Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2025TMID56607
Project Name	ShopSmart: Your Digital Grocery Store
Maximum Marks	4 Marks

Technical Architecture:

Table 1: Technology Stack

Component	**Technology**	**Purpose**
Frontend	**React / Next.js**	**A modern JavaScript library for building a fast, responsive, and interactive user interface.**
Web Server	**Nginx**	**To serve the static frontend application files efficiently.**
API Gateway	**Amazon API Gateway / Kong**	**To manage, secure, and route all incoming API requests to the appropriate backend service.**
Backend Services	**Node.js (Express.js)**	**For building fast, scalable, and I/O-intensive microservices for handling users, products, and orders.**
Databases	**PostgreSQL & MongoDB**	**PostgreSQL: For structured, transactional data like user profiles and order details. MongoDB: For flexible, semi-structured data like product catalogs with varying attributes.**
Cloud Platform	**Amazon Web Services (AWS)**	**A comprehensive cloud provider for hosting all components, offering services like EC2, S3, RDS, and Lambda.**
Payment Gateway	**Stripe / PayPal**	**A third-party service to securely handle all payment card processing, reducing PCI compliance scope.**

Notification Service	AWS Simple Email Service (SES)	To reliably send transactional emails for order confirmations, password resets, etc.

Table 2: Data Flow Description (Online Order Processing)

Step	Action	Data Flow Description
1	Add to Cart	The customer selects a product. The Frontend sends an API request via the API Gateway to the Order Service to add the product ID and quantity to the customer's session or cart record in the Order Database.
2	Checkout	The customer initiates checkout. The Frontend requests the cart contents from the Order Service and the customer's saved address from the User Service.
3	Enter Details	The customer confirms their delivery address and time slot. This data is sent to the Order Service to be associated with the pending order.
4	Submit Payment	The customer enters payment details. The Frontend securely tokenizes this information and sends it to the Payment Service.
5	Process Payment	The Payment Service forwards the payment token and order total to the external Payment Gateway. The gateway processes the transaction and returns a success or failure response.
6	Create Order	Upon successful payment, the Payment Service notifies the Order Service. The Order Service finalizes the order, saves it to the Order Database, and updates product inventory (via the Product Service).
7	Confirmation	The Order Service triggers the Notification Service to send a detailed order confirmation email to the customer. The Frontend displays an order success page.

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/