

Conception d'une Application Web pour l'Accès aux Sujets d'Examens

1. Introduction:

Dans le cadre de ce projet, j'ai identifié un problème majeur rencontré par les étudiants : la dispersion des sujets d'examen et la difficulté à accéder à des ressources de révision pertinentes. Pour répondre à ce problème, je propose une application web permettant de centraliser les sujets d'examen.

Mon approche vise à fournir une solution minimale mais fonctionnelle, en mettant en place une interface simple et intuitive permettant aux étudiants d'accéder facilement aux sujets avec leurs pairs.

2. Architecture Modèle/Vue/Contrôleur (MVC)

2.1 Modèle de Données:

La base de données relationnelle utilisée est SQLite. Elle comprend les tables suivantes :

Table	Champs	Description
<i>users</i>	(id, name, email, password)	Stocke les utilisateurs enregistrés
<i>subjects</i>	(id, title, description, year, course, file_path, formation_id)	Contient les sujets d'examen et Stocke les fichiers associés aux sujets d'examen
<i>formations</i>	(id, nom, description)	Stocke les formations disponibles

2.2 Vues

Les vues sont réalisées en HTML, CSS et JavaScript et incluent :

- **Page d'accueil (/)** : Affiche un message de bienvenue, une brève description du site et un bouton de connexion dans le header.
- **Page de connexion (/login)** : Formulaire de connexion pour les utilisateurs.
- **Page d'inscription (/register)** : Permet aux nouveaux utilisateurs de s'inscrire.
- **Page des formations (/formations)** : Affiche la liste des formations disponibles avec filtres.
- **Page des sujets d'examen (/sujets/:formation_id)** : Affiche la liste des sujets d'examen pour une formation spécifique.
- **Page d'un sujet (/subject/:id)** : Affiche le sujet sélectionné avec ses fichiers et commentaires.
- **Page d'ajout d'un sujet (/upload)** : Permet l'ajout de nouveaux sujets avec fichiers joints.

2.3 Routes et Logique de Contrôle

- **GET /** : Affiche la page d'accueil avec un message de bienvenue, une description du site et un bouton de connexion dans le header.
- **GET /login** : Affiche la page de connexion.
- **POST /login** : Vérifie les informations de connexion et ouvre une session.
- **GET /register** : Affiche la page d'inscription.
- **POST /register** : Enregistre un nouvel utilisateur.
- **GET /formations** : Affiche la liste des formations disponibles.
- **GET /sujets/:formation_id** : Affiche la liste des sujets d'examen disponibles pour une formation spécifique.
- **GET /subject/:id** : Affiche les détails d'un sujet spécifique.
- **POST /upload** : Permet l'ajout d'un nouveau sujet et de fichiers associés.
- **GET /logout** : Déconnecte l'utilisateur.
- **DELETE /subject/:id** : Supprime un sujet spécifique (réservé aux enseignants/admins)

3. Technologies utilisées

- Back-end : Framework en Python (Flask) .
- Base de données : SQLite.
- Front-end : HTML, CSS et JavaScript (avec possibilité d'utiliser des frameworks comme React (js) ou Bootstrap (css)).
- Sécurité : Sessions utilisateur avec gestion des cookies sécurisés.

4. Conclusion

Cette solution minimale répond au problème identifié en centralisant les sujets d'examen et en simplifiant leur accès. Elle permet aux étudiants d'améliorer leur préparation en offrant une plateforme intuitive et efficace. Des fonctionnalités avancées, telles que la vérification des accès, la possibilité pour les étudiants d'échanger leurs résumés, un moteur de recherche amélioré ou encore un système de notation des sujets, pourraient être envisagées pour de futures améliorations.