**Day 3 Task: API Integration and Data Migration**

**1. API Integration Process**

1. **Setup**:
   * Used **Next.js** for frontend pages.
   * Integrated APIs for tasks like fetching products, uploading files, and creating orders.
   * Connected **Sanity CMS** to manage data like user orders and video information.
2. **Steps**:
   * Fetch data from an API:

const res = await fetch('https://api.example.com/products');

const data = await res.json();

Send user uploads (like files):

const formData = new FormData();

formData.append('file', file);

const res = await fetch('/api/upload', { method: 'POST', body: formData });

 **Video and Order Management**:

* After a user submits a request:
  + Save the order in **Sanity CMS**.
  + Trigger video creation using backend APIs.

### ****2. Adjustments to Sanity Schemas****

* **Product Schema**: To store video-related data.

export default {

name: 'product',

type: 'document',

title: 'Product',

fields: [

{ name: 'title', type: 'string' },

{ name: 'description', type: 'text' },

{ name: 'price', type: 'number' },

],

};

**Order Schema**: To track user orders.

export default {

name: 'order',

type: 'document',

fields: [

{ name: 'user', type: 'reference', to: [{ type: 'user' }] },

{ name: 'status', type: 'string' },

],

};

### ****Migration Steps****

1. **Backup**: Export old data from Sanity:

sanity dataset export production ./backup

**Update Schema**: Deploy changes:

sanity deploy

**Fix Old Data**: Run a script to update old records (e.g., add status):

orders.forEach(async (order) => {

await client.patch(order.\_id).set({ status: 'In Progress' }).commit();

});

### ****4. Screenshots****

### 

User video message

### ****5. Example Code Snippets****

* **Fetch Products**:

const res = await fetch('/api/products');

const products = await res.json();

**Save Order**:

await client.create({

\_type: 'order',

user: userId,

status: 'Pending',

});