DAY 2 PLANNING THE TECHNICAL FOUNDATION

—

By uzma mumtaz
Thursday 9 to 12

System Architecture

Frontend (Next.js)

Sanity CMS--------> Product Data API --------> Database (Product Data)

Third-Party API ----> [Shipment Tracking API] ----> Shipping Provider API

Payment Gateway ----> Payment Processing API ----> Transaction Database

Explanation:

1.Frontend (Next.js): The user-facing part of the application, built using Next.js, is responsible for handling the user interface, displaying product information, managing the shopping cart, etc.

2.Sanity CMS: Acts as the content management system (CMS) that stores and manages product details, descriptions, media, and other content. This is connected to the frontend for dynamic data retrieval.

3.Product Data API: This API is used to retrieve product-related data from your backend. It connects with the Product Data stored in your database.

4.Database (Product Data): Holds all the product data like names, prices, images, and other relevant information.

5.Third-Party API: Provides additional functionalities such as external services (like user authentication, social media sharing, etc.) that integrate into your app.

6.Shipment Tracking API: Tracks shipment statuses and updates order delivery information. It connects to third-party Shipping Provider APIs to retrieve tracking information.

7.Shipping Provider API: Provides real-time shipment tracking information (e.g., UPS, FedEx).

8.Payment Gateway: Integrates with your app to handle payments securely, allowing users to complete transactions.

9.Payment Processing API: Facilitates the communication between your frontend and payment gateway to securely process payment information.

10.Transaction Database: Holds records of transactions, user payments, and order histories for future reference or queries.

Api requirement with endpoints...........

Endpoint	Method	Description
/products	GET	all available products from Sanity.
/orders	POST	Create a new order in Sanity.
/shipment	GET	Fetch shipment tracking data from third-party
/cart	GET	Retrieve current cart items for logged-in user.
/cart	POST	Add item(s) to the cart.
/user/login	POST	User login for authentication.
/user/register	POST	User registration for creating an account.
/reviews	POST	Submit a product review and rating.
/reviews/:productId	GET	Fetch all reviews for a specific product.

Shipment Schema

Shipment id   (unique id from shipment)
Orderid   (order id from checkout)
Userid   (user who placed the order)
shipmentAddress (user address details)
Confirmation   (confirmation msg)

WORKFLOW..............

User Registration:

User signs up → Data is stored in Sanity → Confirmation sent to the user.

Login:

User logs in → Authentication request sent to backend → User authenticated → Session created → User redirected to their dashboard.

Product Browsing:

User views product categories → Sanity API fetches product data → Products displayed on frontend.

Cart Management:

WORKFLOW..............

Cart Management:
User adds items to the cart → Cart updated with product details → User proceeds to checkout.

Order Placement:
User adds items to cart → Proceeds to checkout → Order details saved in Sanity → Order placed.

Payment Gateway:
User enters payment details → Payment processed → Confirmation sent to user → Order status updated.

WORKFLOW..............

Product Review & Rating:
User leaves a review and rating for a product → Review and rating stored in Sanity → Displayed on the product page for others.

Order Placement:
Shipment Tracking:

Payment Gateway:
User enters payment details → Payment processed → Confirmation sent to user → Order status updated.

Flowdiagram

user

resturant web landing page
shop

all product detail from sanity backend

cart   cart

dynamic pages product detail
add to cart

cart page
qty price tota
4   200   5000l

customer info
name
adress
phone

Delivery and payment details
trd

order done confrmation by phone