

Plateforme Collaborative d'Apprentissage Hard & Soft Skills

1. Introduction

1.1. Contexte

EMSI Share est une plateforme éducative intelligente qui modernise l'apprentissage via :

- Ressources pédagogiques (cours, quiz, exercices).
- Collaboration (forums, partage de documents).
- Évaluation interactive (quiz notés, classements).

Public cible :

- Élèves ingénieurs : Accès aux ressources, participation aux quiz.
- Enseignants : Création de contenu, modération.
- Administration : Gestion des événements/annonces.

1.2. Objectifs

Pour les Élèves ingénieurs

- Trouver des ressources rapidement.
- Proposer des documents utiles.
- S'évaluer via des quiz interactifs.

Pour les Enseignants

- Publier des cours/quiz facilement.
- Animer des discussions sur le forum.
- Corriger automatiquement les quiz.

Pour l'École

- Centraliser les contenus pédagogiques.
- Promouvoir les événements académiques.
- Réduire la paperasse administrative.

1.3. Technologies Clés

Composant	Technologie	Rôle
Frontend	React.js	Interface dynamique et réactive
Backend	Django / PHP	API RESTful
Base de données	MongoDB Atlas / MySql	Stockage des données
Hébergement	Netlify (Front), Vercel (Back)	Déploiement continu

2. Fonctionnalités Détaillées

2.1. Authentification & Rôles

Méthodes de connexion :

- Email + mot de passe (Bcrypt).
- OAuth 2.0 (Google, Facebook).

Rôles :

- Étudiant : Consulter, proposer des ressources.
- Enseignant : Créer du contenu, modérer les forums.

2.2. Tableaux de Bord

Dashboard Étudiant

- **Hard Skills :**
 - Proposer une ressource (formulaire structuré).
 - Recherche avancée (type, date, format, popularité).
- **Soft Skills :**
 - Événements académiques.
 - Compétitions interactives.
- **Quiz :**
 - Passer des quiz notés (texte, images, vidéos).
 - Voir ses statistiques et progrès.

Dashboard Enseignant

- **Création de Quiz :**
 - Questions types : texte, image, vidéo.
 - Timer pour les examens.
- **Gestion des Ressources :**
 - Valider ou refuser les propositions.

2.3. Modules Clés

Module	Fonctionnalités	Technologies
Quiz	Création, participation, stats, classement	React, FastAPI, MongoDB
Ressources	Upload, recherche, validation (PDF, vidéo, etc.)	AWS S3, FastAPI, MongoDB
Forum	Discussions, commentaires, modération	Socket.io, FastAPI

3. Architecture Technique

3.1. Backend (Python – FastAPI)

- Connexion MongoDB avec Motor.
- Organisation en routers + services + schemas.
- Documentation auto via Swagger.

3.2. Communication API React <-> FastAPI

Action	Endpoint FastAPI	Méthode	Payload
S'inscrire	/auth/signup	POST	email, mdp
Se connecter (JWT)	/auth/login	POST	email, mdp
Récupérer quiz	/quiz/{id}	GET	token
Créer quiz (enseignant)	/quiz/	POST	titre, questions
Poster une ressource	/resources/	POST	fichier, titre
Envoyer un post sur forum	/forum/post	POST	titre, contenu

4. Répartition des rôles (Équipe de 4)

Personne A — Chef de Projet & Backend Lead (FastAPI)

- Gestion des endpoints (/login, /signup, /quizzes, etc.).
- Connexion MongoDB avec Motor.
- Organisation technique (routers, services).
- Swagger/OpenAPI.

Personne B — Frontend Lead & UI/UX Designer (React)

- Création des interfaces React : authentification, dashboards, quiz, forum, ressources.
- Intégration des appels API avec axios.
- Utilisation de react-query pour la synchronisation asynchrone, le cache, le retry.
- Gestion des états critiques avec Redux Toolkit (authentification, quiz, ressources).
- Sécurisation des routes (redirections selon les rôles, expiration de session).
- Création de composants réutilisables (formulaires, cartes, listes, tableaux).
- Design responsive avec Tailwind CSS.
- Maquettes et prototypage sous Figma.

Personne C — Fullstack Logiciel & Sécurité

- Algorithmes : scoring quiz, filtres avancés.
- Sécurité backend (sanitization, anti-spam, validation).
- Logs, debug, analyse de performance.

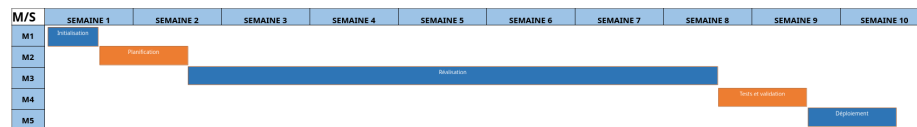
Personne D — QA, Tests & Contenu

- Tests (unitaires, intégration) avec pytest, httpx.
 - Données démo : ressources, quiz, forum.
 - Vérification ergonomie.
-

5. Planning & Livrables (Diagramme de Gantt)

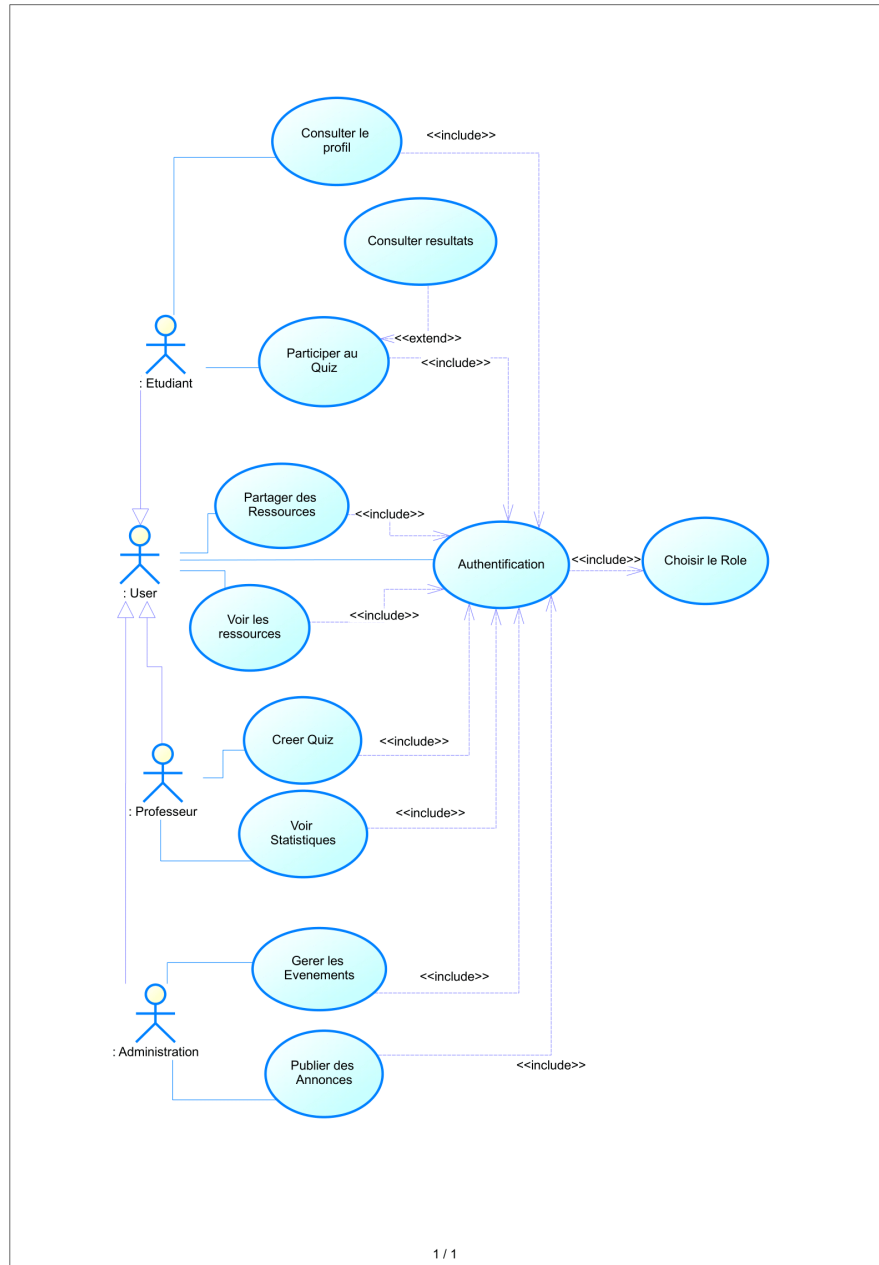
Phase	Durée	Livrables
Conception	2 sem.	Cahier des charges
Développement	6 sem.	Conception + MVP (frontend + backend)
Tests	1 sem.	Tests complets, debug
Déploiement	1 sem.	Mise en ligne (Netlify/Vercel), doc

Diagramme de Gantt



6. Diagramme de Cas d'Utilisation (Use Case Diagram)

Étudiants, Enseignants, Administration



7. Conclusion

EMSI Share est une solution complète et collaborative pour :

- Stimuler l'engagement étudiant.
 - Faciliter le travail des enseignants.
 - Numériser la pédagogie.
-