Backend Engineer Task Documentation

Overview

This document outlines the key tasks and responsibilities for backend engineers working on the API layer of our system. Each task has been broken down into specific services with detailed descriptions of their requirements, functionalities, and expectations.

1. Authentication Service

Description: The authentication service handles user login, registration, and session management. It also supports different user roles (vendor, buyer, admin).

Tasks:

- Implement user authentication using JWT or OAuth for secure login sessions.
- Design and implement role-based access control (RBAC) for vendors, buyers, and admins.
- Ensure session management with proper expiration and renewal mechanisms.
- Encrypt user credentials and store them securely.
- Implement multi-factor authentication (MFA) for added security.
- Create API endpoints for user registration, login, logout, and password recovery.
- Implement account verification via email or SMS.

2. Product Service

Description: This service manages product listings, categories, inventories, and prices. It handles CRUD operations for both vendors and buyers.

Tasks:

- Develop endpoints for creating, updating, retrieving, and deleting products.
- Implement inventory management, including stock level tracking and alerts.
- Design category and subcategory structures with efficient querying.
- Implement price tracking and discount management.
- Create a system to handle bulk product uploads.
- Implement image handling and storage for product images.

3. Order Service

Description: The order service processes orders, manages order statuses, and handles cancellations and returns.

Tasks:

- Develop order processing logic including creation, confirmation, and payment validation.
- Implement order status tracking (Pending, Confirmed, Shipped, Delivered, Canceled, Returned).
- Enable order cancellation and return processing.
- Implement an automated refund mechanism where necessary.
- Develop APIs to fetch order history for vendors and buyers.
- Implement real-time order updates using WebSockets or push notifications.

4. Payment Service

Description: This service integrates with external payment gateways (e.g., Paystack, Flutterwave) to handle customer payments and vendor payouts. It ensures multi-vendor payments are split accordingly.

Tasks:

- Integrate payment gateways for secure transactions.
- Develop logic for handling split payments between multiple vendors.
- Implement transaction history retrieval for users and vendors.
- Ensure secure payment data storage following compliance standards (e.g., PCI DSS).
- Implement automatic vendor payout mechanisms.
- Handle failed or disputed transactions.

5. Notification Service

Description: The notification service sends emails, SMS, or in-app notifications to users regarding order status, delivery updates, new product arrivals, etc.

Tasks:

- Implement an event-driven notification system.
- Set up email and SMS notifications using third-party providers (e.g., Twilio, SendGrid).
- Develop in-app notifications for real-time updates.
- Implement notification templates for different events.
- Ensure users can manage their notification preferences.

6. Vendor Management Service

Description: This service allows vendors to manage their stores, view sales reports, manage inventory, and process customer orders.

Tasks:

- Develop APIs for vendors to manage store details.
- Implement inventory management features for vendors.
- Provide APIs for retrieving sales reports and analytics.
- Allow vendors to manage orders and process returns.
- Implement store performance tracking.

7. Review and Rating Service

Description: This service enables buyers to leave reviews and ratings for products and vendors.

Tasks:

- Develop an API for posting, updating, and deleting reviews.
- Implement a rating system (e.g., 1-5 stars) for products and vendors.
- Develop moderation tools to flag inappropriate reviews.
- Implement review filtering and sorting.

8. Search and Recommendation Service

Description: This service provides search capabilities and personalized product recommendations for users.

Tasks:

- Implement a search engine with filters and sorting options.
- Develop a recommendation algorithm using machine learning or rule-based logic.
- Implement caching strategies to optimize search performance.
- Develop APIs for dynamic product recommendations based on user behavior.

9. Subscription Plans & Traffic Limitation

Description: The system enforces subscription-based traffic limitations for vendors based on their plan.

Tasks:

- Develop logic to track user sessions and interactions per vendor.
- Implement user limits based on the subscription plan (Basic, Premium, Diamond).

- Enforce logout when a vendor exceeds the allowed number of users in a day.
- Develop APIs for managing subscription plans and upgrades.
- Ensure vendors receive alerts before reaching user limits.

Subscription Plan Limitations:

Subscription Plan	Monthly User Limit	Yearly User Limit
Basic Monthly	100 Users	-
Basic Yearly	-	150 Users
Premium Monthly	300 Users	-
Premium Yearly	-	700 Users
Diamond Monthly	Unlimited	-
Diamond Yearly	-	Unlimited

10. Product Importation from Aliexpress

Description: Create a logic to import products from Aliexpress using third-party APIs.

Tasks:

- Research and integrate third-party APIs for importing Aliexpress products.
- Develop endpoints for fetching product details including images, prices, and descriptions.
- Implement logic for mapping Aliexpress product categories to local categories.
- Ensure data consistency and remove duplicate imports.
- Provide an interface for vendors to select and import products manually.

Conclusion

Backend engineers are expected to develop these services with scalability, security, and maintainability in mind. All APIs should follow RESTful principles or GraphQL where applicable. Logging, monitoring, and automated testing should be implemented across all services to ensure reliability and performance.