DAY 2: PLANNING THE TECHNICAL FOUNDATION

Submission Guidelines

- **Title: "Marketplace Technical Foundation - [E-commerce Shoes]"**

Goal

The primary objective of Day 2 is to transition from business-oriented planning to technical preparation required to build the eCommerce marketplace. This involves creating a high-level technical plan, including system architecture, workflows, API requirements, and Sanity CMS integration.

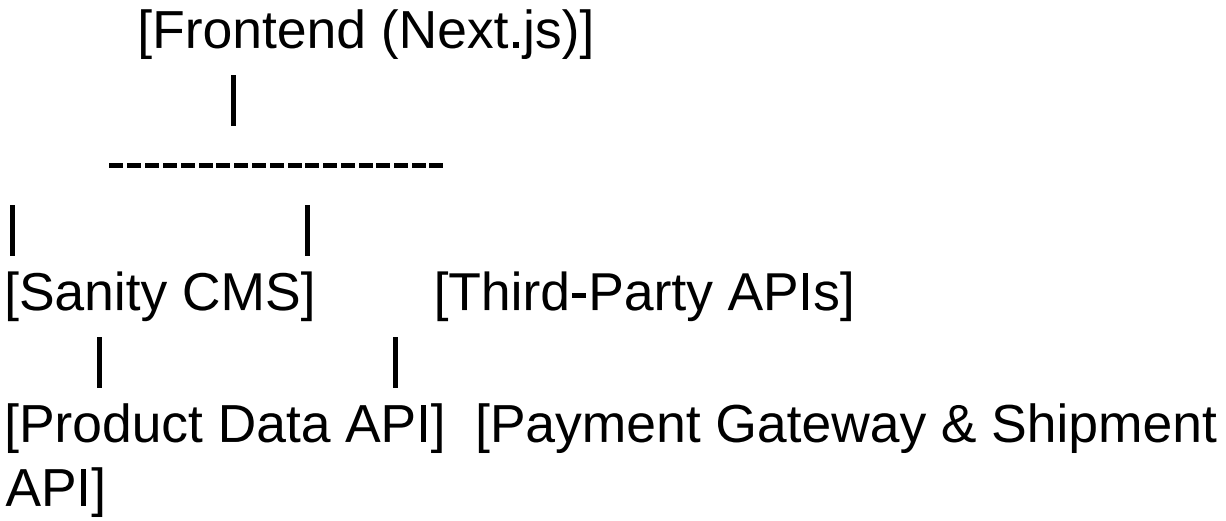
Technical Requirements Frontend Requirements

- **User Interface**: Intuitive, user-friendly interface for seamless product browsing and shopping experience.
- **Responsive Design**: Optimized for both mobile and desktop.
- **Essential Pages**: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation. **Sanity CMS (Backend)**
- **Data Management**: Handle product data, customer information, and order records.
- **Schema Design**: Align data schemas with business goals, focusing on product listings, inventory, pricing, and customer details.

Third-Party APIs

- **Integrations**: Shipment tracking, payment gateways, and other essential services.
- **Data Flow**: Ensure APIs deliver required data for frontend functionalities, including product details, order processing, and payment.

System Architecture



Data Flow Example

- **Product Browsing**: Frontend requests product data from Sanity CMS to display in the product listing page.
- **Order Placement**: Order details submitted to Sanity CMS for processing, including customer info and order contents.
- **Payment Processing**: Payment gateway processes transactions and updates order status.
- **Shipment Tracking**: Data fetched from third-party APIs and displayed to users, with real-time tracking information.

API Requirements

General eCommerce APIs

- **/products (GET)**: Fetch all available products.
    - Response: `{ "id": 1, "name": "Product A",`

"price": 100 }

- **/orders (POST)**: Create a new order. ◦ Payload: { "customerId": "123", "items": ["Product A", "Product B"] }
- **/shipment (GET)**: Track order status.
  - Response: { "orderId": 123, "status": "In Transit", "ETA": "2 days" } **Technical Documentation**

1. **System Architecture Document** ◦ Overview: Describe how frontend, Sanity CMS, and thirdparty APIs interact.
   ◦ Diagrams: Clearly labeled to visualize data flows.
2. **API Specification Document** ◦ Details: Endpoints, methods, payloads, responses for product data, orders, payments, and shipment tracking.
3. **Workflow Diagrams** ◦ Visual representation of user interactions and system processes from browsing to checkout and delivery.
4. **Data Schema Design** ◦ Define key entities like products, orders, customers, and inventory management.
5. **Technical Roadmap** ◦ Outline implementation steps, milestones, and deadlines.

Key Outcomes

- **Aligned Technical Plan**: Tailored for specific eCommerce marketplace needs.
- **Visualized Architecture**: Comprehensive diagrams and workflows.
- **Documented APIs**: Clear, detailed API endpoints for smooth interactions.
- **Drafted Sanity Schemas**: Efficient data management structures for inventory and customer data.

- **Collaborative Feedback**: Incorporated for improved quality and usability.

Industry Best Practices

- **Plan Before Coding**: Reduce rework with clear technical plans.
- **Use Efficient Tools**: Leverage Sanity CMS and third-party APIs.
- **Collaborate**: Regular peer reviews for continuous improvement.
- **Focus on UX**: Ensure a seamless and intuitive shopping experience.