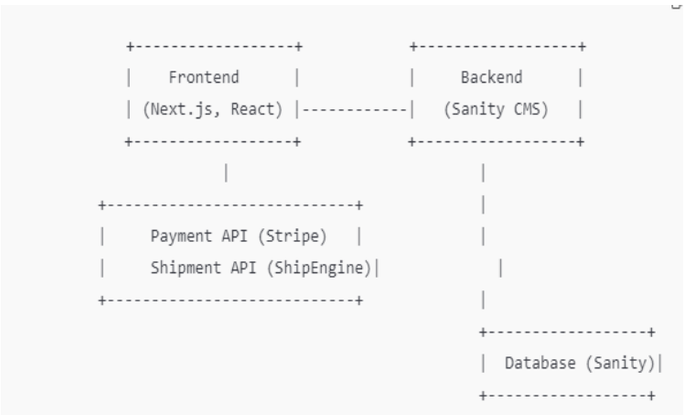
The system architecture leverages Next.js for the frontend, Sanity CMS for content management, and third-party APIs for payment processing and shipment tracking. The key objective is to integrate these components in a way that allows for scalability, user-friendly experiences, and smooth transactions.

Components:

1. **Frontend (Next.js):** The user interface where customers can browse dresses, customize them, and complete purchases.
2. **Sanity CMS:** Content management system used for managing product data, customer details, and orders.
3. **Third-Party APIs:**
    a. **Payment Gateway (e.g., Stripe):** Handles payment processing.
    b. **Shipment API (e.g., UPS):** Tracks the delivery status of orders.
4. **Product Data API:** Fetches product details from Sanity CMS, including custom design options.

Example Architecture Diagram:

- **Frontend (Next.js)**: Handles UI and user interaction.
- **Backend (Sanity CMS)**: Manages products, orders, and customer data.
- **Payment API (Stripe)**: Handles transactions.
- **Shipment API (ShipEngine)**: Handles shipment tracking.
-



2. Key Workflows and User Journey

1. User Registration Flowchart:

- **User signs up → Sanity CMS stores user data → Confirmation email sent → User account is created**

2. Product Browsing Flowchart:

- **User browses categories or searches → Sanity API fetches product data → Products displayed dynamically → User selects a product**

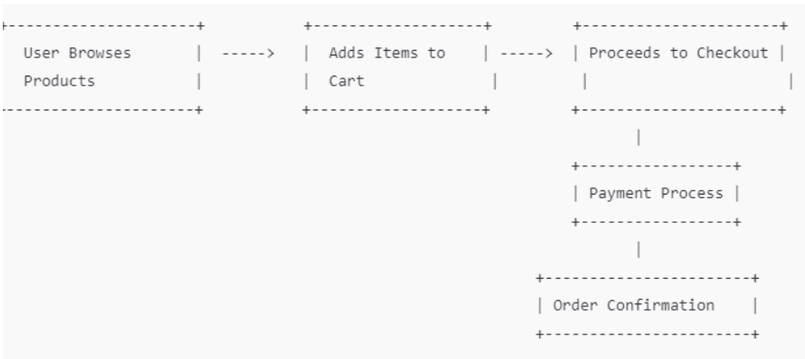
3. Order Placement Flowchart:

- **User customizes a dress → Dress added to cart → User proceeds to checkout → User enters shipping and payment details → Order details stored in Sanity CMS → Payment processed via Stripe → Order confirmed**

4. Shipment Tracking Flowchart:

- **Order placed → Shipment tracking initiated → Third-party API (UPS) fetches status → Shipment status displayed on frontend**

Flowchart for "Order Placement":



3. API Endpoints:

Endpoint	Method	Purpose	Response Example
/products	GET	Fetches all product details	{ "id": 1, "name": "Custom Dress A", "price": 100 }
/orders	POST	Create a new order	{ "orderId": 123, "status": "Success" }
/shipment	GET	Track order status	{ "status": "Shipped", "ETA": "2 days" }

5. Sanity Schema Example

- **Product Schema** (for Sanity CMS):

```
export default {
 name: 'product',   type: 'document',
 fields: [
  { name: 'name', type: 'string', title: 'Product Name' },
  { name: 'price', type: 'number', title: 'Price' },
  { name: 'customizations', type: 'string', title: 'Customization Details' },
  { name: 'stock', type: 'number', title: 'Stock Level' },]};
```

4. API Requirements

1. **/products (GET):**
   Fetch all product details from Sanity CMS.
   - **Response Example:**

```
{
 "id": 1,
 "name": "Custom Design Dress",
 "price": 900,
 "image": "image_url"
}
```

2. **/orders (POST):**
   Submit new order data to Sanity CMS.
   - **Payload Example:**

```
{
 "customerId": "123",
 "products": [{ "productId": 1, "quantity": 2 }],
 "paymentStatus": "paid"
}
```

   - **Response Example:**

```
{
 "orderId": 12345,
 "status": "success"
}
```

3. **/shipment (GET):**
   Fetch real-time shipment status via third-party APIs (e.g., UPS).
   - **Response Example**

```
{  "orderId": 12345,
 "shipmentId": "67890",
 "status": "In Transit",
 "ETA": "2 days"
}
```

---

5. Sanity CMS Schema Example

For the product schema in Sanity CMS:

```
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 Level' },
    { name: 'image', type: 'image', title: 'Product Image' },
    { name: 'customizationOptions', type: 'array', title: 'Customization Options' }
  ]
};
```

5. System Architecture Diagram (Conceptual)

This is a simplified diagram to represent the flow:

```
[Frontend (Next.js)]
       |
  [Sanity CMS] -----> [Product Data API]
       |                  |
[Third-Party API] -----> [Shipment Tracking API]
       |
[Payment Gateway]
```