Day-2

1) Tecnical Requirements:

- Frontend:

  * Use Next.js for creating Pages
  * Pages to include

    # Home # Product listing #Product detail
    # Cart # Checkout #Order confirmati

  * Ensure responsive & user-friendly design

- Backend:

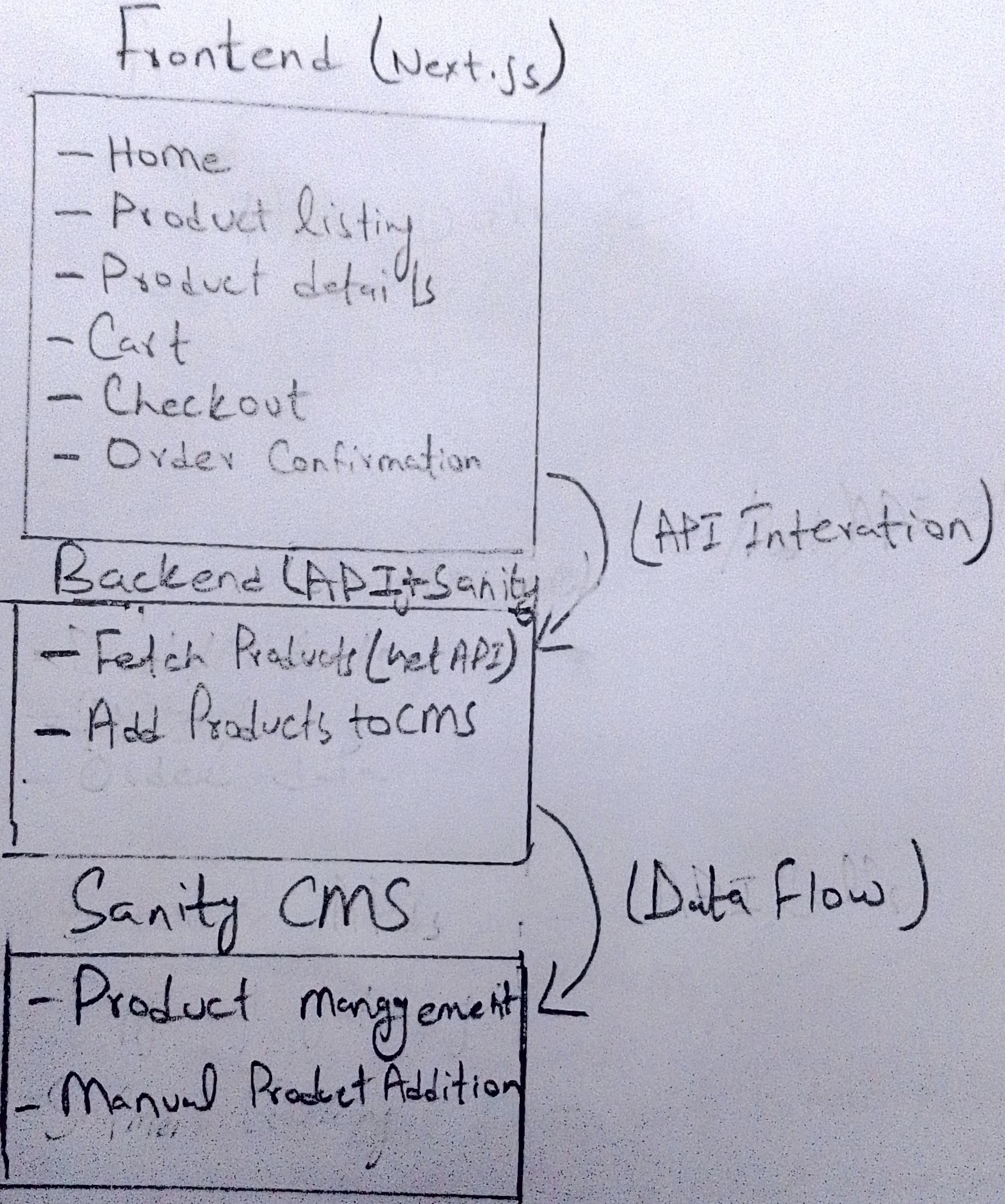
  * Use Sanity CMS for:

    # Managing Product Data
    # Storing Customer details & Orders.

  * Integrate APIs for:

2) System Architecture:

- Frontend (Next.js): Handles user interactions.
- Sanity CMS: Manages & stores data.
- APIs: Enables Payment Processing & Shipment tracking



Frontend (Next.js)

- Home
- Product listing
- Product details
- Cart
- Checkout
- Order Confirmation

(API Interation)

Backend (API + Sanity

- Fetch Products (het API)
- Add Products to CMS

(Data Flow)

Sanity CMS

- Product manggement
- Manual Product Addition

Interection

- fetch Product data from APIs (once) → Import into Sanity
- frondend Fetches Product from sanity CMS for listing & details

3) API Requirements:

- List of API endPoints:
  - * Get/Products: fetch Product list

4) Sanity CMS Schemas:

- Fields
  - * Product name
  - * Price
  - * Stock level
  - * Description
  - * Reviews
  - * Image

5) Key Workflows:

- Use API Once to import Product data into Sanity CMS
- Fetch Product data from sanity CMS & display on frontend
- Basic cart functionality to add Products
- Minimal workflow to collect order details & store in sanity cms