HACKATHON :3
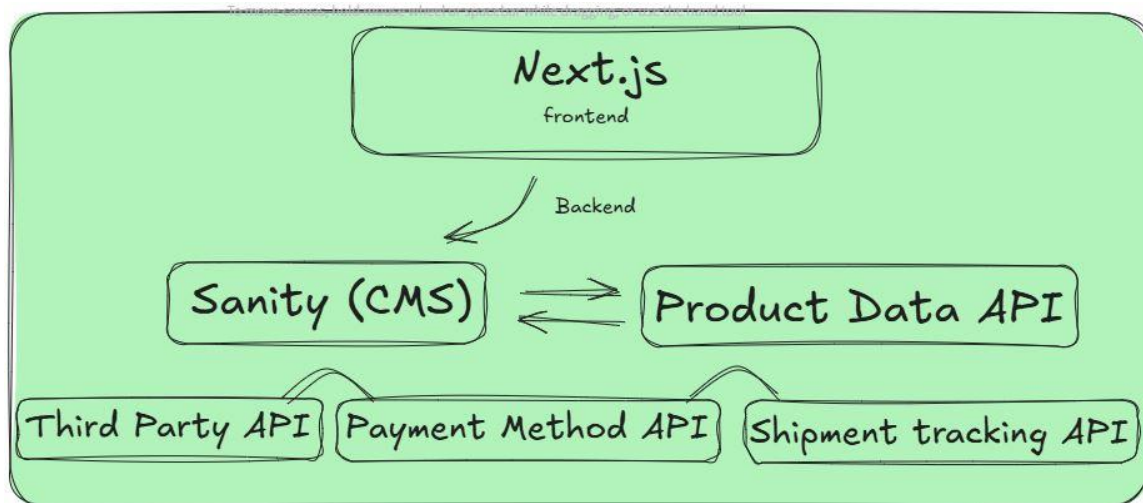
Day 2: Planning the Technical Foundation

Marketplace Technical Overview Foundation

System Architecture
Architecture Diagram:

Key Workflows

User Registration
Users will enters their details → Data sent to Sanity CMS → User profile saved → Confirmation email sent using a third-party service.

Product Browsing
User visits the website → Frontend fetches product data via the Product Data API → Products displayed dynamically.

Order Placement

User adds items to the cart → Proceeds to checkout → Order details sent to Sanity CMS → Payment processed via Payment Gateway API → Order confirmation sent.

Shipment Tracking

Shipment updates fetched from Third-Party API → Status displayed to users on their order page.

API Requirements

Endpoint	Method	Purpose	Response Example
/products	GET	Fetch all product details	{"id": 1, "name": "Product A", "price": 100}
/orders	POST	Create a new order	{"orderId": 123, "status": "success"}
/shipment	GET	Fetch shipment status	{"shipmentId": 456, "orderId": 123, "status": "In Transit"}
/payment	POST	Process payment details	{"transactionId": 789, "status": "success", "amount": 500}

Sanity Schema Design

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' },
    { name: 'image', type: 'image', title: 'Product Image' },
  ],
};
```

Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerName', type: 'string', title: 'Customer Name' },
    { name: 'productIds', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }] },
    { name: 'totalAmount', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', options: { list: ['Pending', 'Shipped', 'Delivered'] } },
  ],
};
```

Prepared by KARIM BUKSH

GIAIC ROLL NO:00453352 (Sunday 2 to 5 pm)