Senior Laravel Developer Assignment

Project: E-Commerce Order Management System

Overview

Build a scalable REST API for an order management system with inventory tracking.

Core Features

1. Product & Inventory Management

- Product CRUD with variants
- Real-time inventory tracking
- Low stock alerts (queue job)
- Bulk product import via CSV
- Product search with Elasticsearch or full-text search

2. Order Processing

- Create orders with multiple items
- Order status workflow: Pending → Processing → Shipped → Delivered → Cancelled
- Inventory deduction on order confirmation
- Order rollback on cancellation (restore inventory)
- Invoice generation (PDF)
- Email notifications for order updates

3. Authentication & Authorization

- JWT authentication with refresh tokens
- Role-based access: Admin, Vendor, Customer
- Admin: Full access
- Vendor: Manage own products and orders
- Customer: Place orders, view order history

Technical Requirements

Laravel Architecture:

- Laravel 10+ with PHP 8.2+
- Service classes for business logic
- Repository pattern for data access
- Actions/Commands for complex operations
- Events & Listeners for decoupled logic
- Queue jobs for async operations (emails, PDFs, notifications)
- Database transactions for data integrity
- API versioning (v1)

Performance & Scalability:

- Query optimization (N+1 prevention, eager loading)
- Database indexing on searchable fields
- Implement database sharding strategy (document in README)
- Response pagination for large datasets

Testing (Critical):

- Feature tests for all API endpoints
- Unit tests for business logic

Deliverables

Submit via Google Form: https://forms.gle/JSqkuo4pJzu5wxXS6

GitHub Repository (Required):

1. Complete Source Code

- Clean, well-structured Laravel application
- Follow PSR-12 coding standards

2. Comprehensive Testing

- Feature tests (all endpoints)
- Unit tests (services, actions)

3. API Documentation

- OpenAPI/Swagger specification
- Postman collection with examples
- Authentication guide

5. README.md (Detailed)

Must include:

- Project overview and features
- Local setup instructions (step-by-step)
- Environment variables documentation
- API authentication guide
- Testing instructions
- Your name, email, GitHub profile

6. Additional Files

- .env.example with all variables
- database/seeders with realistic data

- API rate limiting configuration
- Queue configuration
- Caching strategy document

Exclude: vendor/, node_modules/, storage/logs/, .env

Timeline

Duration: 72 hours from receipt this mail

This assignment simulates real-world complexity. Show us how you handle it.

Good luck! 🚀