Day 2 Task

Marketplace Technical Foundation

E-Commerce

- ## System Architecture :

```
                    [Frontend ( Next.js)]
                            ↕

[Content Management System          ──►   Product Data API
(Sanity CMS) ]
                            ↕

    [Third Party API]               ──►   Shipment Tracking API
                            ↕

    [Payment Gateway]
```
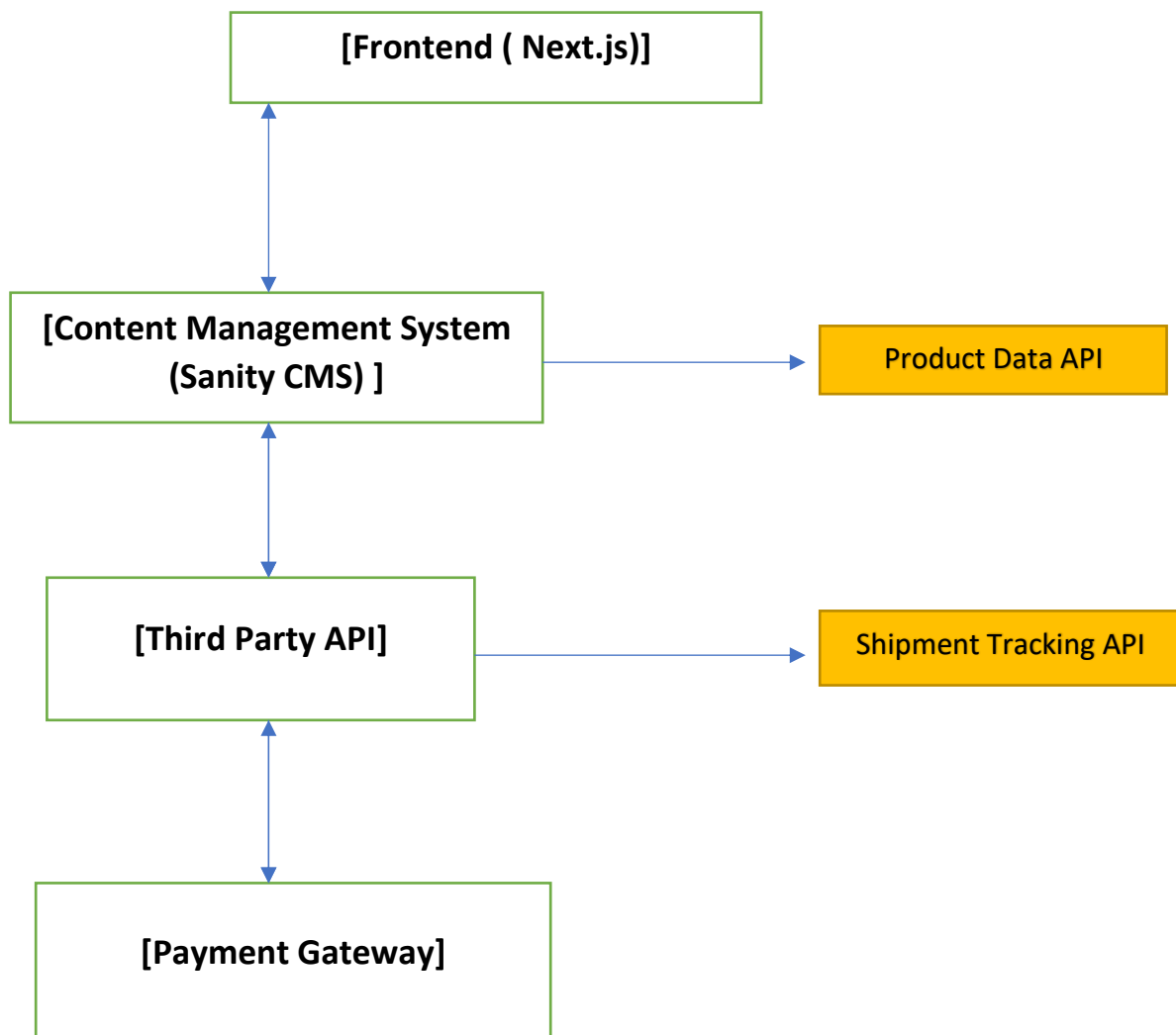
- ## **Work Flow:**
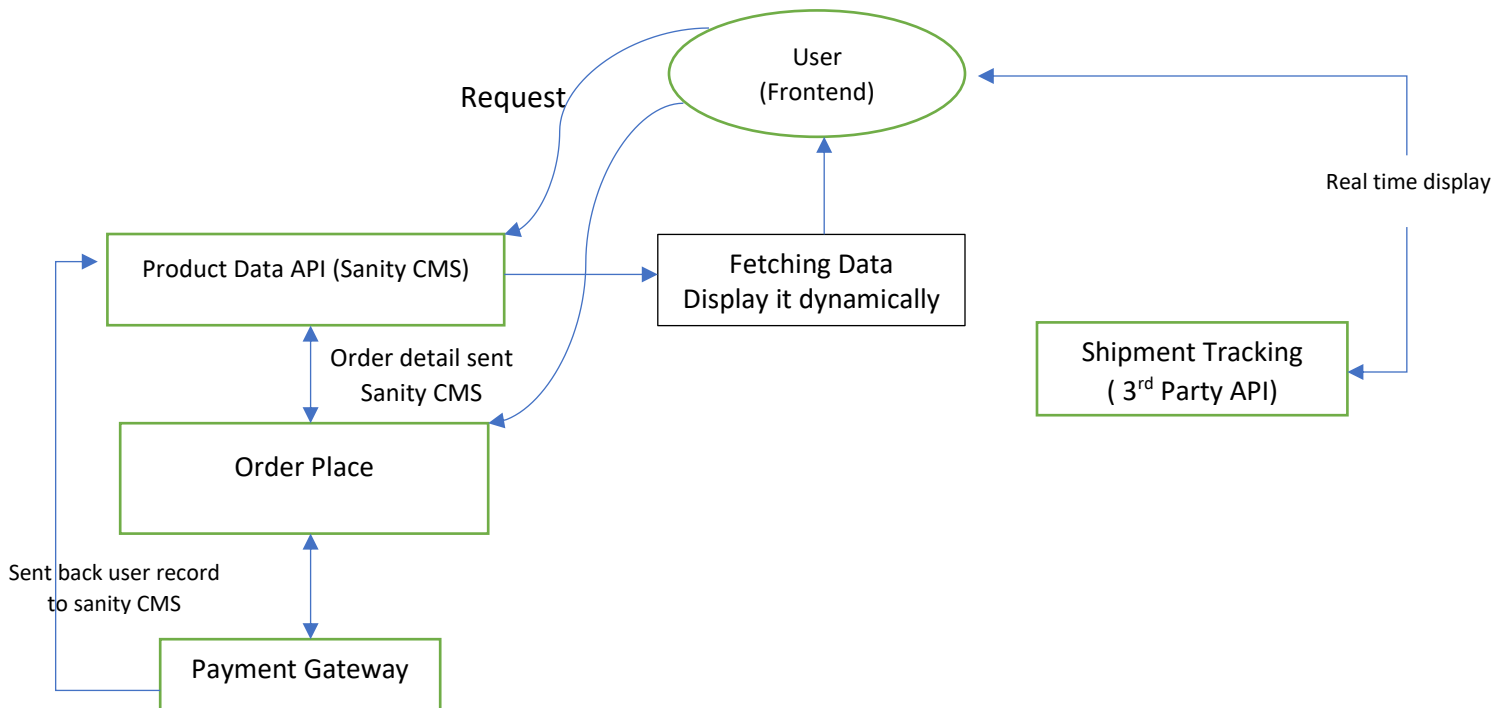
- # **Data Schema Design:**

To design a data schema for a database or CMS, you need to define the entities and their relationships. Based on the entities we've discussed (Products, Orders, Customers, Reviews, and Return Processes), here's a high-level overview of the entities and their relationships:

Entities:

//Product

- id (Primary Key)
- name
- description
- quantity
- price
- discount
- stock
- color
- size
- images

// Order

- id (Primary Key)
- product Id (Foreign Key referencing Product)
- quantity
- amount
- discounted Amount
- Customer
- Consumer Id (Primary Key)
- order Id (Foreign Key referencing Order)
- name
- contact Info
- delivery Zone

//Review

- id (Primary Key)
- customer-Comments
- purchasing-Rate
- delivery-Zone
- product-Id (Foreign Key referencing Product)
- Return-Process
- id (Primary Key)
- order-Id (Foreign Key referencing Order)

- product-Id (Foreign Key referencing Product)
- customer-Id (Foreign Key referencing Customer)
- return-Remarks
- return-Amount
- return-Quantity

• Relationships:

Product to Order: One-to-Many (A product can be part of many orders)

Order to Customer: Many-to-One (Many orders can be placed by one customer)

Product to Review: One-to-Many (A product can have many reviews)

Order to ReturnProcess: One-to-One (Each order can have one return process)

Product to ReturnProcess: One-to-Many (A product can be part of many return processes)

Customer to Return Process: One-to-Many (A customer can initiate many return processes)

• Example ER Diagram:

Here's a simplified representation of the relationships:

Product --< Order >-- Customer

| |

| |

Review   Return Process

This diagram shows the primary relationships between the entities. You can further refine and expand this schema based on additional requirements or constraints.

- **Technical Roadmap:**

TASK	WORK	DURATION
Next.js UI/UX	Routes, Pages, Components	3 days
Sanity	Sanity Schema , Input Data in Sanity	3-4 days
Functionality	Add Logic & Functionality in components	5-6days
Responsive	Make UI/UX Responsiveness to display & do working properly in all screen	2days
Testing	Testing	3-4 days