

Data Flow Example

4. User Browsing:

User visits the marketplace frontend.

Next.js sends a request to the Sanity CMS API to fetch the latest product listings.

Product data is dynamically rendered on the website.

5. Product Details:

User clicks on a product.

Next.js requests detailed product information (image, pricing, stock) from Sanity CMS.

Data is displayed in the frontend.

6. Checkout:

User adds products to the cart and proceeds to checkout.

User provides shipping and payment detail.

Next.js interrogates with the payment Gateway API to process the transaction securely.

7. Order Management:

After payment confirmation.

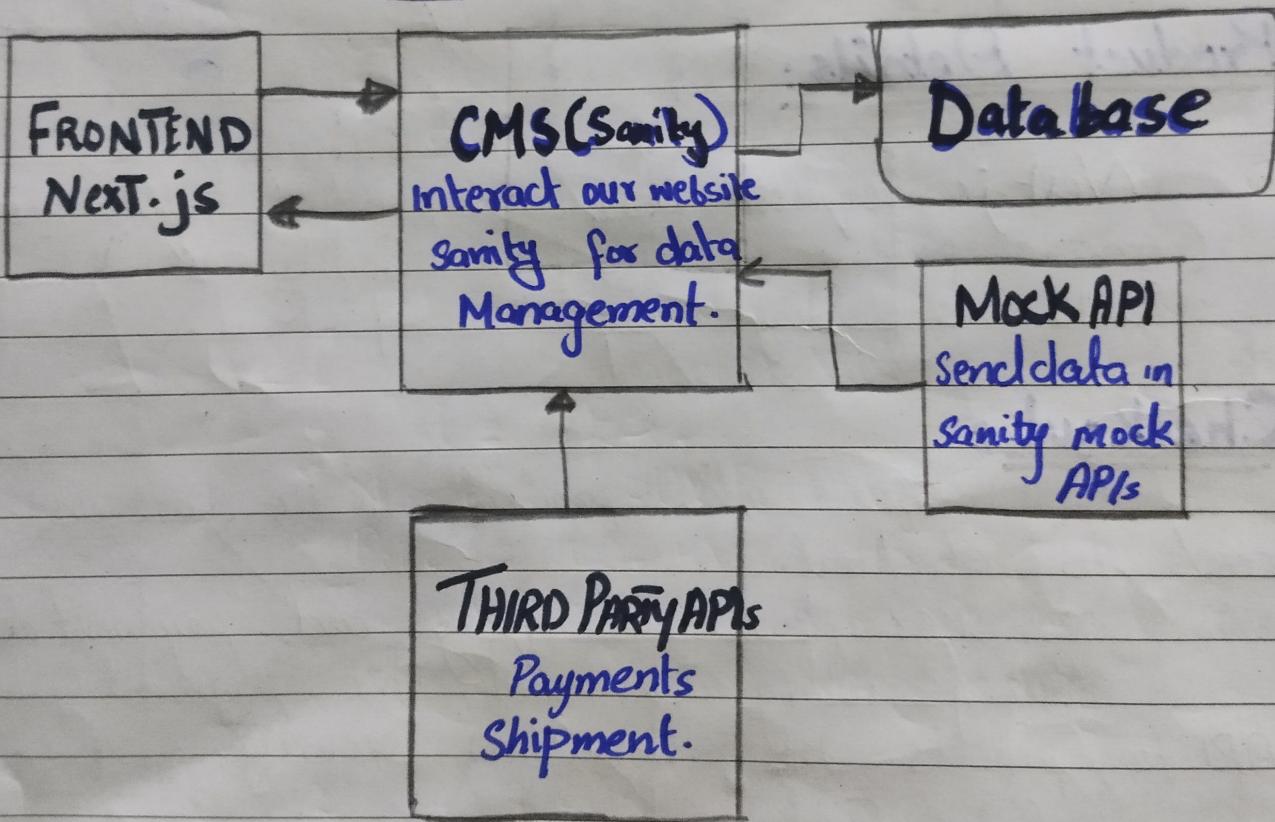
Next.js sends order detail to Sanity CMS for storage.

An order confirmation is displayed to the user.

8. Shipment Tracking:

Shipment tracking API provides delivery updates to the frontend and notification to the customer.

SYSTEM ARCHITECTURE DIAGRAM



How the User / Customer interacts with our Website

Rent a Car Website (Workflow)

user/
customer



Rent a car
e-commerce
website

Go to Product
Detail Page
where user
Select products
and add to
cart

Browses
Rent a car
by category

④

Date 20
MTWTFSS

optional

Card Page
user view
the card
item and product
information.

Login/Register
if not
authenticated

Fills in delivery
detail and
complete
Payment

Receives
Confirmation
via email

(5)

Date 20
MTWTFSS

Here is the system architecture diagram for your market place application, showcasing the interaction between the frontend (Next.js), Backend (Sanity CMS), and Third-Party APIs.

3 PLAN API REQUIREMENT

1- Product Management: /Products

Method: GET

Description: Fetch all available products from Sanity CMS

Response Examples

```
[  
  { "Id": "123",  
    "Name": "SUV Rental",  
    "Price": 100,  
    "Stock": 5,  
    "Image": "SUV-image.jpg" },  
  { "Id": "124",  
    "Name": "Compact Car Rental",  
    "Price": 50  
    "Stock": 10,  
    "Image": "compact - image.jpg" }]
```

2. Product Details

Endpoint Name: /Products/{id}

Method: GET

Description: Fetch detailed information for a specific product by ID.

Response Example:

```
{ "Id": "123",
  "Name": "SUV Rental",
  "description": "Spacious SUV for family trips",
  "Price": 100,
  "Stock": 5,
  "Image": "Suv-image.jpg",
  "features": ["GPS", "Automatic Transmission",
    "Air conditioning"] }
```

3. Add Rental Duration

Endpoint Name: /rental-duration

Method: POST

Description: Add rental details for a specific product

Payload Example:

```
{ "Product Id": "123",
  "duration": "7 days",
  "deposit": 500 }
```

Response Example:

```
{ "Confirmation Id": "789",
  "Status": "Success" }
```

4. Order Management

Endpoint Name: /orders

- Method: POST
- Description: Create a new order in Sanity CMS
- Payload Example:


```
{
        "Customer Info": {
          "Name": "John Doe",
          "email": "john.doe@example.com",
          "Phone": "1234567890"
        }
      }
```

"Product Details":

```
[
  {
    "Id": "123",
    "Name": "SUV Rental",
    "Quantity": 1,
    "Price": 100
  }
]
```

"paymentStatus": "Paid"

Response Example:

```
{
  "orderId": "456",
  "status": "Order Created"
}
```

5. Shipment Tracking

Endpoint Name: /Shipment

Method: GET

Description: Track order Shipment status via a Third-party API.

Response Example:

```
{
  "ShipmentId": "SHIP123",
  "orderId": "456"
}
```

"Status": "In Transit",
"expected DeliveryDate": "2025-01-20" }

6. Customer Management

Endpoint Name: /customers/{id}

Method: GET

Description: Fetch customer detail by ID.

Response Example:

```
{ "id": "001",  
  "Name": "John Doe",  
  "email": "john.doe@example.com",  
  "phone": "1234567890",  
  "orderHistory": [ { "orderID": "456",  
    "date": "2025-01-15",  
    "total": 150 } ] }
```

①

Day 2 ACTIVITIES:

Date _____ 20
MTWTFSS

Transitioning To Technical Planning:-

Here is a high-level system architecture diagram for a marketplace, including workflows and interaction between components:

System Architecture

1. Frontend (NEXT.js):

Acts as the user interface where customers interact with the System.

Includes page for browsing, searching products details, and checkout.

2. Backend (Sanity CMS):

Manage and stores products data, customer details and order records.

Provides APIs to the frontend for dynamic data retrieval.

3. Third-Party APIs:

Payment Gateway: Handle secure payment transactions during check out.

Shipment Tracking API: Provides order delivery updates.

Notification Service: Send email or SMS updates to customers for order confirmations and delivery notification.