(1)

Day 2 Activities:

Date________20___ M T W T F S □

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-

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 click on a product.

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

7- Order Management:

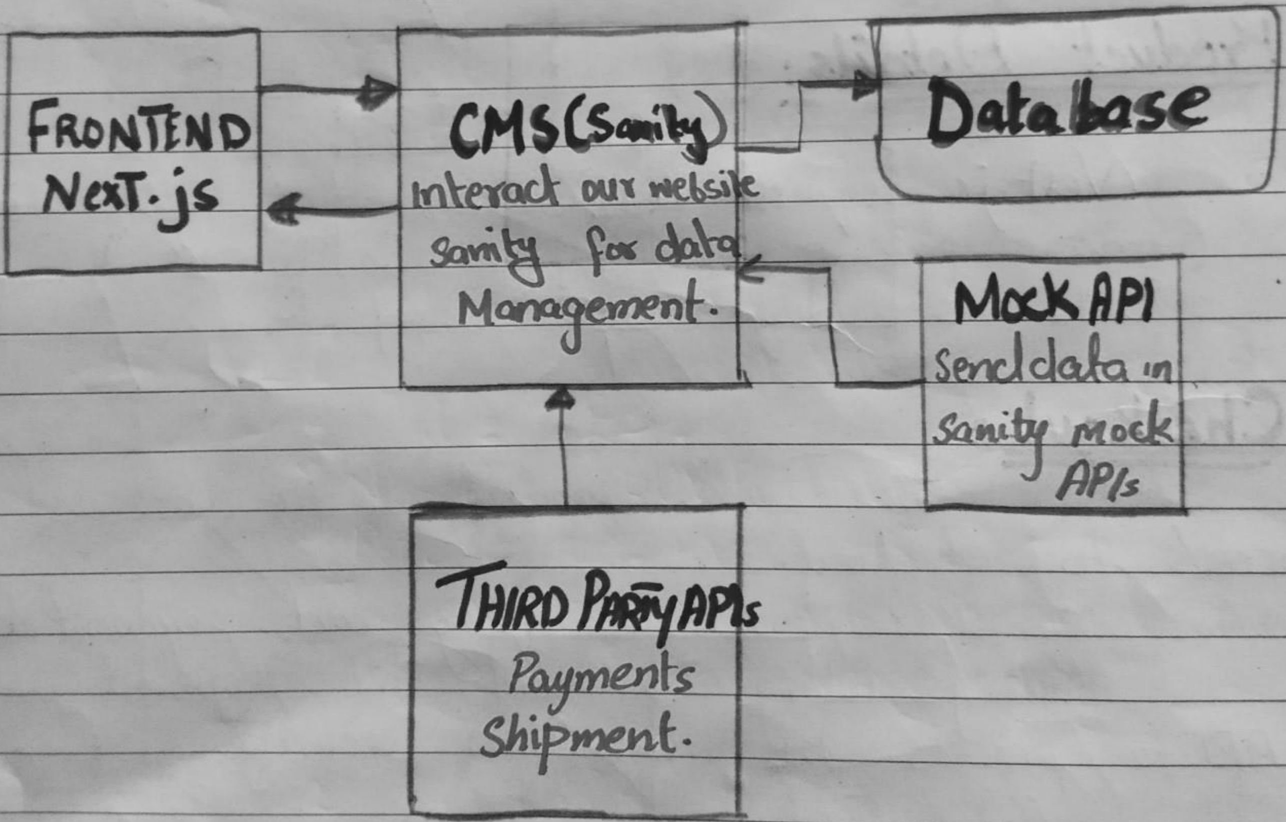
After payment confirmation.

Next, js sends order detail to Sanity CMS for Storage.

An order confirmation is displayed to the user.

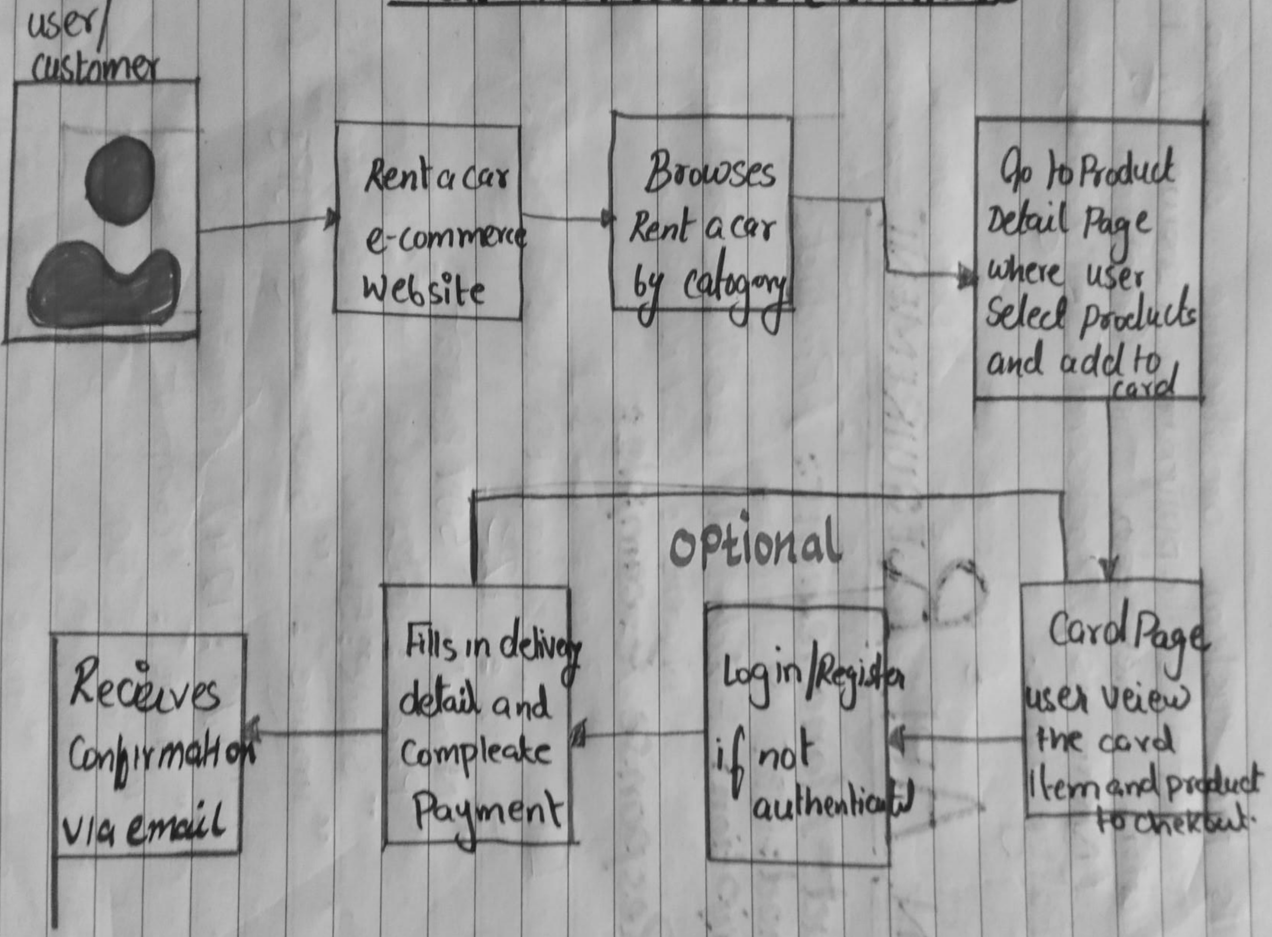
8. **Shipment Tracking:** Shipment tracking API provides delivery updates to the frontend and notification to the customer.

"SYSTEM ARCHITECTURE DIAGRAM"


FRONTEND
Next.js
CMS(Sanity)
interact our website sanity for data Management.
Database
Mock API
Send data in sanity mock APIs
THIRD PARTY APIs
Payments
Shipment.

How the User/Costomer interact with our Website

Rent a Car Website (Workflow)


user/customer
Rent a car e-commerce website
Browses Rent a car by catogory
Go to Product Detail Page where user Select products and add to card
optional
Card Page user veiew the card Item and product to chekout.
Log in/Register if not authenticated
Fills in delivery detail and compleate Payment
Receives Confirmation via email

Here is the system architecture diagram for your market place application, showcasing the interaction between the frontend (Next.js), Backsend (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 Excample:

```
[
{ "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 produc

Payload Example:

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

Response Example.

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

4. Order Management

Endpoint Nome: /orders

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

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

"ProductDetails":
[{"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 }] }
```