

## Objectif global (2 semaines)

Créer une **application de gestion des tâches multi-utilisateur** avec :

- Authentification (Admin / Employé),
  - Création et affectation de tâches,
  - Envoi d'e-mails et notifications internes,
  - Tableau de bord simplifié,
  - Planificateur automatique (rappels journaliers).
- 

### ♦ Semaine 1 – Construction du cœur du système

#### Jour 1 : Mise en place

- Installation Laravel, configuration DB, routes, Git.
- Création du modèle **Task** (titre, description, statut, date d'échéance, user\_id, assigned\_by).
- CRUD complet des tâches.
- Introduction à MVC + Blade.

#### Jour 2 : Authentification & rôles

- Installation de Laravel Breeze.
- Ajout des rôles : Admin (manager) / User (employé).
- Middleware pour restreindre les pages.
- L'Admin peut créer des utilisateurs.

#### Jour 3 : Affectation & relations

- Relation **User hasMany Tasks, Task belongsTo User**.

- Formulaire d'ajout avec choix du collaborateur à qui la tâche est assignée.
- Filtrage : "Mes tâches", "Tâches assignées par moi".

#### **Jour 4 : Notifications & mails**

- Notification lors d'une nouvelle tâche.
- Envoi d'un mail automatique à l'utilisateur assigné.
- Affichage des notifications dans l'interface.

#### **Jour 5 : Interface & fin module 1**

- Amélioration du design (Tailwind ou Bootstrap).
- Table dynamique avec statut coloré (À faire / En cours / Terminé).
- Test global + mise en GitHub.

---

### **♦ Semaine 2 – Automatisation & perfectionnement**

#### **Jour 6–7 : Planificateur & rappels**

- Implémentation de `schedule:run`.
- Envoi d'un mail chaque matin listant les tâches du jour.
- Changement automatique du statut à "Retardée" si la date est dépassée.

#### **Jour 8 : Tableau de bord**

- Compteurs rapides : total tâches, en retard, terminées, assignées.
- Graphique simple (tâches par statut ou par utilisateur).
- Filtres dynamiques (date, priorité, statut).

#### **Jour 9 : Commentaires et pièces jointes (optionnel si temps)**

- Table `comments` liée à `tasks`.
- Upload de fichier simple (rapport, image, doc).

## **Jour 10 : Tests finaux & déploiement**

- Tests CRUD et mails.
- Mise en ligne sur un hébergeur test (ou local + GitHub).
- Documentation courte du projet.