

STUDENT NAME : AHMED RAZA

Roll NO: 00102147

DAY 02 : Technical Foundation

STEP 01 : Define Technical requirement-

Fronted Requirements :

- 1 User-friendly interface for browsing and renting products-
- 2 Responsive design for mobile and desktop -
- 3 Key Pages : Home , Product listing , product details , Rental card , checkout , order tracking-

Backened Requirement (Sanity cms)

- Product data managment (name , description , availability , rental terms) -
- Customer profile and order history-
- Rental order records and payment processing-

## Third Party Integrations:

- Payment gateway (Stripe, Paypal) for secure transactions-
- shipment tracking API for delivery status-

## STEP 02: Design system Architecture-

### System Components-

#### 1- Frontend (Next.js):

Display product listings, processes orders, and handles user interactions.

#### 2- Backend (Sanity CMS):

Store and manages product, user and other data-

#### 3- Third-party APIs:

Handle payments, shipping and authentication

### Overview

[Users] → [Frontend (Next.js)] → [Sanity CMS] →

[Payment Gateway] → [shipment API]

## Key Workflows -

- 1- User Registration : User sign up → Data stored in sanity cms-
- 2- Product browsing : Product fetched from sanity → Displayed on Next.js frontend-
- 3- Order processing : User rents a product → payment processed → order stored in sanity cms
- 4- Shipment tracking : Order status fetched from API → Displayed to the user-

## STEP 03 : Plan API Requirements.

### Essential API Endpoints.

- 1- Fetch Available Products.
  - Endpoint : /Products
  - Method : GET
  - Response : { "id": 1 , "name": "camera", "Price": 20 , "availability": "In stock" }
- 2- Create Rental Order.
  - Endpoints : / rental-orders.
  - Method : POST

(3)



- Payload : { "ProductId": 1 , "UserId": 123 , "rental period":  
"7 days" }

- Response : { "orderId": 456 , "status": "confirmed" }

### 3. Process Payment:

- Endpoint : / Payments

- Method : POST

- Payload : { "orderId": 456 , "amount": 100 ,  
"Payment method": "stripe" }

- Response : { "PaymentId": 789 , "status": "success" }

### 4 Track shipment

- Endpoint : / shipment - status .

- Method : GET

- Response : { "orderId": 456 , "status": "Out for delivery",  
"ETA": "2 hours" }

## STEP 04: Technical Documentation

system architecture overview : Daigram and component description-

API documentation:

list of endpoints, methods, payloads and expected reasons-

Workflow Description:

steps from user interaction to backend processing-

Data Schema:

Entities and relationship stored in sanity cms-

STEP 05: Collaborate and Refine

- Discuss API designs and workflows with mates for feedback.
- Ensure technical alignment with business objective.

X ————— X

(5)