

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

خطة عمل منظمة لكل branch مع مهام داخلية و commit messages

Branching Strategy

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

1 feature/setup-project

مهام:

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

Commits:

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

2 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 •
-

feature/database-schema

مهام:

- إنشاء الموديلات + المجرشنس: •
- Course (belongsTo Instructor, hasMany Lessons) ○
 - Lesson (belongsTo Course) ○
 - Enrollment (pivot Student ↔ Course) ○
- Assignment (belongsTo Student & Course, file upload) ○
 - إضافة Seeders و Factories لتجربة البيانات.
 - ضبط العلاقات في Models. •

Commits:

- feat: create courses and lessons migrations + models •
 - feat: create enrollments pivot table •
 - feat: create assignments table with file upload support •
 - chore: add factories and seeders for testing data •
-

feature/courses-crud

مهام:

- إنشاء CourseController (resource). •
- إعداد CRUD Routes •
- إنشاء Blade views (index, create, edit, show). •
- إضافة eager loading و validation. •
- تحسين UX بإضافة Vue component للمodal. •

Commits:

- feat: add course CRUD controllers and routes •
- feat: implement Blade views for managing courses •

fix: improve validation rules for course creation •

5 feature/lessons-crud

مهام:

- إنشاء LessonController (resource). •
- إعداد Routes nested (courses.lessons). •
- رفع ملفات الموارد (PDF, video). •
- ربط الدروس بالكورسات. •
- إنشاء 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 •
-

6 feature/enrollments

مهام:

- تسجيل الطلاب في الدورات. (attach/detach) •
- عرض كورسات الطالب في لوحة الطالب. •
- حماية المسارات عبر middleware/Gates. •
- إنشاء UI للتسجيل والغاء التسجيل. •

Commits:

- feat: implement student enrollments system •
 - feat: add Blade UI for student course enrollments •
-

7 feature/assignments

مهام:

- رفع الواجبات (file upload) من قبل الطالب. •
- التقييم من قبل المدرس وتحديث grade. •
- إرسال إشعارات عند رفع أو تقييم الواجب. •

- عرض حالة الواجبات في لوحة الطالب.

Commits:

- feat: add assignment submission with file upload •
 - feat: implement grading system for assignments •
 - feat: trigger notifications on assignment submission and grading •
-

8 feature/notifications

مهام:

- إعداد Notifications (database + mail). •
- اختبار MailHog لإرسال رسائل البريد محلياً. •
- عرض الإشعارات في Blade. •
- استخدام Vue Toast أو flash messages لتجربة تفاعلية. •

Commits:

- feat: implement database notifications •
 - feat: add mail notifications with MailHog •
 - feat: integrate Vue toast notifications for frontend feedback •
-

9 feature/dashboard

مهام:

- إنشاء AdminController + Route للوحة التحكم. •
- إحصاءات: عدد الطلاب، عدد الدورات، الواجبات. •
- تمرير البيانات إلى Blade. •
- رسم رسوم بيانية (Chart.js) أو Vue Chart component. •

Commits:

- feat: implement admin dashboard with charts •
 - feat: add course and student statistics •
 - style: enhance dashboard UI with Tailwind •
-

10 feature/ui-polish

مهام:

- إضافة Dark/Light Mode toggle.
- تحسين Tailwind responsiveness باستخدام partial Blade.
- توحيد رسائل الخطأ والنجاح في Blade components لتقليل تكرار الكود.
- إعادة استخدام Blade components لتنقلي تكرار الكود.

Commits:

- style: implement dark/light mode toggle
- style: improve responsive design across pages
- refactor: clean up Blade components and UI partials

1 1 feature/docs

مهام:

- كتابة README مع شرح التنصيب والاستخدام.
- إضافة صور للشاشات.(screenshots).
- توثيق أي API صغيرة charts مثلاً.
- شرح Git branching و commit messages.

Commits:

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

نصائح للمبتدئ

1. التزم بتنفيذ كل branch بالترتيب المذكور.
2. جرب كل خطوة قبل الانتقال للخطوة التالية.
3. استعمل factories و seeders لتعبئة بيانات اختبارية بسرعة.
4. استخدم Blade components لتقليل التكرار.
5. التزم برسائل commit منظمة وواضحة.
6. احفظ نسخة من المشروع قبل كل دمج في develop/main.

Workflow مختصر

- Branches → develop → feature → test → merge
- Commits → descriptive, small, and organized
- Testing → manual + seeder data
- UI → Blade + Vue + Tailwind + responsive