Solution for Day 3: **API Integration and Data Migration**

1. Understand the Provided API
Use the provided API endpoints or external sources for data integration. Test endpoints like:

Endpoint: /products

- Fetch product details.
- Response example:

```
{
"id": 1,
"product_name": "Laptop",
"price": 1000,
"stock": 10,
"category": "Electronics"
}
```

- Steps to Test:
  1. Use Postman or browser DevTools to test endpoints.
  2. Confirm data structure (e.g., field names, types) aligns with your schema.

2. Validate and Adjust Your Schema
Compare your Sanity CMS schema from Day 2 with the API data structure. Adjust field names or types as needed.
Example Schema Adjustment: If the API returns product_title but your schema uses name, modify your schema or handle mapping during migration.

Updated Sanity Schema Example:

```
export default {
name: 'product',
type: 'document',
fields: [
    { name: 'name', type: 'string', title: 'Product Name' }, // Maps to `product_name`
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'category', type: 'string', title: 'Category' },
],
};
```

3. Data Migration
Choose a migration method:

1. Using the Provided API:
   - Write a script to fetch and transform data into Sanity CMS.
2. Manual Import:
   - Export API data as JSON/CSV and import it using Sanity tools.

Migration Script Example:

```
const sanityClient = require('@sanity/client');
const fetch = require('node-fetch');

const client = sanityClient({
projectId: 'yourProjectId',
dataset: 'production',
useCdn: false,
token: 'yourSanityToken',
});

async function migrateProducts() {
```

```
    try {
        const response = await fetch('https://example.com/api/products');
        const products = await response.json();

        products.forEach(async (product) => {
        await client.create({
            _type: 'product',
            name: product.product_name,
            price: product.price,
            stock: product.stock,
            category: product.category,
        });
        });

        console.log('Migration completed successfully!');
    } catch (error) {
        console.error('Error during migration:', error);
    }
    }

    migrateProducts();
```

4. API Integration in Next.js
Steps to Integrate APIs:

1. Create Utility Functions:
Fetch data from APIs and handle errors.

Utility Function Example:

```
        export async function fetchProducts() {
        try {
            const response = await fetch('/api/products');
            return await response.json();
        } catch (error) {
            console.error('Error fetching products:', error);
            return [];
        }
        }
```

2. Render Data in Components:
 - Use React hooks to display API data.

React Component Example:

```
        import { useEffect, useState } from 'react';
        import { fetchProducts } from '../utils/api';

        const ProductList = () => {
        const [products, setProducts] = useState([]);

        useEffect(() => {
            fetchProducts().then(setProducts);
        }, []);

        return (
            <div>
            {products.map((product) => (
                <div key={product.id}>
                <h3>{product.name}</h3>
                <p>Price: ${product.price}</p>
                <p>Stock: {product.stock}</p>
                </div>
            ))}
```

```
            </div>
        );
        };

        export default ProductList;
```
3. Test API Integration:
    - Use Postman or browser tools to verify endpoint responses.
    - Log API responses to ensure data consistency.

5. Error Handling

Best Practices:
    - Log errors centrally for debugging.
    - Display user-friendly error messages on the frontend.
    - Use fallback data or loaders for better user experience.

6. Expected Output

1. Sanity CMS:
    - Populated with data from the API or external source.
Example:
    - Product: Laptop
    - Price: $1000
    - Category: Electronics

2. Frontend:
    - Dynamic product listings displayed on the homepage or product page