2. Key Workflows

1. Browsing Products

- Users access the product list, which displays Top Selling and New Arrivals.
- Clicking a product card navigates to the product detail page.

2. Cart Management

- Users add products to the cart via the product detail page.
- Cart items are displayed for review before proceeding to checkout.

3. Checkout Process

- User cart details are processed via the backend.
- Payment API handles transactions, while Sanity CMS records order **details.**

4. Shipment Tracking

- Shipment API provides real-time updates on delivery status.

3. API Endpoints

/products	GET	Fetch all available products	{"id": 1, "name": "Product A", "price": 100 }
/orders	POST	Create a new order	{ "orderId": 123, "status": "Success" }
/shipment	GET	Track order status via third-party	{"shipmentId": 456, "status": "In Transit"}

5. API-Driven Workflow

Product Workflow

1. **GET /products**Fetches a list of all products from the database.
2. Example Response:
3. [
4. { "id": 1, "name": "Product A", "price": 100, "category": "clothes" },
5. { "id": 2, "name": "Product B", "price": 200, "category": "jeans" }
6.]

Order Workflow

1. **POST /orders**Creates a new order with details such as customer information and products purchased.
2. Example Payload:
3. {
4. "customerName": "Raza",
5. "email": "Raza@example.com",
6. "products": [
7. { "id": 1, "quantity": 2 },
8. { "id": 2, "quantity": 1 }
9.]
10. }
11.
12. Example Response:
13. {
14. "orderId": 123,
15. "status": "Success",
16. "orderDate": "2025-01-01T12:00:00Z"
17. }
18.

Shipment Workflow

1. **GET /shipment**Retrieves shipment details for a specific order.
2. Example Query:
3. { "orderId": 123 }
4.
5. Example Response:
6. {
7. "shipmentId": 456,
8. "status": "In Transit",
9. "expectedDelivery": "2025-01-05"
10. }
11.

MarkterPlace Builder Hackathon 2025

Technical foundation

- ***Architecture Diagram***

This architecture is designed to showcase the relationship between different components of the eCommerce platform, including the frontend, backend, APIs, and Sanity CMS. Key workflows and schemas are also highlighted.

- **Frontend**: Built with reusable components such as:
  - Navbar
  - Sale Banner
  - About
  - Product List and Cards
  - Footer
  - Shop
  - Sign in
  - sign up
  - contact
  - order details

- **Backend**: Connects the frontend to APIs and database operations.
- **APIs**:
  - **Shipment API**: Tracks and updates shipment details.
  - **Payment API**: Processes transactions securely.
  - **Message API**: Handles notifications or transactional messages.

1. **Sanity CMS Workflows**:
   - Product details, categories, and order metadata are structured via Sanity schemas.
   - These schemas include fields for name, price, stock, description, and more.

2. Key Workflows

1. Browsing Products

- Users access the product list, which displays Top Selling and New Arrivals.
- Clicking a product card navigates to the product detail page.

2. Cart Management

- Users add products to the cart via the product detail page.
- Cart items are displayed for review before proceeding to checkout.

3. Checkout Process

- User cart details are processed via the backend.
- Payment API handles transactions, while Sanity CMS records order **details.**

4. Shipment Tracking

- Shipment API provides real-time updates on delivery status.

3. API Endpoints

/products	GET	Fetch all available products	{"id": 1, "name": "Product A", "price": 100 }
/orders	POST	Create a new order	{ "orderId": 123, "status": "Success" }
/shipment	GET	Track order status via third-party	{"shipmentId": 456, "status": "In Transit"}

5. API-Driven Workflow

Product Workflow

1. **GET /products**Fetches a list of all products from the database.
2. Example Response:
3. [
4. { "id": 1, "name": "Product A", "price": 100, "category": "clothes" },
5. { "id": 2, "name": "Product B", "price": 200, "category": "jeans" }
6.]