Day 02 Of Hackathon 3 Planning The Technical Foundation

Introduction:

My E-Commerce website your one-stop destination for trendy women's clothing, stylish men's apparel, and must-have accessories. Discover fashion that fits your lifestyle, crafted with quality and designed to impress. Our Marketplace aims to provide high-quality, affordable, and stylish clothing and accessories for men and women.

1.Define Technical Requirements.

1.Frontend Requirements.

- **Framework:**

o For Framework we use Next.js for dynamic UI and server-side rendering.

- **Styling:**

o For Styling we use Tailwind CSS for dynamic UI and server-side rendering**.**

· **Responsive Design:**

o Mobile, tablet, and desktop compatibility.

· **Pages Included:**

o Homepage, Shop, Products, Product Details, About, Team, Contact, Price, Login/Register, Cart, Wishlist, Checkout, Order Confirmation, Tracking.

2.Sanity CMS as Backend.

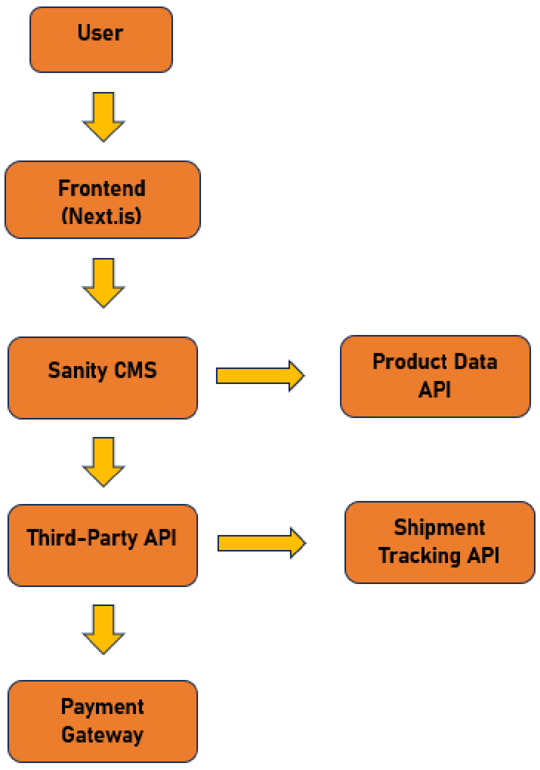
Use Sanity CMS to manage product data, customer details, and order records. Sanity acts as the database.

3. Third-Party APIs.

Integrate APIs for shipment tracking, payment gateways, and other required backend services.

2. Design System Architecture

Creating a diagram to show how system components interact below.



- ### Components Interact:

Frontend Next.js: Frontend Next.js Interact by sharing data via props, context, and state, while fetching dynamic content through API calls.

Sanity CMS: Sanity CMS interacts by providing real-time content via APIs, allowing frontend frameworks to fetch, manage, and display structured data dynamically.

Third party API: Third-party APIs interact by sending HTTP requests (GET, POST, etc.) and receiving responses (JSON/XML) to exchange data between systems.

Payment Gateway: Payment gateways interact by securely processing payment details from the frontend and returning transaction statuses via API responses.

- ## Key Workflows

1.Product Browsing: Users search for products or navigate through categories, view product details, and filter results.

2.Adding to Cart: Users select products, specify quantity or size, and add them to the shopping cart.

3.Order Placement: Users proceed to checkout, provide shipping information, choose a payment method, and confirm the order.

4.Shipment Tracking: Users receive tracking information post-order placement and can track the status of their shipment in real-time.

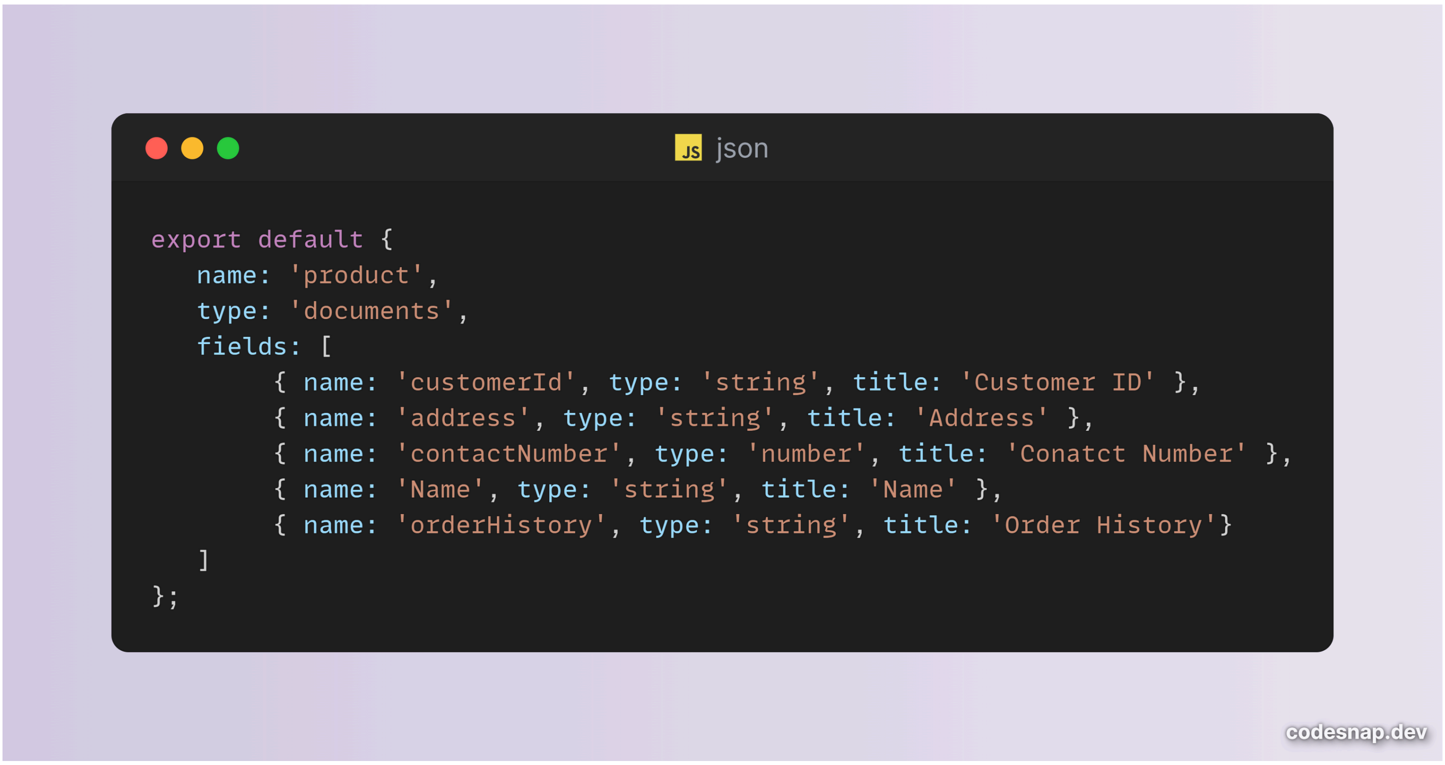
- ## API Endpoint

Endpoints	Method	Description	Response Example
/products	GET	Fetch products details	{“id”: 1, “name”: ”T-shirt”, “category”: “men’s wear”, “price”: 1000, “Description”: “Stylish Men’s Wear” “stock”: 50, “image”: shirt.png”,}
/order	POST	Place a order	{“orderId”: 34569, “productId”: 1, “Quantity”: 2, “payment”: 2000, “totalAmount”: 2500, “userId”: 123}
/customer	POST	Register/Update customer details	{“customerId”: 566, “address”: “xyz”, “contactNumber”: 0315557687, “Name”: “Asiya”, “orderHistory”: “Order Confirmed”}

/deliveryZone	GET	Fetch updates of delivery zone	{“zoneName”: “latifabad”, “coverageArea”: 1567, “assignedDriver”: “leopard”}
/shipment	GET	Track the status of Shipment	{“orderId”: 34569, “shipmentId”: 1566767878, “status”: “Delivered”, “deliveryDate”: 2025-02-28}

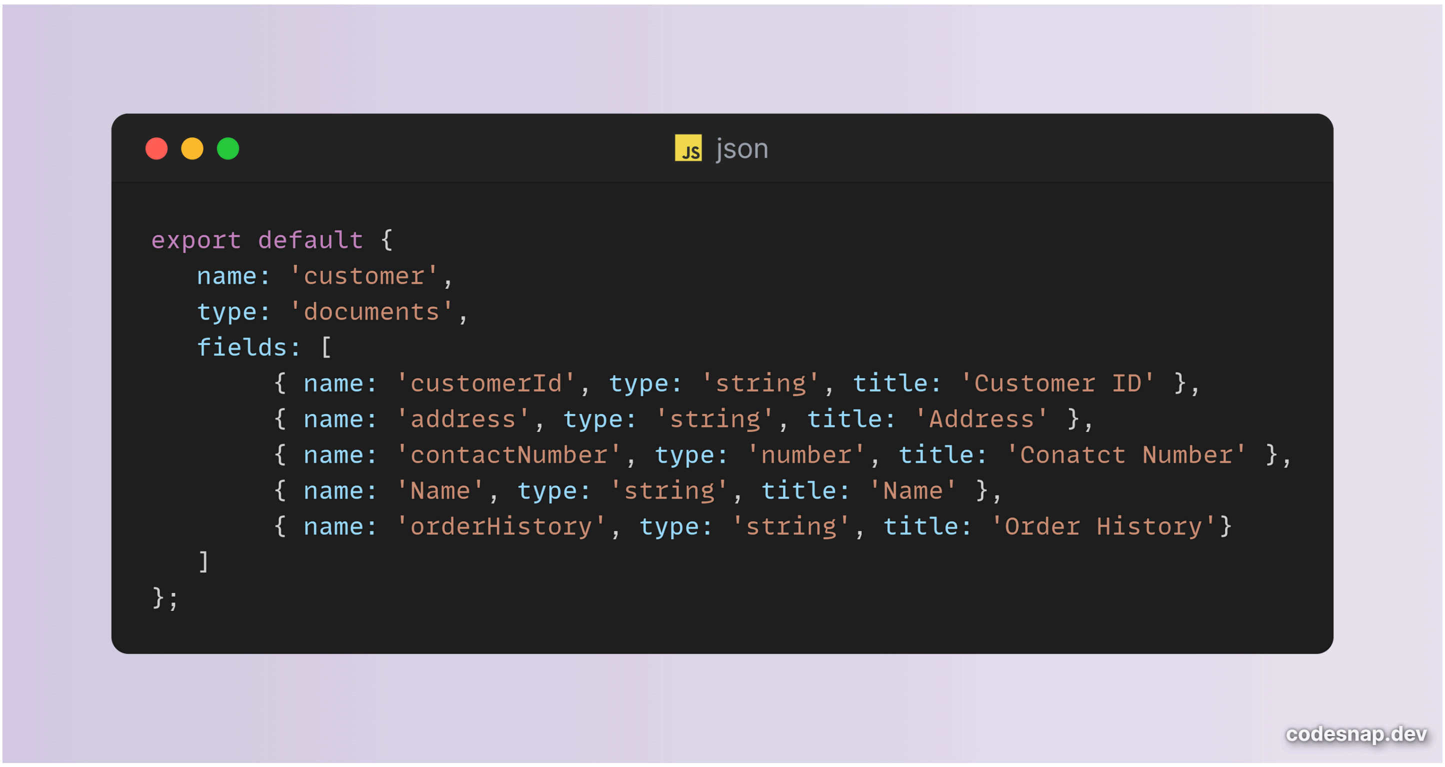
3. Sanity Schema

Product Schema:



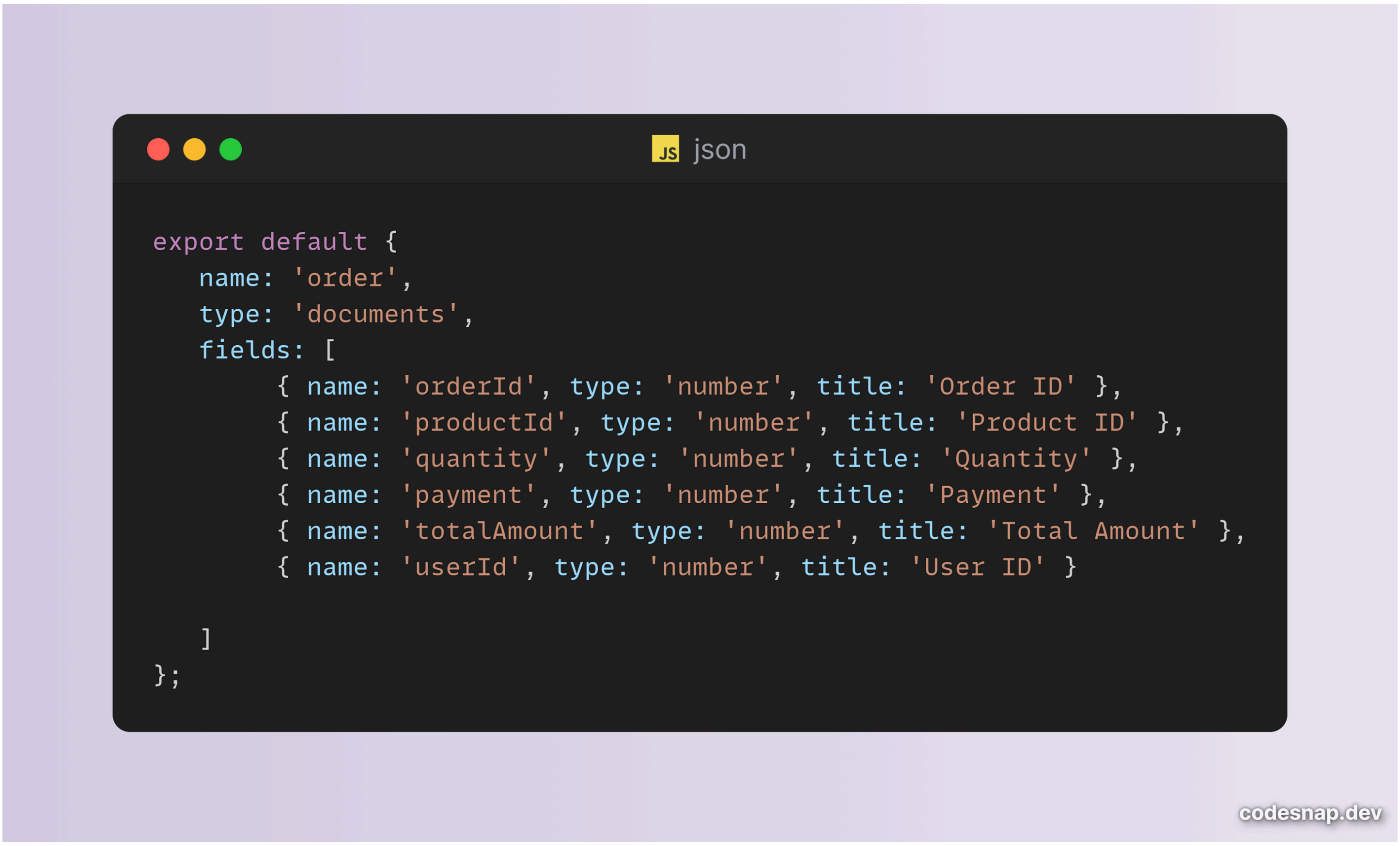
```
export default {
   name: 'product',
   type: 'documents',
   fields: [
        { name: 'customerId', type: 'string', title: 'Customer ID' },
        { name: 'address', type: 'string', title: 'Address' },
        { name: 'contactNumber', type: 'number', title: 'Conatct Number' },
        { name: 'Name', type: 'string', title: 'Name' },
        { name: 'orderHistory', type: 'string', title: 'Order History'}
   ]
};
```

Customer Schema:



```
export default {
   name: 'customer',
   type: 'documents',
   fields: [
        { name: 'customerId', type: 'string', title: 'Customer ID' },
        { name: 'address', type: 'string', title: 'Address' },
        { name: 'contactNumber', type: 'number', title: 'Conatct Number' },
        { name: 'Name', type: 'string', title: 'Name' },
        { name: 'orderHistory', type: 'string', title: 'Order History'}
   ]
};
```

Order Schema:



```
export default {
   name: 'order',
   type: 'documents',
   fields: [
        { name: 'orderId', type: 'number', title: 'Order ID' },
        { name: 'productId', type: 'number', title: 'Product ID' },
        { name: 'quantity', type: 'number', title: 'Quantity' },
        { name: 'payment', type: 'number', title: 'Payment' },
        { name: 'totalAmount', type: 'number', title: 'Total Amount' },
        { name: 'userId', type: 'number', title: 'User ID' }

   ]
};
```

MADE BY: ASIYA KHAN