Technical Challenge for Mid-Level Software Engineer

Objective

Create a REST API that enables user registration, login, and a basic shop functionality. The API will manage user accounts, product purchasing, and basic administrative features. You will also document the API endpoints using Postman to ensure they are easy to test and understand. The API should include secure authentication and integrate with Stripe for payment handling. Deployment of the service on a free-tier hosting platform is required.

Requirements

1. Setup

- Use **GitHub** for version control. Share the repository link after the initial commit.
- Use **MySQL 8** as the database.
- Implement database migrations to populate the DB with a few sample products (design their properties as you see fit).
- Dockerize the service, ensuring that the entire application can run inside a container.
- Use **Postman** to create a collection of all endpoints, including examples for easy testing and usage.
- Deploy the API on a free-tier instance using platforms like AWS, Heroku, Railway.app, Vercel, or Fly.io to enable endpoint testing via Postman.

2. Authentication

- Implement secure user registration and login functionality using **JWT** tokens for authentication.
- Protect all endpoints except public ones, requiring users to log in to access features.

3. Purchasing

- Allow authenticated users to purchase products. Integrate Stripe's development environment to handle payment processing. (Stripe Quickstart Guide)
- Enable users to view their purchase history:
  - Detailed per-item statistics.
  - Overall statistics for all their purchases.

4. Admin Features

- Create an **admin dashboard** endpoint (restricted to admin users):
  - Display statistics about purchased items (e.g., sales per item, revenue generation, etc.).
  - Include features to update product pricing, descriptions, and other details based on insights from the statistics.
- Create a **public dashboard** endpoint (accessible without login):
  - Show basic product information and statistics (e.g., available products, top-selling items).
  - Provide a way to attract public users to register by offering insights into popular products or discounts.

Hints and Notes

GitHub Repository

- Approach this as a real-world task. Commit frequently, and document your progress clearly.
- Use a structured commit history and a README file to explain your approach.

Postman

- The Postman collection should include:
  - Example requests and responses for all endpoints.
  - Authentication examples showing how to pass JWT tokens.

Stripe Integration

- Use Stripe's test keys to simulate payment functionality. Ensure payments are tied to user accounts and purchase records.

Deliverables

1. A **working API** deployed to Heroku or AWS free tier, with all endpoints functional and accessible.
2. A **Postman collection** demonstrating all endpoints with example requests and responses.
3. A **GitHub repository** containing:
   - The full source code.
   - Docker configuration for containerizing the application.
   - Database schema and migration files.
   - README with instructions on how to set up, run, and test the application locally.

Timeline

The task is estimated to take **4–8 hours**. Focus on delivering a complete, functional solution that demonstrates your understanding of backend systems, database design, and RESTful principles.

Good luck, and happy coding!