

Marketplace Builder Hackathon [3] 2025 (Day-2)

E-Commerce Website Interface Kit + Design System

Marketplace Plan

(Vandana chohan)

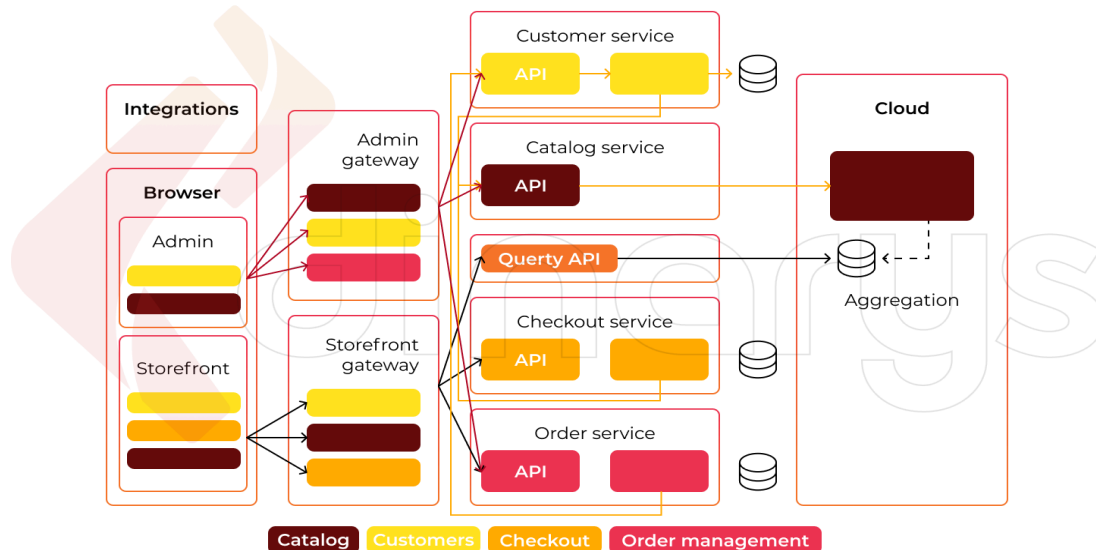
Objective

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

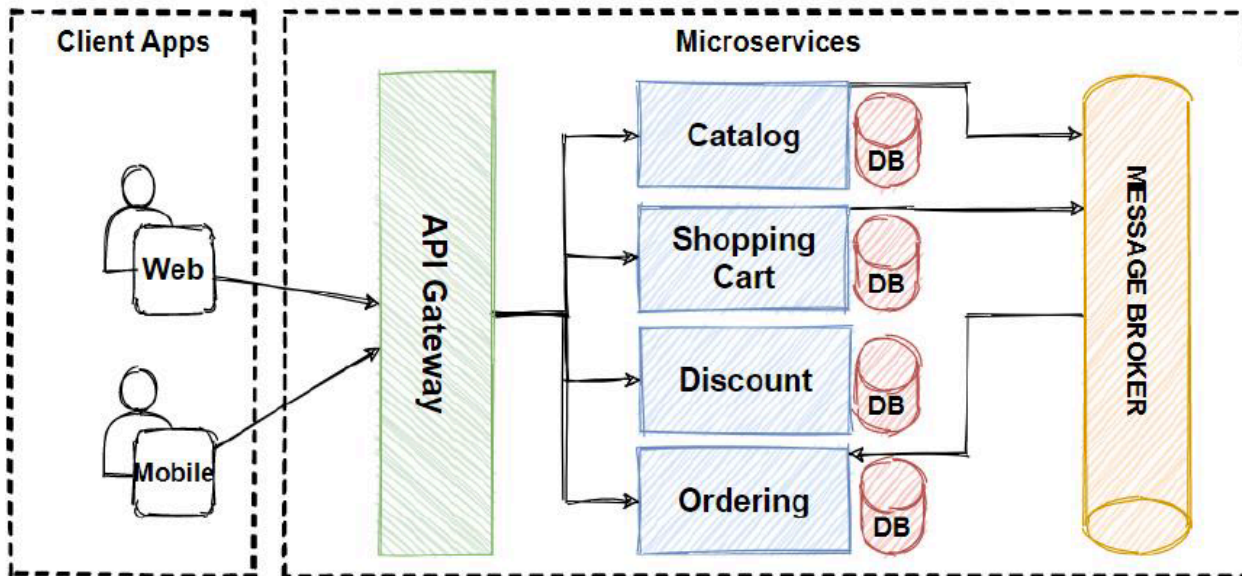
- Product browsing and management via Sanity CMS.
- Authentication using Clerk.
- Order tracking with ShipEngine API.
- Secure payments via Stripe.
- Modern tools like useContext for cart functionality.

System Architecture Diagram

Graph TD:



Features & Workflow



Frontend

Next.js , React

- Use pages, components like.. (Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation....
- 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.
- Third-party APIs for shipment tracking and Plan API Endpoints

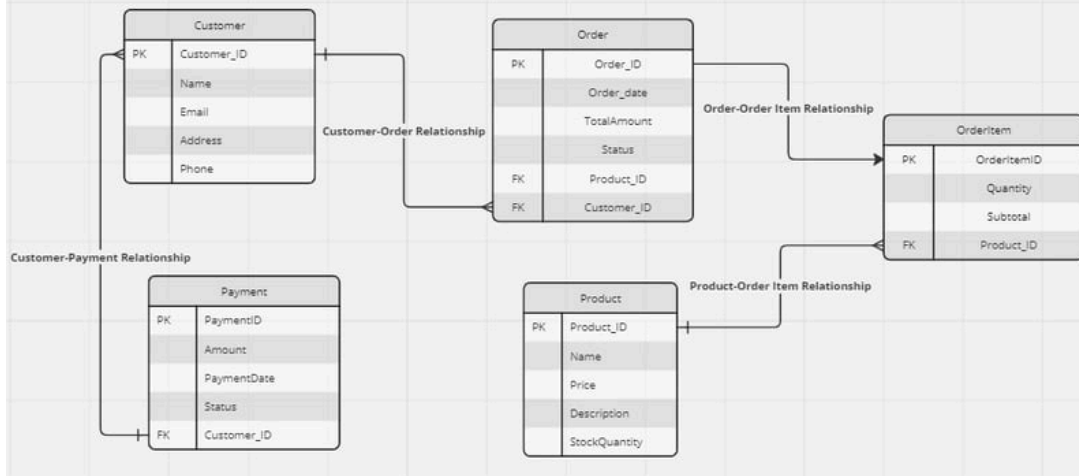
Plan API Endpoints

Endpoint	Method	Purpos	Response Example
/product	GET	Fetch all product details	{ "id": 1, "name": "Product A", "price": 100 }
/orders	POST	Create a new order	{ "orderId": 123, "status": "Success" }
/shipment	GET	Track order status	{ "shipmentId": 456, "status": "In Transit" }

Tools & Libraries

- A. Clerk: Authentication.
- B. Sanity CMS: Content management.
- C. ShipEngine API: Shipping and tracking.
- D. Stripe: Payment gateway.
- E. React Context API: Cart functionality.

ER Diagram Sample for Ecommerce Project



Data Models

Product Schema:

```
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' }
  ]
};
```

Deliverables

- System Architecture Diagram: Shows component interaction.
- Sanity Schemas: For products and orders.
- API Endpoints: For shipping, tracking, and payments.
- Frontend Pages: Authentication, product browsing, cart management, and order confirmation.

- Portfolio-Ready Submission: Polished project showcasing full-stack ecommerce skills (Vandana Chohan)