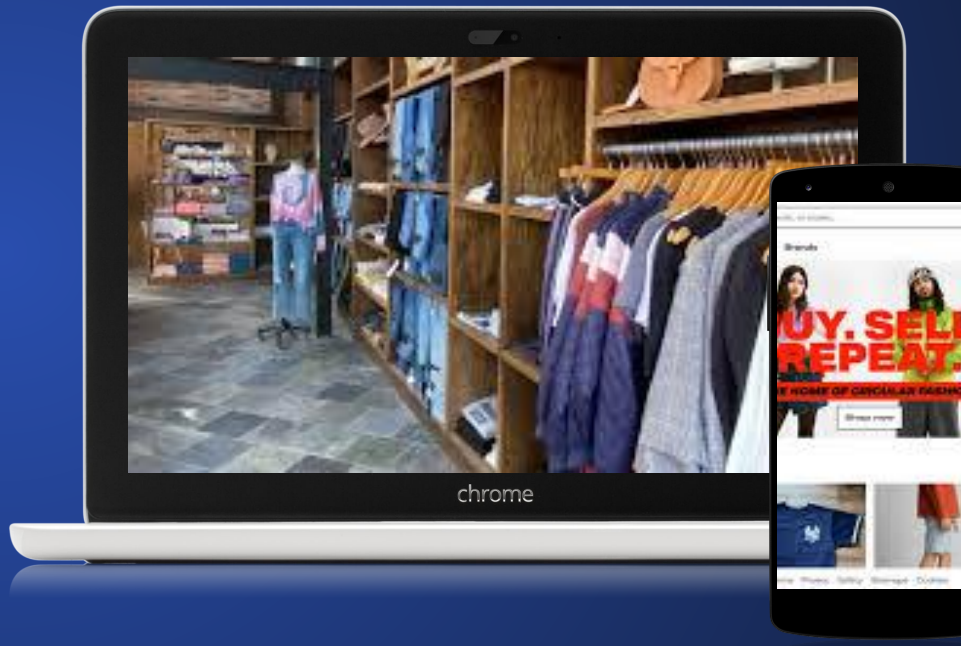
Prepared by
Alishba Majeed

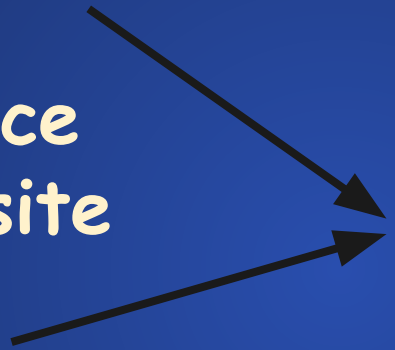
Hackathon Day 2

The Technical Foundation Planning



OutLines

My Marketplace Clothing Website



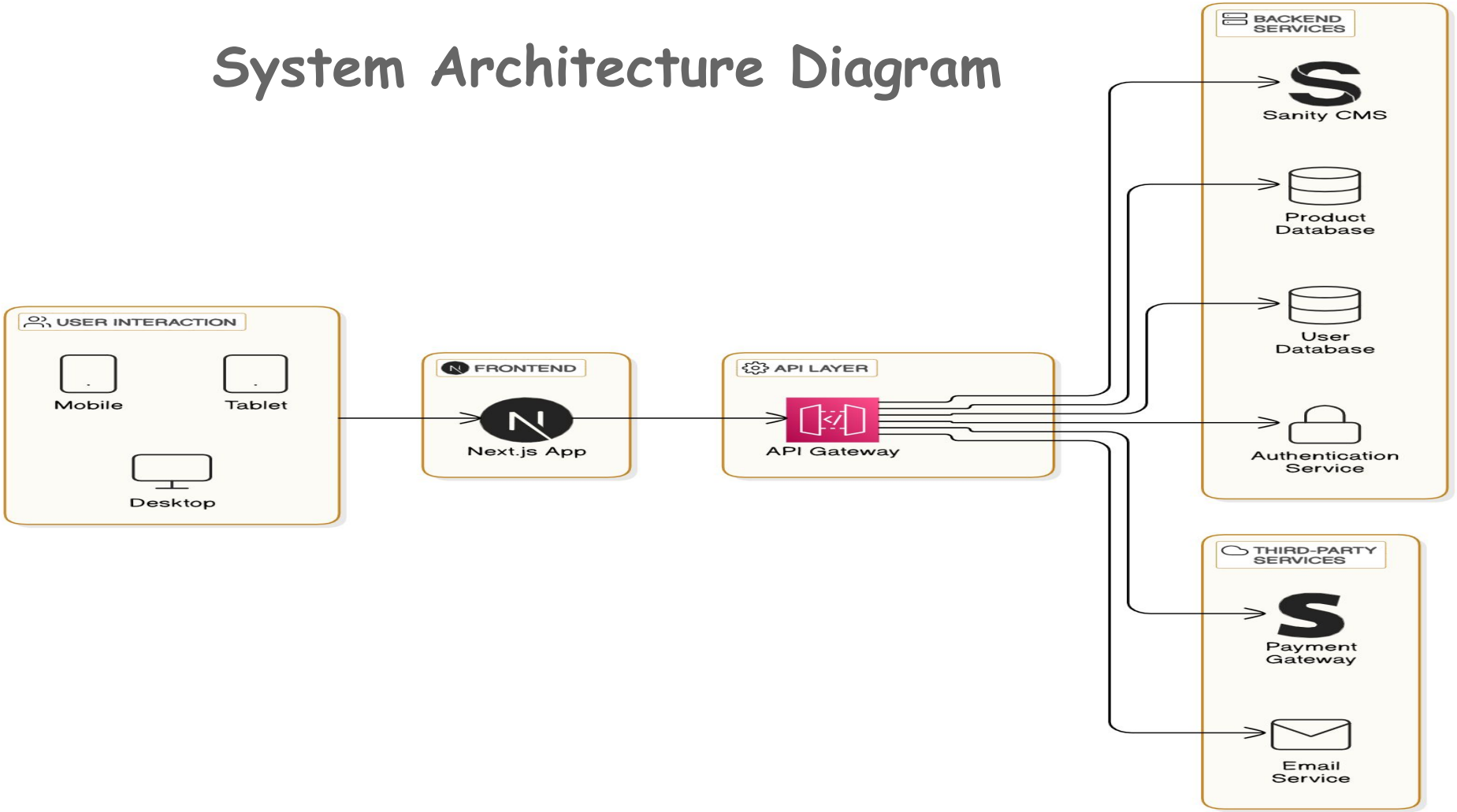
- Overview
- System Architecture
- Frontend Development Plan
- Backend Development Plan
- Integration and Workflow
- Conclusion

Technical Plan for My Marketplace Clothing Website

Overview :

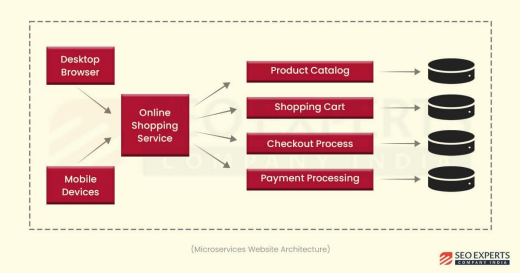
My Technical Plane is to design my Clothing MarketPlace Website in such a way that it is efficient Scalable, and user-friendly. I want the Website's structure to perfectly align with my Business Goals where i convert business Requirements into technical solution.

System Architecture Diagram



System Architecture for Marketplace Clothing Website

The marketplace architecture is designed to ensure scalability, performance, and security, while providing a seamless experience for users, admins, and third-party integrations. Here's the detailed structure:

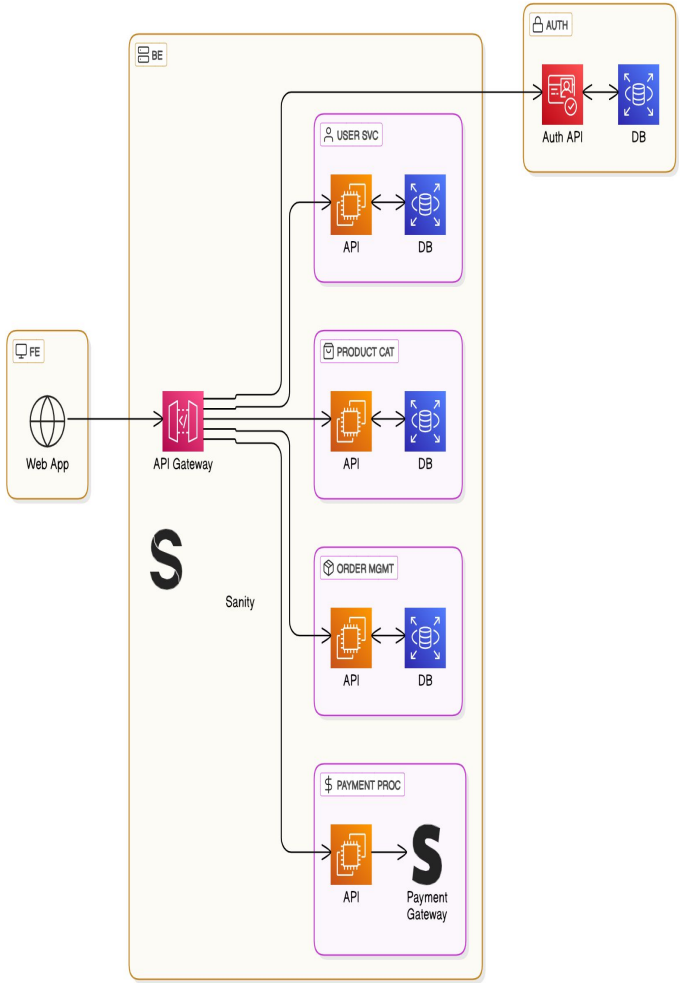


Key Components:

1. **Frontend:**
   - Next.js: Handles server-side rendering (SSR) and client-side rendering (CSR) for dynamic, fast-loading pages. This ensures better SEO and user experience.
   - Tailwind CSS: A responsive, mobile-first design framework used to create modern, visually appealing interfaces.
   - Features:
     - Intuitive navigation with category filters and product search.
     - Interactive user dashboards for orders and profiles.
     - Optimized checkout and cart functionalities.

Backend:

- **Sanity CMS:**
  - Centralized content management for product details, categories, banners, and user reviews.
  - Easy updates to product information and other content through an admin-friendly interface.
- **API Layer:**
  - REST or GraphQL APIs connect the frontend and backend, ensuring seamless data exchange.
- **Order Management:**
  - Backend services process orders and store shipping information, ensuring a smooth purchase experience.



Sanity Database Structure for Marketplace Clothing Website

The database structure ensures efficient content management, scalability, and fast data retrieval. Below is the breakdown of key components:

1. **Users**
   - Stores user information including accounts, addresses, and preferences.
   - Schema Fields: Full name, email, role, addresses, and subscription preferences.
2. **Products**
   - Manages inventory, descriptions, pricing, categories, and product images.
   - Schema Fields: Title, description, price, inventory, images, and category references.
3. **Orders**
   - Tracks user purchases with details on products, payment status, and order progress.
   - Schema Fields: User reference, product references, total price, payment details, and status.
4. **Categories**
   - Defines product groups for better organization.
   - Schema Fields: Name, description, and category image.
5. **Reviews (Optional)**
   - Allows users to review products with ratings and comments.
   - Schema Fields: User reference, product reference, rating, and comment.

Payment Gateway:

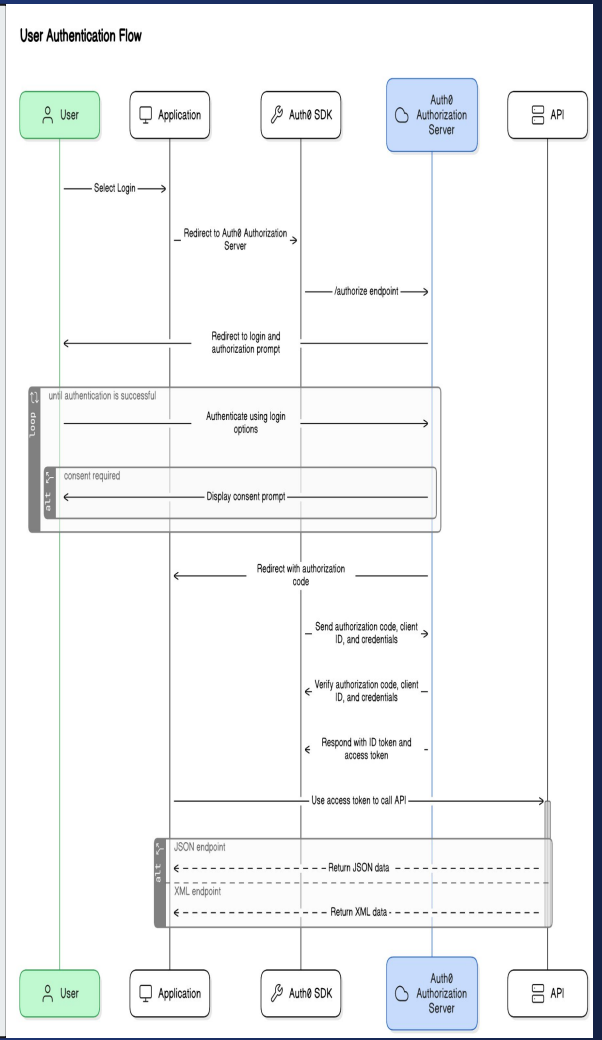
- **Stripe API:**
  - Manages secure online payments, supporting multiple payment methods like cards and wallets.
  - Ensures PCI compliance and fraud protection.

Authentication & Authorization:

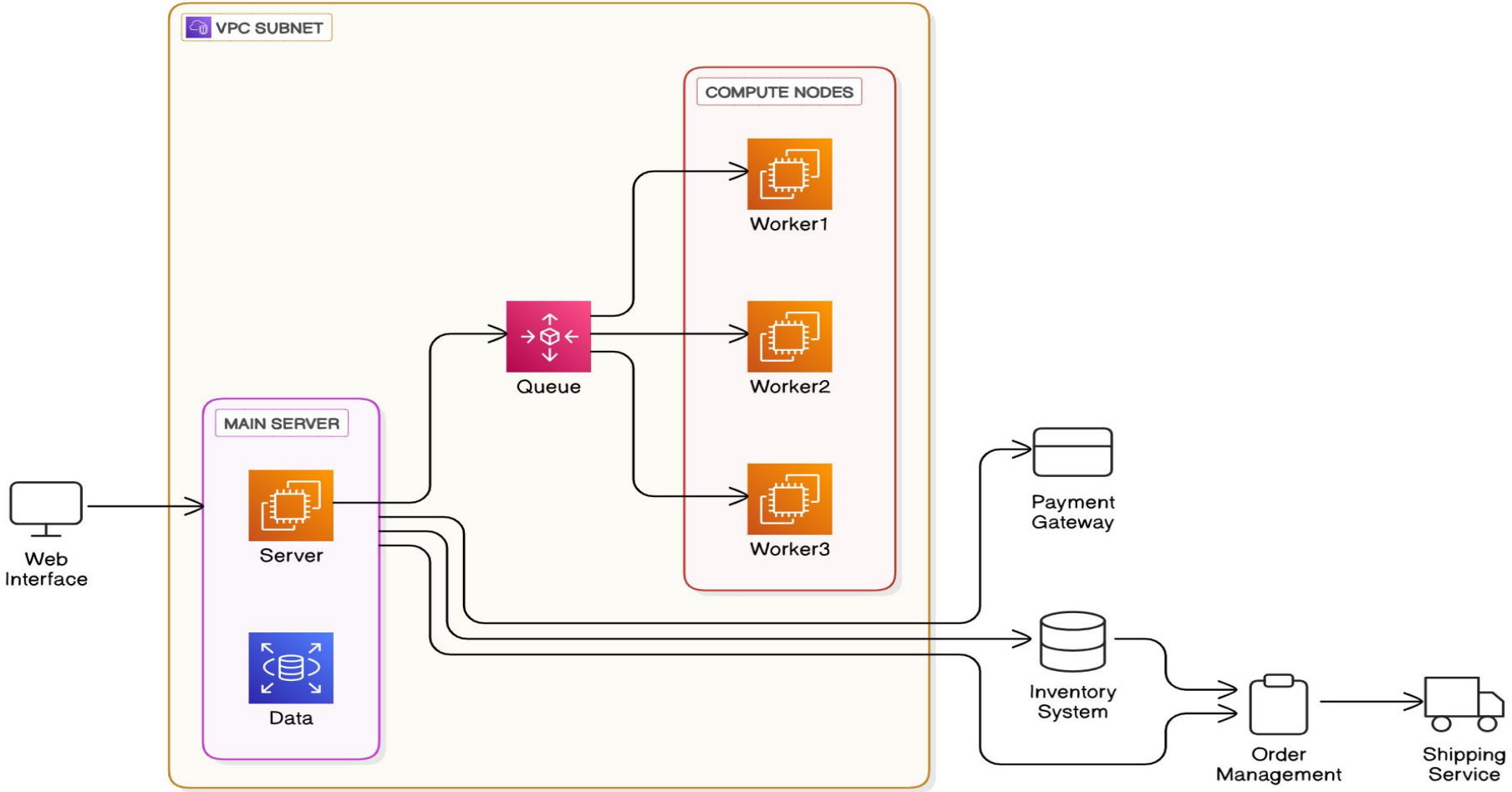
- **Auth0 or JWT:**
  - Secure login, registration, and session management for users and admins.
  - Role-based access control (e.g., admin vs. customer privileges).

Third-Party Services:

- **Email Notifications:**
  - Tools like SendGrid or Resend are used for sending order confirmations, shipping updates, and promotional emails.
- **Analytics:**
  - Google Analytics or custom solutions for tracking user behavior and optimizing the platform.



Workflow for Clothing E-commerce Website



Key Workflows

Workflow 1: User Browses Products

1. User visits the website.
2. Frontend fetches product data from the `/products` endpoint.
3. Sanity CMS provides product details (ID, name, price, stock, image).
4. Products are displayed in a responsive grid format.

Workflow 2: User Places an Order

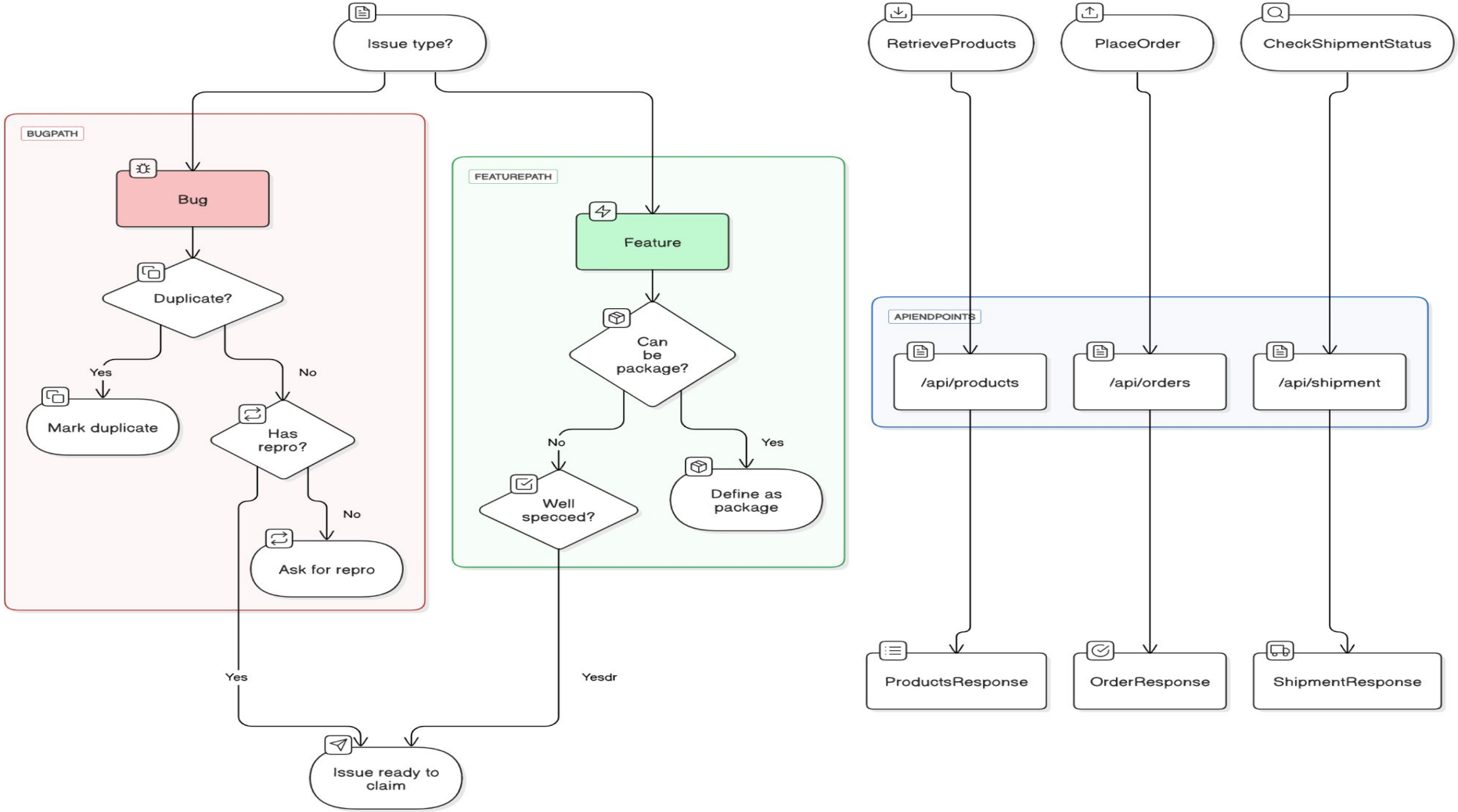
1. User selects products and adds them to the cart.
2. Frontend sends a POST request to `/orders` with:
    - Customer details
    - Product information
    - Payment status
3. Sanity CMS records the order.
4. Payment Gateway processes the transaction.
5. Confirmation email is sent to the user.

Workflow 3: User Tracks Shipment

1. User visits the “Track Order” page.
2. Frontend sends a GET request to `/shipment` with the order ID.
3. Third-party shipment API responds with:
    - Shipment ID
    - Status
    - Expected delivery date
4. Frontend displays the tracking information.

API Integration End Point Api

Endpoint	Method	Description	Payload/Query Parameters	Response Example
/api/products	GET	Retrieve a list of all available products.	None	[{ "id": "1", "name": "Long shirtr", "price": 5.99, "stock": 120, "image": "url-to-image" }, { "id": "2", "name": "Dark Truffle", "price": 9.99 }]
/api/orders	Post	Place a new order with customer and product details.	{ "customer": { "name": "Jane Doe", "email": "jane@example.com" }, "products": [{ "id": "1", "quantity": 2 }], "paymentStatus": "Paid" }	{ "orderId": "ORD123456", "status": "Confirmed" }
/api/shipment	GET	Retrieve the shipment status for a specific order.	?orderId=ORD123456	{ "shipmentId": "SHIP67890", "orderId": "ORD123456", "status": "Out for Delivery", "expectedDeliveryDate": "2025-01-20" }


Issue type?
BUGPATH
Bug
Duplicate?
Yes
No
Mark duplicate
Has repro?
No
Ask for repro
FEATUREPATH
Feature
Can be package?
No
Yes
Well specced?
Define as package
Yes
Yesdr
Issue ready to claim
RetrieveProducts
PlaceOrder
CheckShipmentStatus
APIENDPOINTS
/api/products
/api/orders
/api/shipment
ProductsResponse
OrderResponse
ShipmentResponse

Conclusion

The marketplace system is designed to offer a seamless shopping experience by combining modern tools like Next.js, Sanity CMS, and Stripe. With a robust architecture and clear workflows, this platform ensures scalability, responsiveness, and high performance. By following this plan, you can implement a professional eCommerce solution that caters to both business and user needs effectively.

THANK you

PREPARED BY
ALISHBA MAJEED