**LMS Project Tasks — Monolithic Laravel + Blade + Vue + Tailwind**

**خطة عمل منظمة لكل branch مع مهام داخلية و commit messages**

**🌱 Branching Strategy**

* main → النسخة النهائية للتسليم.
* develop → الفرع الرئيسي للتطوير.
* feature/<name> → كل ميزة لها فرع خاص.
* hotfix/<name> → لإصلاح عاجل بعد التسليم.

**1️⃣ feature/setup-project**

**مهام:**

* إنشاء مشروع Laravel جديد.
* ضبط قاعدة البيانات في .env.
* تثبيت Laravel Breeze مع Blade + Tailwind + Vite.
* إضافة Vue 3 لدعم الـ components.
* تشغيل migration الأساسي.
* إنشاء symlink للمجلد storage.

**Commits:**

* chore: install Laravel 12 and setup base project
* chore: configure .env for local environment
* chore: install Laravel Breeze (blade) and tailwind/vite setup

**2️⃣ feature/auth-roles**

**مهام:**

* إضافة عمود role في جدول users (admin, instructor, student).
* إنشاء seeders للمستخدمين الأساسيين.
* إعداد Gates / Policies للتحكم في الصلاحيات.
* اختبار تسجيل دخول وصلاحيات لكل role.

**Commits:**

* feat: add role field to users table
* feat: implement role-based authorization with Gates & Policies
* test: add seeders for admin, instructor, and student users

**✅ المرحلة 3: تصميم قاعدة البيانات + الموديلات**

**branch:** feature/database-schema  
**مهام:**

* إنشاء الموديلات + الميجريشنات:
  + Course (belongsTo Instructor, hasMany Lessons).
  + Lesson (belongsTo Course).
  + Enrollment (pivot Student ↔ Course).
  + Assignment (belongsTo Lesson, created by Instructor).
  + Submission (belongsTo Student & Assignment).
* ضبط العلاقات في Models.
* إنشاء Seeders + Factories لبيانات تجريبية.

**Commits:**

* feat: add courses, lessons, assignments, submissions migrations and models
* feat: create enrollments pivot table
* chore: add factories and seeders for testing data

**✅ المرحلة 4: إدارة الكورسات (Admin)**

**branch:** feature/courses-crud  
**مهام:**

* إنشاء CourseController (resource).
* إعداد Routes + Gates للتحكم بالصلاحيات.
* إنشاء Blade views (index, create, edit, show).
* Validation + Eager Loading.
* تحسين UX (modals + Vue component للبحث/تصفية).

**Commits:**

* feat: add course CRUD controllers and routes
* feat: implement Blade views for managing courses
* fix: improve validation rules for course creation

**✅ المرحلة 5: إدارة الدروس (Admin)**

**branch:** feature/lessons-crud  
**مهام:**

* إنشاء LessonController.
* Routes nested داخل الكورسات (courses/{course}/lessons).
* رفع ملفات الموارد (PDF, Video, Docs).
* ربط الدروس بالكورسات.
* Blade views لإدارة الدروس.

**Commits:**

* feat: add lesson CRUD controllers and routes
* feat: implement file upload for lessons
* feat: add Blade UI for lessons management

**✅ المرحلة 6: تسجيل الطلاب في الكورسات (Enrollments)**

**branch:** feature/enrollments  
**مهام:**

* تسجيل الطلاب في الكورسات (attach/detach).
* عرض الكورسات الخاصة بالطالب.
* حماية المسارات بـ Gates.
* إنشاء Blade UI للتسجيل وإلغاء التسجيل.

**Commits:**

* feat: implement student enrollment system
* feat: add Blade UI for course enrollment management

**✅ المرحلة 7: إدارة الواجبات (Instructor)**

**branch:** feature/assignments  
**مهام:**

* المدرس ينشئ واجبات مرتبطة بدروس الكورس.
* الطالب يرفع الحل (نص/ملف).
* المدرس يصحح ويضع الدرجة + الملاحظات.
* عرض حالة الواجب للطالب (Submitted / Graded).

**Commits:**

* feat: add assignment creation and submission system
* feat: implement grading workflow for instructors
* feat: display assignment status for students

**✅ المرحلة 8: الإشعارات (Notifications)**

**branch:** feature/notifications  
**مهام:**

* إعداد Notifications (database + mail).
* ربط MailHog للتجربة.
* عرض الإشعارات داخل Blade + Vue toast/flash.
* إرسال تنبيهات عند:
  + رفع واجب.
  + تصحيح واجب.

**Commits:**

* feat: implement database and mail notifications
* feat: add MailHog integration for local testing
* feat: integrate Vue toast notifications

**✅ المرحلة 9: لوحة التحكم (Dashboards)**

**branch:** feature/dashboard  
**مهام:**

* **لوحة Admin**: إحصائيات (عدد الطلاب، الكورسات، الواجبات).
* **لوحة Instructor**: واجبات للتصحيح، نشاط الطلاب.
* **لوحة Student**: الكورسات المسجل بها + حالة الواجبات.
* استخدام Chart.js أو Vue Chart.

**Commits:**

* feat: add admin dashboard with statistics
* feat: implement instructor dashboard
* feat: add student dashboard UI

**✅ المرحلة 10: تحسين الواجهات (UI Polish)**

**branch:** feature/ui-polish  
**مهام:**

* Dark/Light Mode toggle.
* تحسين الـ responsiveness بـ Tailwind.
* توحيد رسائل النجاح/الخطأ.
* إعادة استخدام Blade Components.

**Commits:**

* style: add dark/light mode toggle
* style: improve responsive design
* refactor: reuse Blade components for UI consistency

**✅المرحلة 11: إدارة الأخطاء وتوثيق الأحداث (Error Handling & Logging)  
branch:** feature/error-logging **مهام:**

* إعداد Laravel Logging وتهيئة قنوات مختلفة (single, daily, stack).
* توثيق الأحداث المهمة عبر Event Logging (DB, log files).
* تسجيل الأخطاء في exception log
* عمل شاشه error 500 لعرض الأخطاء

**Commits :**

* feat: add custom exception handler and error pages
* feat: configure log channels
* feat: implement event logging for key actions
* chore: add logging for critical system events

**✅ المرحلة 12: التوثيق والتسليم**

**branch:** feature/docs  
**مهام:**

* كتابة README (شرح التنصيب والاستخدام).
* إضافة Screenshots للشاشات.
* توثيق API (إن وجدت).
* شرح Git branching + commit messages.

**Commits:**

* docs: add project README with setup instructions
* docs: add screenshots for demo
* docs: add API documentation

**📌 Workflow ملخص**

* Branches → develop → feature/\* → test → merge.
* Commits → صغيرة، واضحة، descriptive.
* Testing → يدوي + seeders.
* UI → Blade + Vue + Tailwind (responsive).
* آخر دمج → تسليم نسخة نظيفة من main.