



Planning détaillé des tâches — Lectorium Rosicrucianum (Web)

Semaine 1 — Authentification & Modélisation DB

Objectif : Préparer la base pour les utilisateurs et l'authentification

Frontend (Samuel) :

- Créer les formulaires d'inscription et de connexion (email, mot de passe, matricule, niveau/grade)
- Ajouter la validation côté client (format email, mot de passe fort, champs obligatoires)
- Gestion des messages d'erreur et des notifications (toast / alert)
- Intégrer la logique frontend pour la connexion et l'inscription via API

Backend (Haïdarath) :

- Créer la base de données MySQL avec tables : users, roles, centres, activities, inscriptions, payments
- Développer les endpoints API pour CRUD utilisateurs (register, login, logout)
- Implémenter JWT / Sanctum pour authentification sécurisée
- Tester les endpoints avec Postman / Insomnia

Livrables :

- Authentification opérationnelle (frontend <-> backend)
- Schéma de base de données fonctionnel

Semaine 2 — Pages publiques

Objectif : Créer les pages accessibles aux visiteurs

Frontend (Samuel) :

- Page d'accueil avec présentation de l'association, hero, sections travaux & podcasts
- Liste des activités publiques avec filtres (centre, date, type)
- Footer et header responsive
- SEO de base : meta tags, titres de page, Open Graph

- Mise en place de la navigation entre les pages publiques

Backend (Haïdarath) :

- Endpoints API pour récupérer les activités, podcasts et actualités (GET /activities, GET /podcasts, GET /news)
- Gestion des filtres côté serveur
- Tests endpoints pour données publiques

Livrables :

- Pages publiques fonctionnelles et responsive
 - API publique opérationnelle
-

Semaine 3 — Espace membre

Objectif : Développer le tableau de bord membre et inscriptions

Frontend (Samuel) :

- Dashboard membre : activités inscrites, statut de paiement
- Formulaire d'inscription à une activité
- Intégration des notifications : confirmation inscription, rappel activité
- Historique des inscriptions et paiements
- Interface responsive et utilisable sur desktop/tablette

Backend (Haïdarath) :

- Endpoints API : GET /user/activities, POST /inscriptions, GET /payments
- Vérification et validation des inscriptions
- Gestion du statut de paiement
- Tests endpoints avec différents cas (inscription complète, paiement manquant)

Livrables :

- Dashboard membre complet et connecté à l'API
 - Inscription et suivi paiement fonctionnels
-

Semaine 4 — Espace administrateur

Objectif : Dashboard admin pour gestion complète

Frontend (Samuel) :

- Layout du dashboard admin
- Formulaire pour création/modification/clôture activités
- Interface pour gérer inscriptions et visualiser les paiements
- Export CSV/Excel des listes d'inscriptions
- Gestion des rôles : ajout/suppression d'admins
- Responsive design et UX optimisée pour admin

Backend (Haïdarath) :

- Endpoints CRUD activités (POST /activities, PUT /activities/{id}, DELETE /activities/{id})
- Endpoint gestion inscriptions et export CSV/Excel (GET /inscriptions/export)
- Endpoint gestion rôles (POST /roles, DELETE /roles/{id})
- Tests endpoints et sécurisation des routes admin

Livrables :

- Dashboard admin opérationnel
- Export des données et gestion rôles fonctionnels

Semaine 5 — Intégration paiements**Objectif : Intégrer Flutterwave, FedaPay, CinetPay****Frontend (Samuel) :**

- Création des interfaces de paiement (sélection du service, montant, confirmation)
- Intégration des API de paiement pour déclencher la transaction
- Gestion des notifications de succès/échec du paiement
- Mise à jour du statut de paiement sur le dashboard membre

Backend (Haïdarath) :

- Endpoints paiement sécurisés (POST /payments)
- Vérification côté serveur des transactions et mise à jour des inscriptions
- Tests complets des flux de paiement avec les trois services

Livrables :

- Paiement en ligne opérationnel
 - Statut de paiement mis à jour en temps réel
-

Semaine 6 — Tests et optimisation

Objectif : Garantir la stabilité et la qualité de la plateforme

Frontend (Samuel) :

- Tests fonctionnels sur toutes les pages et dashboards
- Tests responsive et correction bugs UI
- Optimisation performance (lazy loading, images, composants)

Backend (Haïdarath) :

- Tests unitaires et tests fonctionnels Laravel
- Optimisation requêtes SQL et endpoints API
- Sécurisation des endpoints sensibles

Livrables :

- Plateforme stable, tests réussis
 - Performance optimisée
-

Semaine 7 — Déploiement et documentation

Objectif : Livraison finale et documentation

Frontend (Samuel) :

- Déploiement sur Vercel (préprod et prod)
- Vérification des liens API et affichage final
- Documentation utilisateur : guide d'utilisation membres et admins

Backend (Haïdarath) :

- Déploiement backend sur serveur PHP/MySQL
- Configuration domaine et SSL
- Rédaction documentation technique API

Livrables :

- Plateforme web déployée et fonctionnelle

- Documentation complète (technique + utilisateur)

Si tu veux, je peux maintenant **faire un tableau visuel type Gantt**, qui montre clairement **Frontend vs Backend**, les tâches par semaine et les livrables, pour avoir un document prêt à présenter à l'équipe ou aux admins.

Veux-tu que je fasse ça ?

You said:

Ornella et Haidarath s'occuperont du backend

ChatGPT said:

Parfait ! Alors, on met à jour le planning en considérant que **tu t'occupes seul du frontend**, et que **Ornella et Haïdarath gèrent ensemble le backend**. Voici le planning détaillé par semaine, avec toutes les tâches frontend et backend clairement séparées :

Planning détaillé par semaine — Lectorium Rosicrucianum (Web)

Semaine 1 — Authentification & Modélisation DB

Objectif : Préparer la base pour les utilisateurs et l'authentification

Frontend (Samuel) :

- Créer formulaires d'inscription et connexion (email, mot de passe, matricule, niveau/grade)
- Ajouter validation côté client (email valide, mot de passe fort, champs obligatoires)
- Gestion des messages d'erreur et notifications (toast / alert)
- Intégration de la connexion et inscription via API

Backend (Ornella & Haïdarath) :

- Créer la base de données MySQL : tables users, roles, centres, activities, inscriptions, payments
- Développer endpoints API pour CRUD utilisateurs (register/login/logout)
- Implémenter JWT / Sanctum pour authentification sécurisée
- Tests endpoints avec Postman/Insomnia

Livrables :

- Authentification opérationnelle
 - Schéma DB fonctionnel
-

Semaine 2 — Pages publiques

Objectif : Créer les pages accessibles aux visiteurs

Frontend (Samuel) :

- Page d'accueil : présentation de l'association, hero, sections travaux & podcasts
- Liste activités publiques avec filtres (centre, date, type)
- Footer et header responsive
- SEO de base : meta tags, titres, Open Graph
- Navigation entre pages publiques

Backend (Ornella & Haïdarath) :

- Endpoints API pour récupérer activités, podcasts et actualités (GET /activities, GET /podcasts, GET /news)
- Gestion des filtres côté serveur
- Tests endpoints données publiques

Livrables :

- Pages publiques fonctionnelles et responsive
 - API publique opérationnelle
-

Semaine 3 — Espace membre

Objectif : Développer le dashboard membre et inscriptions

Frontend (Samuel) :

- Dashboard membre : activités inscrites, statut de paiement
- Formulaire d'inscription à une activité
- Intégration notifications (confirmation inscription, rappel activité)
- Historique des inscriptions et paiements
- Interface responsive desktop & tablette

Backend (Ornella & Haïdarath) :

- Endpoints API : GET /user/activities, POST /inscriptions, GET /payments
- Vérification et validation inscriptions
- Gestion du statut de paiement
- Tests endpoints avec différents cas

Livrables :

- Dashboard membre complet et connecté à l'API
 - Inscription et suivi paiement fonctionnels
-

Semaine 4 — Espace administrateur

Objectif : Dashboard admin pour gestion complète

Frontend (Samuel) :

- Layout dashboard admin
- Formulaire création/modification/clôture activités
- Interface pour gérer inscriptions et visualiser paiements
- Export CSV/Excel des listes d'inscriptions
- Gestion rôles : ajout/suppression admins
- Responsive design & UX optimisée

Backend (Ornella & Haïdarath) :

- Endpoints CRUD activités (POST /activities, PUT /activities/{id}, DELETE /activities/{id})
- Gestion inscriptions et export CSV/Excel (GET /inscriptions/export)
- Gestion rôles (POST /roles, DELETE /roles/{id})
- Sécurisation des routes admin et tests

Livrables :

- Dashboard admin opérationnel
 - Export des données et gestion rôles fonctionnels
-

Semaine 5 — Intégration paiements

Objectif : Intégrer Flutterwave, FedaPay, CinetPay

Frontend (Samuel) :

- Interfaces de paiement (choix du service, montant, confirmation)
- Intégration des API de paiement pour déclencher transaction
- Gestion notifications succès/échec
- Mise à jour du statut paiement sur dashboard membre

Backend (Ornella & Haïdarath) :

- Endpoints paiement sécurisés (POST /payments)
- Vérification côté serveur des transactions et mise à jour inscriptions
- Tests complets flux de paiement avec les trois services

Livrables :

- Paiement en ligne opérationnel
 - Statut paiement mis à jour en temps réel
-

Semaine 6 — Tests et optimisation**Objectif : Garantir stabilité et qualité de la plateforme****Frontend (Samuel) :**

- Tests fonctionnels sur toutes les pages et dashboards
- Tests responsive et correction bugs UI
- Optimisation performance (lazy loading, images, composants)

Backend (Ornella & Haïdarath) :

- Tests unitaires et fonctionnels Laravel
- Optimisation requêtes SQL et endpoints API
- Sécurisation endpoints sensibles

Livrables :

- Plateforme stable, tests réussis
 - Performance optimisée
-

Semaine 7 — Déploiement et documentation**Objectif : Livraison finale et documentation**

Frontend (Samuel) :

- Déploiement Next.js sur Vercel
- Vérification liens API et affichage final
- Documentation utilisateur (guide membres et admins)

Backend (Ornella & Haïdarath) :

- Déploiement Laravel sur serveur PHP/MySQL
- Configuration domaine et SSL
- Rédaction documentation technique API

Livrables :

- Plateforme web déployée et fonctionnelle
- Documentation complète (technique + utilisateur)