

Projet SI40 / WE4B : Plateforme pédagogique style « Moodle »

Contexte et fonctionnalités à implémenter

Dans la continuité du projet précédent (SI40/WE4A), la seconde partie portera toujours sur le développement d'une plateforme pédagogique inspirée de Moodle. Cette fois, l'accent sera mis sur la réalisation du **front-end avec Angular**. Pour la base de données, il est obligatoire d'utiliser **MongoDB**, la base NoSQL orientée documents, particulièrement adaptée à la gestion flexible de données complexes en format JSON. Vous êtes amenés à améliorer la plateforme en ajoutant les fonctionnalités suivantes :

1. Historique de connexion et d'activités des utilisateurs

Cette fonctionnalité vise à enregistrer les actions réalisées par les utilisateurs sur la plateforme : connexions, consultations de cours, soumissions de devoirs, etc. Les données sont stockées sous forme de logs JSON dans une collection dédiée. Cela permet notamment de suivre l'activité des utilisateurs et produire des statistiques.

2. Gestion des discussions et forums

La plateforme doit proposer un système de forums où les utilisateurs peuvent créer des sujets, répondre à des messages, et échanger. Chaque forum peut être stocké dans une collection (ex, forums), avec des sous-documents pour les messages. Les discussions peuvent être regroupées par cours, avec des autorisations spécifiques selon le rôle (étudiant, enseignant).

3. Gestion des devoirs et systèmes de notation

Cette fonctionnalité permet aux étudiants de soumettre leurs devoirs, et aux enseignants d'y accéder, de les corriger et de les noter.

- Les étudiants peuvent téléverser leurs devoirs via un formulaire Angular.
- Les enseignants consultent les soumissions, donnent une note et un commentaire.
- La gestion de l'état du devoir (en attente, corrigé, en retard, etc.) est optionnelle (bonus), mais valorisée si bien implémentée.

4. Stockage des fichiers et métadonnées

Les fichiers transmis (cours, devoirs, ressources) doivent être stockés avec leurs métadonnées : nom, type, taille, date, etc. Les fichiers sont généralement stockés :

- Soit sur le serveur (avec seulement les infos dans MongoDB),
- Soit dans MongoDB via GridFS (si vous souhaitez intégrer le fichier directement dans la base) (<https://www.mongodb.com/docs/manual/core/gridfs/>)

Back-end : libre choix de la technologie

Vous pouvez utiliser le langage/framework back-end de votre choix, à condition de développer une API qui permet à l'application Angular de communiquer avec MongoDB.

Structure globale des données (MongoDB)

Même si MongoDB n'utilise pas de schémas relationnels classiques (comme MCD ou MLD), il est essentiel de réfléchir à la structure logique des données. Cela implique de concevoir des collections regroupant des documents JSON, où chaque document suit une logique clé/valeur adaptée aux besoins fonctionnels, à savoir :

- Organiser les données par collections thématiques.
- Structurer chaque document en identifiant clairement les champs nécessaires.
- Réfléchir à l'usage de sous-documents ou de références selon les cas (embedding vs referencing).

Soutenance et livrables

Soutenance

Pour clôturer le projet, chaque groupe devra réaliser une soutenance orale accompagnée de la remise d'un rapport écrit et du code source complet.

- Contenu attendu lors de la soutenance (côté SI40)

Il est essentiel de présenter les éléments suivants :

- Explication des choix de modélisation
 - Présentation des collections créées dans MongoDB.
 - Description de la structure des documents JSON.
 - Justification de l'utilisation du referencing ou de l'embedding à chaque fois que l'un des deux est utilisé.
- Présentation des choix techniques pour le back-end
 - Langage et framework choisis
 - Description de l'intégration entre MongoDB et Angular : comment l'API back-end permet au front-end d'accéder et de manipuler les données.
- Démonstration en direct
 - Lancement de mongosh.
 - Affichage d'exemples d'enregistrements depuis vos collections pour illustrer la structure réelle des données stockées.

Livable

Le livable devra obligatoirement comprendre :

- Le code source complet du projet :
 - Front-end Angular
 - Back-end
 - Base de données
- Un rapport écrit incluant :
 - Les technologies utilisées et l'architecture générale.
 - La description des fonctionnalités développées.
 - Les choix de modélisation de la base.
 - Les captures ou extraits de données si nécessaire.

Dates

- Le tout devra être soumis au plus tard le **28 Juin à 23:59**
- Les soutenances auront lieu **le 30 juin et le 1^{er} juillet**. Les créneaux seront communiqués plus tard.