

GENERAL RULES (WAJIB BAGI SEMUA):

1. Modern PHP & Framework:

- Wajib menggunakan **Laravel (Versi 10 atau 11)**.
- **Strict Typing:** Wajib menyertakan `declare(strict_types=1);` di baris pertama setiap file PHP.
- **Type Hinting:** Wajib menggunakan *Type Hinting* pada setiap properti, parameter function, dan return types (misal: `public function store(Request $request): JsonResponse`).
- **Clean Code:**
 - Hindari "Fat Controllers". Logic bisnis yang kompleks harus dipisah ke dalam **Service Class, Action Class**, atau **Repository**.
 - Gunakan **Form Request** untuk validasi, jangan validasi manual di Controller.

2. No UI Libraries / Headless First:

- Fokus utama challenge ini adalah **API/Backend Logic** dan **Architecture**.
- Jika membutuhkan UI (misalnya untuk form upload di tugas Import, atau tombol beli di tugas Inventory), gunakan **Blade** sederhana dengan **Tailwind CSS**.
- Dilarang menggunakan *starter kit* UI yang berat (seperti Filament, Nova, atau Voyager) untuk fitur utama challenge.
- Response API wajib dalam format JSON yang standar dan konsisten.

3. Submission:

- Kirim Link Repository GitHub.
- **Commit History:** Wajib ada history commit yang jelas dan deskriptif. Jangan melakukan "One-Commit-Push".
- **Documentation:** Sertakan file `README.md` yang berisi:
 - Requirement server (versi PHP, ekstensi yang dibutuhkan).
 - Cara setup project (`composer install`, `cp .env`, `php artisan migrate`).
 - Cara menjalankan background worker (untuk tugas Queue).

TUGAS INDIVIDU (SPESIFIK):

Fatan Khayis Khabibullah

Focus: Performance & Async Process **Case:** "Bulk Data Import/Export Engine"

Challenge: Buat fitur untuk meng-upload file CSV berisi 100.000 - 1.000.000 baris data user (Nama, Email, Alamat).

Logic Requirements:

- **Memory Management (Streams):** Dilarang meload seluruh file ke dalam RAM (jangan pakai `fs.readFileSync` utuh). Gunakan **Node.js Streams** untuk membaca file baris per baris (*chunk*).
- **Background Job/Queue:** Request upload tidak boleh *blocking* (loading terus sampai selesai). Saat user upload, server langsung response "Processing".
- **Async Processing:** Proses validasi dan insert ke database harus berjalan di *background* (gunakan library antrian seperti BullMQ, Redis, atau Worker Threads).
- **Stability:** Sediakan endpoint untuk mengecek status progress import (misal: "Processed 45,000 of 100,000 rows"). Server tidak boleh *crash* (Out of Memory) meskipun file berukuran 500MB+.

DEADLINE

48 JAM dari pesan ini dikirim. Jika tidak dikumpulkan atau tidak sesuai Rules, otomatis gugur.

Goodluck!