Day 2 Activities: Transitioning to Technical Planning

For e-commerce website Figma Template:
https://figma4template.vercel.app/

1. <mark>Define Technical Requirements:-</mark>
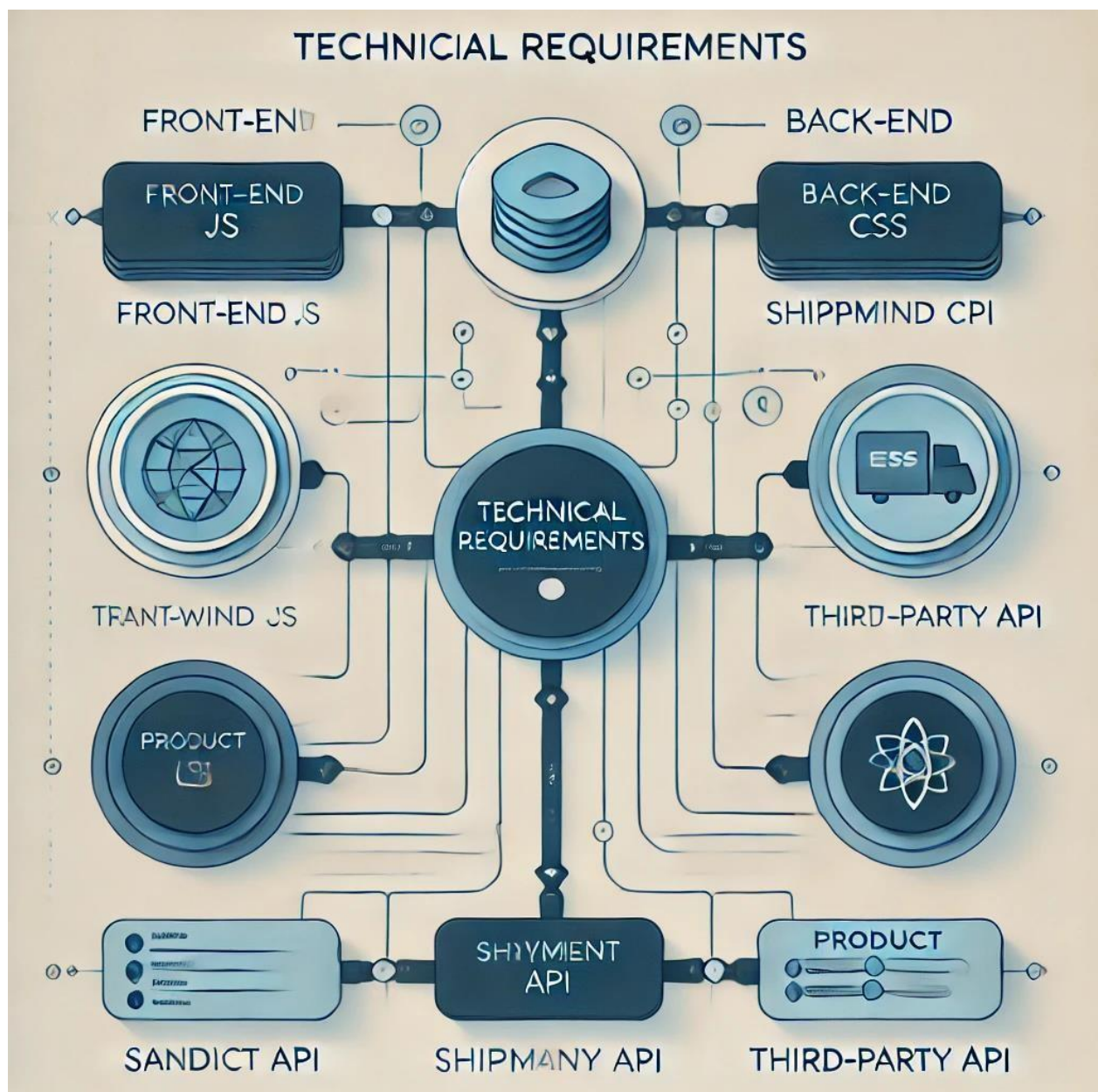
Frontend Requirements   (NEXT.JS + TAILWIND CSS)

User-Friendly Interface

- Intuitive and visually appealing design to enhance user experience.
- Features like:
    - Search bar for easy product discovery.
    - Category filters for refining searches.

Responsive Design

- **Flexible layouts** and optimized images to improve load time and usability.
- Use libraries like **Tailwind CSS** for responsiveness.

Purpose:

○ SSR (Server-Side Rendering) for better SEO and performance.

○ Dynamic routing for pages like products, categories, and user profiles.

Pages:

1) Landing Page with all its product components.
2) Shop Cart
3) Shop List
4) Shop Grid Default
5) Order Complete
6) Products
7) Product Details
8) Check out
9) Shipping

Sanity CMS as Backend:

Sanity CMS provides a powerful and flexible solution for managing product data, customer details, and order records. Here's how you can leverage it for your marketplace:

Key Features

Centralized Content Management:

Real-time Updates:

Scalable Schemas:

API:

Use Sanity's GraphQL and GROQ queries to fetch data efficiently.

○ Product API

○ Shipment API

○ Payment API

Third-Party APIs:

Integrating third-party APIs ensures that your marketplace has the necessary functionality for shipment tracking, payments, and other backend services

1. Shipment Tracking APIs Integration
Features:
- Fetch tracking details using an order's shipment ID.
- Provide real-time updates on delivery status.
- Show estimated delivery dates on the frontend.

2. Payment Gateway APIs
To enable secure and seamless transactions**.**
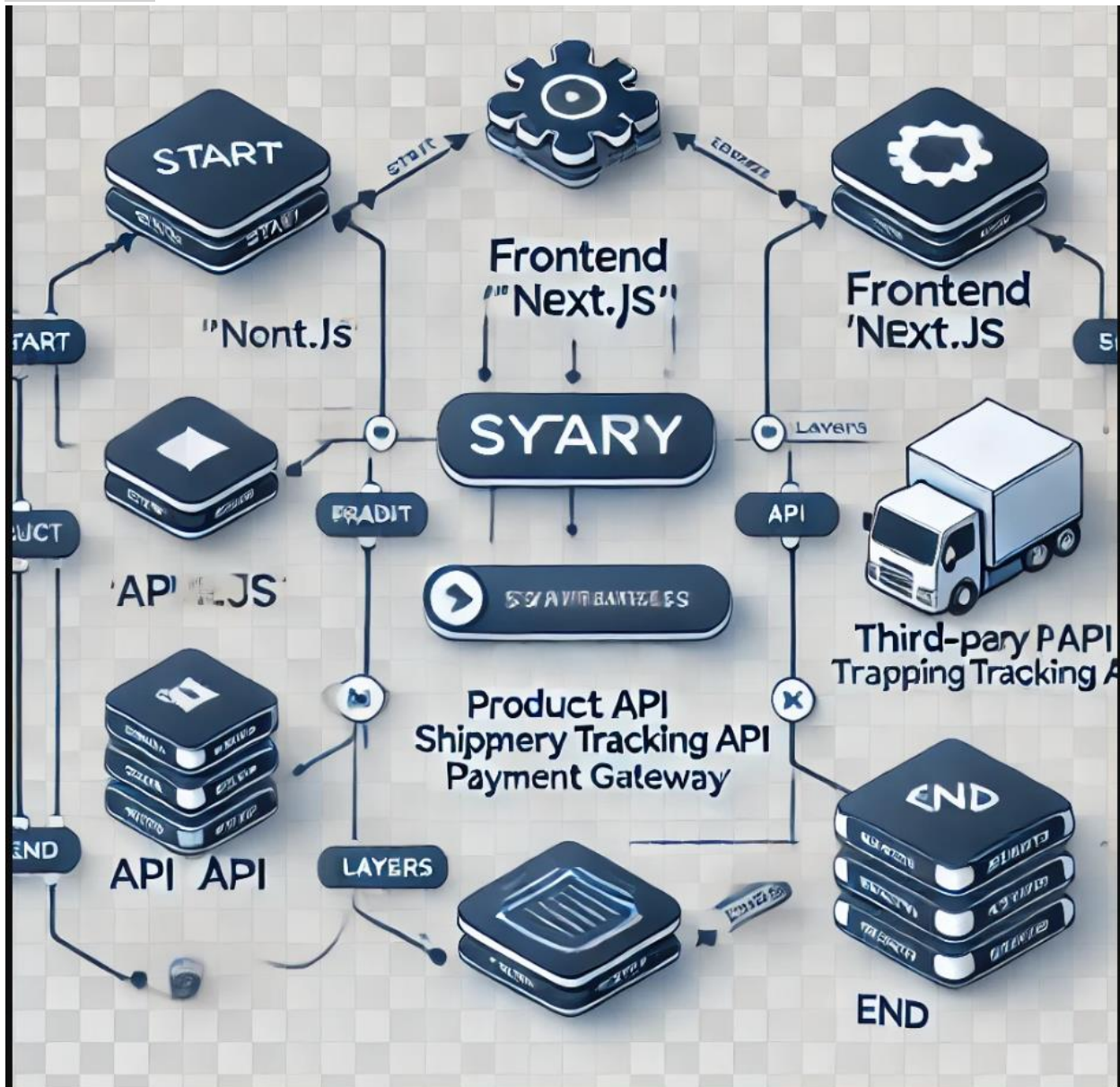#### Features:
- Support for various payment methods (credit cards, wallets, UPI, etc.).
- Payment status updates for the order (e.g., Pending, Paid, Failed).

- Webhooks for real-time updates on payment success/failure.

SystemArchitecture OverView:

Diagram:

2: Design System Architecture

- [Frontend (Next.js)] -----> [Sanity CMS]
- | | | | [Payment Gateway] [Third-Party APIs]

- | | [User Notifications] <-> [Shipment Tracking]

3: Plan API Requirements:

APIEndPoints.xlsx

	Endpoint	Method	Purpose	Response Example
2	/api/register	POST	Register a new user account.	{ "message": "User registered successfully", "userId": "12345" }
3	/api/login	POST	Log in a user and generate an authentication token.	{ "token": "abcdef12345", "userId": "12345" }
4	/api/logout	POST	Log out a user and invalidate the token.	{ "message": "User logged out successfully" }
5	/api/products	GET	Fetch all available products.	[{ "id": "1", "name": "Product A", "price": 100 }]
6	/api/products/{id}	GET	Fetch details of a specific product by its ID.	{ "id": "1", "name": "Product A", "price": 100, "details": "Product details..." }
7	/api/cart	GET	Fetch the current items in the user's cart.	[{ "productId": "1", "name": "Product A", "quantity": 2 }]
8	/api/cart	POST	Add items to the user's cart.	{ "message": "Product added to cart", "cart": { "productId": "1", "quantity": 2 } }
9	/api/checkout	POST	Proceed to checkout and create an order.	{ "orderId": "ORD12345", "status": "Order created" }
10	/api/shipment	GET	Track the status of a shipment.	{ "shipmentId": "SHIP12345", "status": "In Transit", "expectedDelivery": "2025-01-25" }
11	/api/payment	POST	Process payment for an order.	{ "paymentId": "PAY12345", "status": "Success" }
12	/api/order-status	GET	Retrieve the status of an order by its ID.	{ "orderId": "ORD12345", "status": "Processing" }
13	/api/orders	GET	Fetch a list of all orders for a user.	[{ "orderId": "ORD12345", "status": "Delivered", "totalAmount": 100 }]
14				
15				

WORK FLOW DIAGRAM: