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

Objective:

To build a scalable, user-friendly e-commerce platform with the following features:

● Product browsing and management via Sanity CMS.
● Authentication using Clerk.
● Order tracking with ShipEngine API.
● Secure payments via Stripe.
● Modern tools like useContext for cart functionality.

System Architecture Diagram:

Graph TD:

● User[User] -->|Signin| Clerk → [Clerk Authentication]
● User -->|Browses| Frontend → [Next.js]
● Frontend -->|Fetches Data| → [Sanity CMS]
● Frontend -->|Manages Cart| → [useContext API]
● Frontend -->|Places Order| → [Stripe Checkout]
● Frontend -->|Generates Label| → [ShipEngine API]
● Sanity -->|Stores| → [Product Data]
● ShipEngine -->|Tracks Order| → [Order Tracking]

Features & Workflow:

Frontend:

User Authentication (Clerk):
● Use Clerk's pre-built authentication components.
● Manage user sessions without storing data in Sanity CMS.

Product Browsing:
● Fetch and display products from Sanity CMS using GROQ queries.

Cart Management:
● Use useContext to manage cart state globally.
● Add/remove items and calculate totals dynamically.

Checkout Process:
● Collect user details and payment via Stripe-hosted checkout.
● Display order confirmation after successful payment.

Order Tracking:
● Generate a shipping label ID using ShipEngine.
● Provide label ID to users for tracking.

Backend:

Sanity CMS:
● Manage products and orders using Sanity Studio.

Custom APIs:
● /api/products: Fetch product data.
● /api/shipping-label: Generate shipping labels using ShipEngine.
● /api/track-order: Retrieve tracking details using ShipEngine.
● /api/checkout: Integrate with Stripe for payments.

Admin Panel:
● Use Sanity Studio for inserting and managing data.

API Requirements:

Endpoint Method Description
/api/products GET Fetch product data from Sanity CMS.
/api/shipping-label POST Generate a shipping label using ShipEngine.
/api/track-order GET Retrieve order status using ShipEngine label ID.
/api/checkout POST Integrate Stripe for payment processing.

Tools & Libraries:

● **Clerk**: Authentication.
● **Sanity CMS**: Content management.

● **ShipEngine API**: Shipping and tracking.
● **Stripe**: Payment gateway.
● **React Context API:** Cart functionality.

Data Models:

Sanity Schemas

Product Schema:

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string' },
    { name: 'subname', type: 'string' },
    { name: 'discount', type: 'number' },
    { name: 'price', type: 'number' },
    { name: 'description', type: 'text' },
    {
      name: 'image',
      type: 'image',
      options: {
        hotspot: true, // Optional: This
        allows image cropping in the
        Sanity Studio
      },
    },
  ],
};
```

Deliverables:

1. **System Architecture Diagram:** Shows component interaction.
2. **Sanity Schemas**: For products and orders.
3. **API Endpoints:** For shipping, tracking, and payments.
4. **Frontend Pages:** Authentication, product browsing, cart management, and order confirmation.
5. **Portfolio-Ready Submission**: Polished project showcasing full-stack e-commerce skills.