

Petit Rapport : Projet de Gestion de Séminaires

IMSP

Le développement web est une branche de l'informatique permettant de mettre en place des sites internet afin de connecter le monde. C'est dans cette idée que le site de gestion de séminaire de l'IMSP a été demandé d'être mis en place pour faciliter les échanges entre les utilisateurs de ladite entité. Dans la réalisation de ce projet, nous avons eu à suivre certaines méthodologies pour nous simplifier la tâche. Dans les lignes suivantes, nous présenterons nos diagrammes de cas d'utilisation et de classe après avoir présenté le projet en général.

1. Objectif du Projet

Le projet vise à développer une plateforme web pour la gestion des séminaires au sein des entités de recherche de l'IMSP. L'objectif principal est d'améliorer l'organisation, la programmation, la diffusion des informations relatives aux séminaires et l'accès aux supports présentés. La plateforme permet la gestion des demandes de présentation, la validation par un secrétariat, la publication des programmes, et l'accès aux documents pour les participants.

2. Rôles des Utilisateurs

La plateforme gère trois rôles principaux :

- Secrétaire : Responsable de la gestion administrative des séminaires. Valide ou refuse les demandes, fixe les dates, publie les informations, envoie les notifications, et met à disposition les fichiers finaux.
- Présentateur : Soumet une demande, suit le statut, envoie un résumé, et consulte les séminaires disponibles.
- Étudiant : Consulte les séminaires publiés et télécharge les documents.

3. Fonctionnalités Principales Mises en Place ou Discutées en Détail

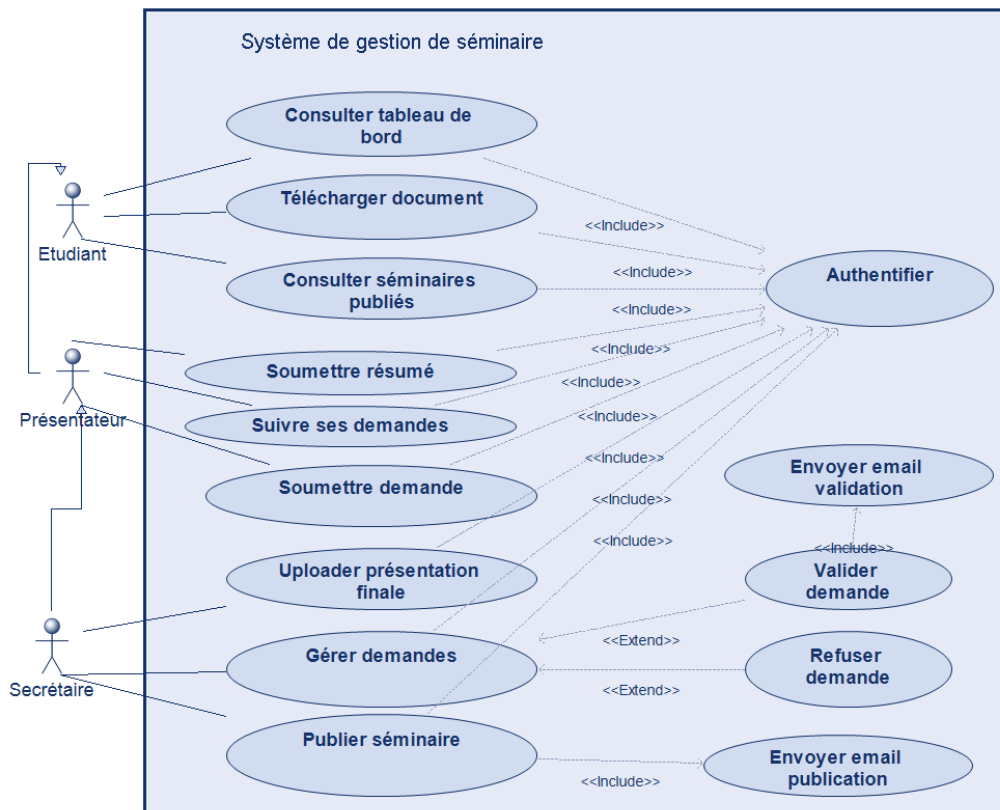
- Authentification complète via Laravel Breeze.
- Gestion des rôles avec spatie/laravel-permission.
- Tableaux de bord spécifiques selon le rôle.

- Soumission, validation, refus de demande de séminaire.
- Publication de séminaires avec notifications par e-mail.
- Upload et téléchargement de documents (résumés, présentations finales).

4. Environnement Technique

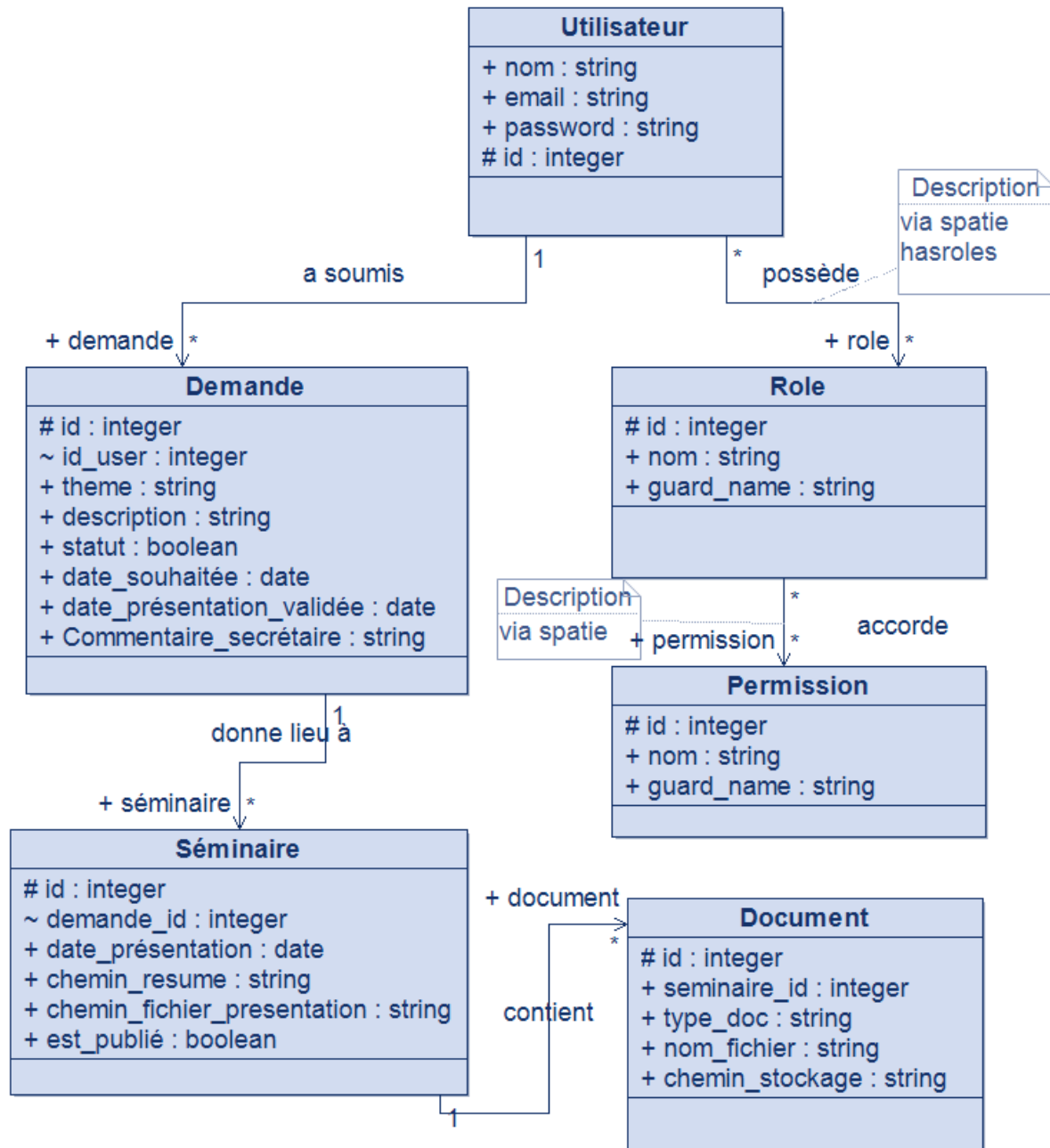
- Backend : Laravel (PHP)
- Frontend: Blade, Tailwind CSS, Alpine.js
- Base de Données : MySQL
- Serveur Web : Nginx
- Développement : Docker (PHP, Nginx, MySQL, MailHog)
- Rôles : spatie/laravel-permission
- Notifications : Laravel Notifications (MailHog)

5. Diagramme des Cas d'Utilisation



- Acteurs : Étudiant, Présentateur, Secrétaire, Système
- Cas : Authentification, Tableau de bord, Consultation, Téléchargement, Soumission, Validation, Refus, Publication.

6. Diagramme de Classes



- User, Demande, Séminaire, Document, Role, Permission, Notifications.

7. Étapes de Développement (Résumé)

◆ Environnement Docker

Nous avons commencé par la mise en place d'un environnement de développement conteneurisé avec **Docker**. L'objectif était d'assurer un déploiement cohérent et reproductible sur différentes machines. Pour cela, nous avons utilisé un fichier docker-compose.yml afin de lancer plusieurs services :

- **PHP** (avec Laravel)
- **Nginx** pour le serveur HTTP
- **MariaDB** comme base de données
- **phpMyAdmin** pour la gestion de la base via une interface
- **MailHog** pour le test d'envoi d'emails localement

Cela nous a permis de développer dans un environnement proche de la production tout en simplifiant l'installation des dépendances.

◆ Configuration de Laravel Breeze

Une fois l'environnement Docker opérationnel, nous avons installé **Laravel Breeze**, un kit de démarrage léger pour l'authentification. Breeze fournit :

- Le système d'enregistrement et de connexion
- Des routes sécurisées
- Des vues Blade avec Tailwind CSS
- Un système de session et de redirection après connexion

Cette base nous a permis d'intégrer rapidement les fonctionnalités de base de gestion des utilisateurs tout en gardant un code lisible et modulaire.

◆ Migrations et Seeders

Pour structurer la base de données, nous avons utilisé les **migrations Laravel**, permettant de versionner et modifier facilement le schéma de la base. Les tables principales créées incluent :

- users
- demandes
- séminaires
- documents
- rôles
- permissions

Les **seeders** ont été utilisés pour préremplir la base avec des rôles (secrétaire, étudiant, etc.) et des utilisateurs de test. Cela facilite le développement en disposant immédiatement d'un jeu de données représentatif.

◆ Intégration de Spatie Laravel Permission

Pour la gestion des rôles et permissions, nous avons intégré le **package Spatie Laravel Permission**. Il nous permet :

- D'assigner plusieurs rôles à un utilisateur
- De vérifier les permissions dans les contrôleurs et les vues
- De restreindre l'accès à certaines routes via des middleware (role, permission)

Exemple : une secrétaire peut accéder à la liste des demandes à valider, alors qu'un étudiant ne peut voir que ses propres demandes.

◆ Styling avec Tailwind CSS

Le style de l'interface a été réalisé avec **Tailwind CSS**, qui est intégré nativement avec Breeze. Grâce à ce framework utilitaire :

- Le design est resté simple, responsive et cohérent
- Les composants Blade sont faciles à styliser
- Nous avons utilisé des classes pour la mise en page, les couleurs, les boutons, les cartes, etc.

Ce choix a permis un développement rapide sans se perdre dans des fichiers CSS complexes.

◆ Implémentation des rôles

Une partie importante a été l'implémentation des **rôles dans le code** :

- Création d'un middleware pour filtrer l'accès selon le rôle
- Ajout de conditions dans les vues pour afficher certains boutons ou sections uniquement aux utilisateurs autorisés
- Création d'un tableau de bord différencié : par exemple, la secrétaire voit toutes les demandes, les étudiants voient uniquement les leurs

Les rôles ont été gérés dynamiquement grâce aux méthodes fournies par Spatie (assignRole(), hasRole(...)).

◆ Débogage et tests

Tout au long du projet, nous avons effectué :

- Du **débogage** via les messages d'erreur Laravel, la commande `php artisan serve`, les logs Docker, et les outils du navigateur
- Des **tests manuels** : enregistrement d'utilisateurs, soumission de demandes, envoi de documents, publication des séminaires...
- La vérification du comportement selon les rôles (tests d'accès et d'autorisation)

Nous avons également utilisé des **assertions simples** dans des tests automatisés pour s'assurer du bon comportement des routes et des contrôleurs.

8. Prochaines Étapes Suggérées

◆ Finaliser la gestion des documents

La plateforme permet aux utilisateurs (étudiants ou présentateurs) de **déposer des documents** liés aux séminaires (résumé, présentation, annexes, etc.). Nous avons :

- Créé un modèle Document lié au modèle Seminaire
- Ajouté une interface pour l'**upload de fichiers** avec vérification du type et de la taille
- Défini une nomenclature de stockage sur le serveur (storage/app/public/...)
- Permis à la secrétaire ou à l'administrateur de **télécharger les documents** depuis l'interface

La finalisation a consisté à tester tous les cas (dépôt, suppression, renommage automatique, affichage conditionnel).

◆ Affiner la logique temporelle

La logique temporelle concernait :

- La **date souhaitée de présentation** soumise par l'étudiant
- La **date validée par la secrétaire**
- La cohérence entre les deux (ex. : validation ne peut être antérieure à la demande)
- Les séminaires passés et futurs doivent être différenciés dans l'affichage

Nous avons utilisé des règles de validation dans les formulaires Laravel, et géré les cas particuliers (modification de date, filtrage par date, tri chronologique dans les listes).

◆ Compléter les tests

Pour assurer la robustesse de la plateforme, des **tests ont été complétés** :

- Tests unitaires des modèles (Demande, Seminaire, Document)
- Tests d'intégration sur le cycle complet : connexion → soumission → validation → téléchargement
- Tests de rôles : accès bloqué pour utilisateurs non autorisés
- Tests manuels d'interface : envoi de fichiers, redirections, alertes de succès/erreur

Ces tests garantissent que la logique métier et les accès sont respectés, et facilitent la maintenance.

◆ Rédiger une documentation utilisateur

Une **documentation utilisateur** a été rédigée pour faciliter la prise en main de la plateforme :

- Présentation du fonctionnement général
- Procédure de création de compte
- Soumission d'une demande de séminaire
- Dépôt et téléchargement des fichiers
- Validation d'une demande (interface secrétaire)

- Rappel des rôles et des restrictions

Cette documentation peut être fournie au format PDF ou intégrée dans l'interface (FAQ ou page d'aide).

◆ Préparer le déploiement

Enfin, nous avons préparé le **déploiement de la plateforme** :

- Optimisation de la configuration Docker pour un environnement de production
- Ajout de variables d'environnement (.env) pour la base de données, les emails, les clés de chiffrement
- Vérification des permissions sur les dossiers (storage, bootstrap/cache)
- Création d'un script d'installation automatique (migration + seed + liens symboliques)
- Éventuellement configuration d'un serveur distant (ex. : VPS avec Docker, ou Heroku/Render)

Ces étapes permettent d'assurer que la plateforme peut être utilisée par les étudiants et la secrétaire sans difficulté technique.

9-Schéma Relationnel de la Base de Données (Description)

Voici les tables principales et leurs relations :

1. Table users

Cette table stocke les informations sur tous les utilisateurs de l'application (Étudiants, Présentateurs, Secrétaires).

- **Colonnes principales :**
 - id (Clé Primaire, BIGINT UNSIGNED, Auto-incrément)
 - name (VARCHAR) - Nom de l'utilisateur
 - email (VARCHAR, UNIQUE) - Adresse e-mail (sert aussi de login)
 - password (VARCHAR) - Mot de passe haché
 - email_verified_at (TIMESTAMP, Nullable) - Date de vérification de l'e-mail
 - remember_token (VARCHAR(100), Nullable)
 - created_at, updated_at (TIMESTAMPS)
- **Relations :**
 - Un User (Présentateur) peut avoir plusieurs Demandes (Relation Un-à-Plusieurs avec demandes).
 - Un User peut avoir plusieurs Roles (Relation Plusieurs-à-Plusieurs avec roles, gérée par la table pivot model_has_roles du package Spatie).

2. Table roles (Gérée par Spatie/laravel-permission)

Stocke les différents rôles disponibles dans l'application.

- **Colonnes principales :**
 - id (Clé Primaire, BIGINT UNSIGNED, Auto-incrément)
 - name (VARCHAR, UNIQUE pour un guard_name donné) - Nom du rôle (ex: "Secrétaire", "Présentateur", "Étudiant")
 - guard_name (VARCHAR) - Généralement "web" pour les applications web.
 - created_at, updated_at (TIMESTAMPS)
- **Relations :**
 - Un Role peut être assigné à plusieurs Users (Relation Plusieurs-à-Plusieurs avec users).

3. Table model_has_roles (Table pivot gérée par Spatie)

Lie les utilisateurs (ou autres modèles) à leurs rôles.

- **Colonnes principales :**
 - role_id (Clé Étrangère vers roles.id)
 - model_type (VARCHAR) - Le nom de la classe du modèle (ex: App\Models\User)
 - model_id (BIGINT UNSIGNED) - L'ID de l'instance du modèle (ex: user.id)
 - (Clé primaire composite sur role_id, model_type, model_id)

4. Table demandes

Stocke les demandes de séminaire soumises par les Présentateurs.

- **Colonnes principales :**
 - id (Clé Primaire, BIGINT UNSIGNED, Auto-incrément)
 - user_id (Clé Étrangère vers users.id, onDelete='cascade') - L'ID du Présentateur ayant soumis la demande.
 - theme (VARCHAR) - Thème proposé pour le séminaire.
 - description (TEXT, Nullable) - Description ou motivation.
 - statut (ENUM: 'en_attente', 'validee', 'refusee', DEFAULT 'en_attente') - Statut actuel de la demande.
 - date_souhaitee (DATE, Nullable) - Date de présentation souhaitée par le Présentateur.
 - date_presentation_validee (DATE, Nullable) - Date fixée par la Secrétaire si la demande est validée.
 - commentaire_secretaire (TEXT, Nullable) - Commentaire de la Secrétaire (ex: motif de refus, informations supplémentaires).
 - created_at, updated_at (TIMESTAMPS)
- **Relations :**
 - Une Demande appartient à un User (le Présentateur).

- Une Demande (si validée) peut avoir un Seminaire (Relation Un-à-Un avec seminaires).

5. Table seminaires

Stocke les informations sur les séminaires qui ont été validés et programmés.

- **Colonnes principales :**
 - id (Clé Primaire, BIGINT UNSIGNED, Auto-incrément)
 - demande_id (Clé Étrangère vers demandes.id, UNIQUE, onDelete='cascade') - Lien vers la demande d'origine.
 - date_presentation (DATE) - Date effective de la présentation.
 - chemin_resume (VARCHAR, Nullable) - Chemin vers le fichier du résumé soumis par le Présentateur.
 - chemin_fichier_presentation (VARCHAR, Nullable) - Chemin vers le fichier de la présentation finale soumis par la Secrétaire.
 - est_publie (BOOLEAN, DEFAULT false) - Indique si le séminaire est publié et visible par les Étudiants.
 - created_at, updated_at (TIMESTAMPS)
- **Relations :**
 - Un Seminaire appartient à une Demande.
 - Un Seminaire peut avoir plusieurs Documents (Relation Un-à-Plusieurs avec documents).

6. Table documents

Stocke les informations sur les fichiers liés aux séminaires (si vous utilisez cette approche en plus des champs chemin_ dans la table seminaires).

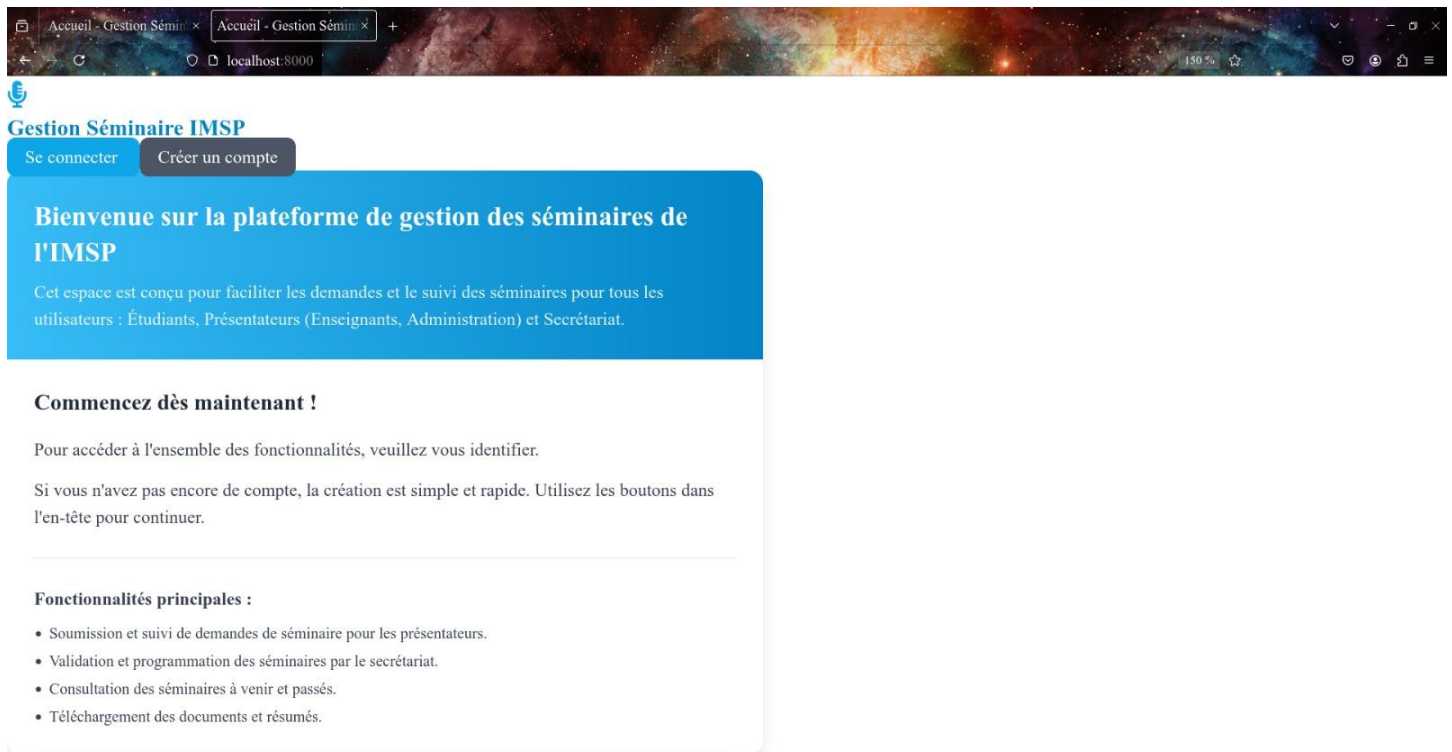
- **Colonnes principales :**
 - id (Clé Primaire, BIGINT UNSIGNED, Auto-incrément)
 - seminaire_id (Clé Étrangère vers seminaires.id, onDelete='cascade') - Lien vers le séminaire concerné.
 - type_document (VARCHAR) - Type de document (ex: 'resume', 'presentation_finale', 'autre_materiel').
 - nom_fichier_original (VARCHAR) - Nom original du fichier uploadé.
 - chemin_stockage (VARCHAR) - Chemin où le fichier est physiquement stocké sur le serveur.
 - mime_type (VARCHAR, Nullable) - Type MIME du fichier (ex: 'application/pdf').
 - taille_fichier (BIGINT UNSIGNED, Nullable) - Taille du fichier en octets.
 - created_at, updated_at (TIMESTAMPS)
- **Relations :**
 - Un Document appartient à un Seminaire.

10-Visualisation des Relations Principales:

- **User (1) -- (0..*) Demande** : Un utilisateur (présentateur) peut faire plusieurs demandes. Une demande est faite par un seul utilisateur.
- **Demande (1) -- (0..1) Séminaire** : Une demande validée peut donner lieu à un seul séminaire. Un séminaire provient d'une seule demande.
- **Séminaire (1) -- (0..*) Document** : Un séminaire peut avoir plusieurs documents associés. Un document appartient à un seul séminaire.
- **User (0..*) -- (0..*) Role** : Un utilisateur peut avoir plusieurs rôles, et un rôle peut être assigné à plusieurs utilisateurs (via la table model_has_roles).

Résultats du site en images :

Accueille :





Gestion Séminaire IMSP

[Dashboard](#) [Nouvelle Demande](#) [Mes Demandes](#)

Rachidi Idoniyi
rachidon2002@gmail.com

[Mon Profil](#)
[Déconnexion](#)

Tableau de Bord du Présentateur

Espace Présentateur

Gérez vos demandes de séminaires et suivez leurs avancements.

Bienvenue, Rachidi Idoniyi !

Actions rapides :

- [Soumettre une nouvelle demande de séminaire](#)
- [Voir le statut de vos demandes](#)
- (Bientôt) Consulter les séminaires disponibles.
- (Bientôt) Télécharger des documents.
- [Consulter les séminaires disponibles](#)



Gestion Séminaire IMSP

[Dashboard](#)

Harrison
secretaire@example.com

[Mon Profil](#)
[Déconnexion](#)

Tableau de Bord Secrétaire

Bienvenue sur votre espace de gestion, Harrison !

Vos tâches principales incluent :

- Consulter et traiter les demandes de séminaires.
- Publier les séminaires et leurs informations.
- Gérer les documents (upload/téléchargement).
- Envoyer des notifications.
- [Gérer les demandes de séminaires](#)
- [Gérer et Publier les Séminaires](#)
- [Consulter les séminaires disponibles](#)
- (Bientôt) Gérer les documents (upload/téléchargement).
- (Bientôt) Envoyer des notifications spécifiques.

Gestion Séminaire IMSP

[Dashboard](#)
Harrisson
secretaire@example.com
[Mon Profil](#)
[Déconnexion](#)

Gestion des Demandes de Séminaire

Demandes en Attente

Aucune demande en attente de traitement.

Gestion Séminaire IMSP

[Dashboard](#) [Nouvelle Demande](#) [Mes Demandes](#)
Rachidi Idoniyi
rachidon2002@gmail.com
[Mon Profil](#)
[Déconnexion](#)

Mes Demandes de Séminaire

Suivi de vos demandes

Consultez ici l'état de toutes les demandes de séminaire que vous avez soumises.

[Nouvelle Demande](#)

Thème	Soumis le	Statut	Date Souhaitée	Date Présentation	Validée	Résumé	Commentaire Secrétaire
k!lbhnl	27/05/2025 09:09	Refusée	08/06/2025	-	-	-	demandes trop proches
jhhhhxy	27/05/2025 08:15	Validée	06/06/2025	06/06/2025		Soumis Modifier	ykhhhhhhl

Créer un compte

Name

Email

Password

Confirm Password

Je souhaite m'inscrire en tant que :

Étudiant

[Already registered?](#)

Register

Profile

Profile Information

Update your account's profile information and email address.

Name

Email

Save

Update Password

Ensure your account is using a long, random password to stay secure.

Current Password

New Password

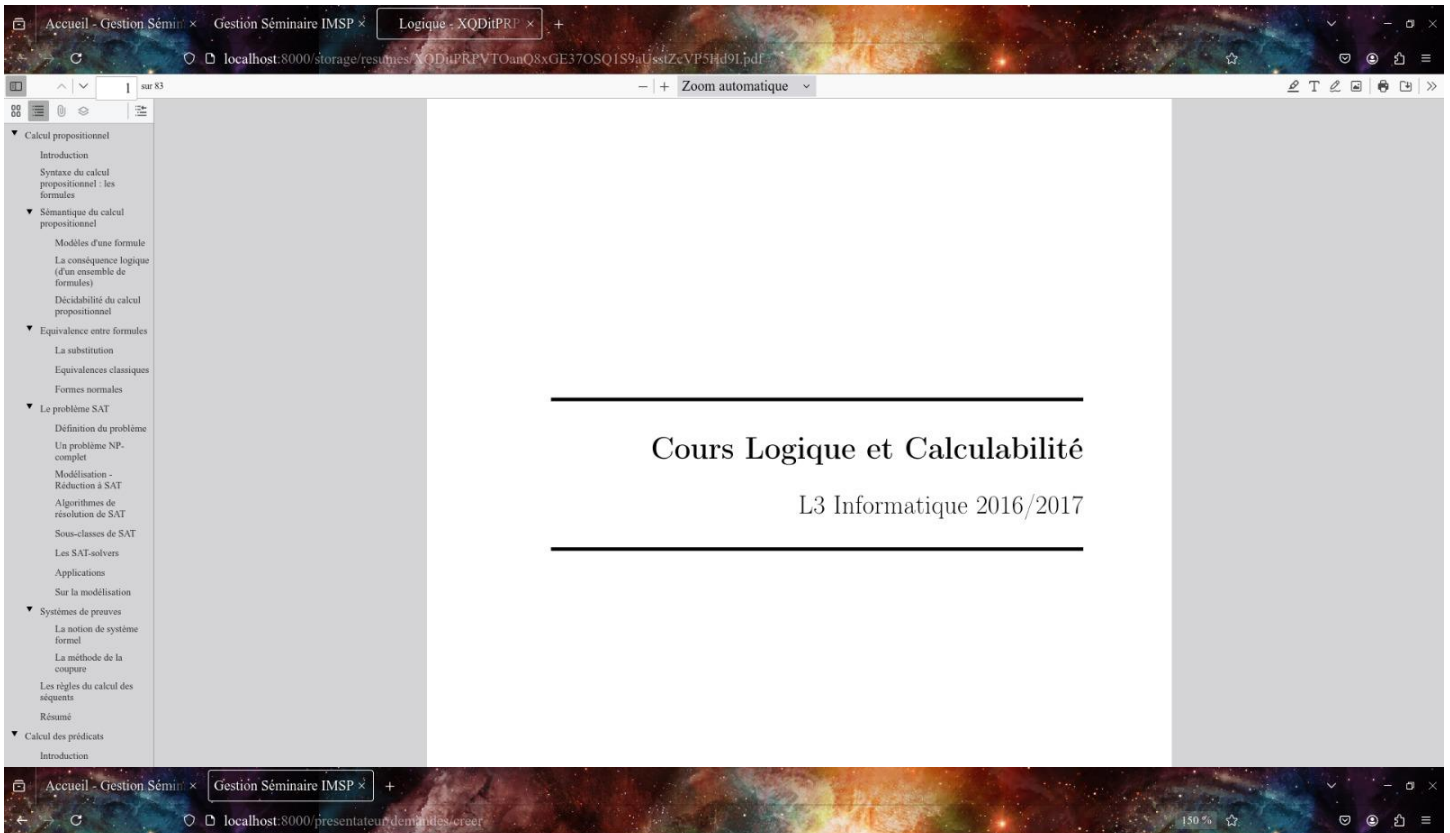
Confirm Password

Save

Delete Account

Once your account is deleted, all of its resources and data will be permanently deleted. Before deleting your account, please download any data or information that you wish to retain.

Delete Account



Gestion Séminaire IMSP

[Dashboard](#) [Nouvelle Demande](#) [Mes Demandes](#)

Rachidi Idoniyi

rachidon2002@gmail.com

[Mon Profil](#)

[Déconnexion](#)

Soumettre une nouvelle demande de séminaire

Nouvelle Demande

Veuillez remplir les informations ci-dessous.

Thème du séminaire

Description / Motivation

Date souhaitée pour la présentation



Gestion Séminaire IMSP

[Dashboard](#)

Harrisson

secretaire@example.com

[Mon Profil](#)

[Déconnexion](#)

Séminaires Disponibles

Programme des Séminaires

Consultez les séminaires à venir et passés qui ont été publiés.

jhhhhxy

Présenté par : Rachidi Idoniyi

Date : 06/06/2025

Description : yxikkkkkkkkkkk

[Voir le Résumé](#)



Gestion Séminaire IMSP

[Dashboard](#) [Nouvelle Demande](#) [Mes Demandes](#)

Rachidi Idoniyi

rachidon2002@gmail.com

[Mon Profil](#)

[Déconnexion](#)

Séminaires Disponibles

Programme des Séminaires

Consultez les séminaires à venir et passés qui ont été publiés.

jhhhhxy

Présenté par : Rachidi Idoniyi

Date : 06/06/2025

Description : yxikkkkkkkkkkk

[Voir le Résumé](#)



Gestion Séminaire IMSP

Dashboard

Mohamed SABI KPARE

sabi@gmail.com

Mon Profil

Déconnexion

Séminaires Disponibles

Programme des Séminaires

Consultez les séminaires à venir et passés qui ont été publiés.

jhhhhxy

Présenté par : Rachidi Idoniyi

Date : 06/06/2025

Description : yxikkkkkkkkkkk

[Voir le Résumé](#)



Gestion Séminaire IMSP

Dashboard

Harrisson

secretaire@example.com

Mon Profil

Déconnexion

Gestion et Publication des Séminaires

Liste des Séminaires Programmés

Thème	Présentateur	Date Présentation	Résumé Soumis	Statut	Actions
jhhhhxy	Rachidi Idoniyi	06/06/2025	Oui	Publié	<div><div>Dépublier</div><div>Gérer Présentation</div></div>



Gestion Séminaire IMSP

[Dashboard](#)

Mohamed SABI KPARE

sabi@gmail.com

[Mon Profil](#)

[Déconnexion](#)

Tableau de Bord de l'Étudiant

Bienvenue, Mohamed SABI KPARE !

Sur votre tableau de bord, vous pourrez :

- [Consulter les séminaires disponibles](#)
- (Bientôt) Télécharger les documents des séminaires passés.



Gestion Séminaire IMSP

[Dashboard](#)

Harrisson

secretaire@example.com

[Mon Profil](#)

[Déconnexion](#)

Uploader la Présentation Finale pour : jhhhhxy

Fichier de Présentation Final

Séminaire du 06/06/2025 par Rachidi Idoniyi

Fichier de présentation (PDF, PPT, PPTX - Max 10MB) [Parcourir...](#) Aucun fichier sélectionné.

[Annuler](#) [Soumettre la présentation](#)

...