

Day 2: Dollar Bazaar ka Technical Foundation Plan

1. Day 1 Goals ka Recap:

- **Masla Jo Solve Karna Hai:** Online shopping ko affordable banaya jaye, customers ko \$1 deals wali shops se connect karke, bina physical visit ki zarurat ke.
- **Target Audience:** Middle-class aur low-income families jo budget-friendly shopping options dhoond rahi hain.
- **Unique Value Proposition (UVP):** Ek aisi platform jo sirf \$1 products ko showcase kare trusted stores se, jo convenience aur affordability dono provide kare.

2. Technical Requirements Define Karna:

Frontend Requirements:

- **User Interface (UI):** Simple aur user-friendly design, jo har customer asaani se samajh sake.
- **Responsive Design:** Mobile aur desktop dono devices ke liye compatible.
- **Zaroori Pages:**
 - Homepage jahan featured \$1 deals ho.
 - Product listing page jahan filtering ka option ho.
 - Product detail page jahan images, descriptions aur reviews ho.
 - Cart aur checkout pages jo secure payment options ke sath ho.
 - Order confirmation aur tracking page.

Backend Requirements:

- **Data Management:** Sanity CMS use karna jo product data, shop details aur customer orders manage kare.
- **Order Processing:** Real-time order status updates ka system.
- **Security:** Secure user authentication aur payment processing.

Third-Party APIs:

- **Payment Gateway** (e.g., Stripe ya PayPal).
- **Shipment tracking API.**

3. System Architecture:

High-Level Overview:

[Frontend (Next.js)]

|

[Sanity CMS] ---> [Product Data API]

|

[Payment Gateway] ---> [Shipment Tracking API]

Workflows:

- User Browsing:

1. User platform visit karta hai.
2. Frontend, Sanity CMS se product data fetch karta hai.
3. Products dynamically display hoti hain.

- Order Placement:

1. User items cart mein add karta hai aur checkout karta hai.
2. Order details Sanity CMS ko bheji jati hain.

3. Payment gateway ke zariye process hota hai.
4. Shipment details API se fetch karke user ko display hoti hain.

4. API Requirements Plan Karna:

Endpoints:

- GET /products
 - Sab \$1 products fetch karo.
 - Response: { id, name, price, stock, imageURL }
- POST /orders
 - Ek naya order place karo.
 - Payload: { customerInfo, productDetails, paymentStatus }
 - Response: { orderId, confirmationStatus }
- GET /shipment
 - Shipment status fetch karo.
 - Response: { shipmentId, status, expectedDelivery }

5. Technical Documentation:

System Architecture Document:

- Clear interaction diagram.
- Har component ka role describe karna.

API Specification Document:

- Detailed endpoints, methods, payloads, aur responses.

Data Schema Design:

- **Product:** { id, name, price, stock, image }
- **Order:** { orderId, customerInfo, productDetails, status }
- **Shipment:** { shipmentId, orderId, status, expectedDelivery }

6. Future Plans:

- **Scalability:** Zyada cities tak expand karna.
- **Strong Delivery System:** Reliable delivery network jo timely aur safe delivery ensure kare.
- **Flash Sales:** Exclusive deals aur discounts ka system introduce karna.

Day 2 Ka Goal:

Ensure karna ke technical foundation business goals ke sath align ho, aur ek strong framework provide kare implementation aur scalability ke liye.