

Cahier des charges – Projet Fil Rouge DWWM (EduLien)

1. Introduction

M2i, leader de la formation IT, Digital et Management depuis plus de 30 ans, propose plus de 2 400 cursus en présentiel, à distance ou en e-learning. Dans le cadre de la formation Développeur Web et Web Mobile, les stagiaires doivent réaliser un projet fil rouge reproduisant une situation professionnelle réaliste : le développement d'une application éducation (EduLien) .

2. Présentation du projet

Le projet Edulien est un projet éducatif qui vise initialement à rendre continu le processus éducatif entre les enseignants et les étudiants.

Les enseignants et les élèves peuvent suivre le processus éducatif sur leurs propres pages.

Par exemple, les enseignants peuvent envoyer des devoirs, des notes ou des annonces aux élèves par le biais de l'application et les suivre.

3. Choix des Technologies

Frontend :

- PHP pour la génération dynamique de contenu côté serveur
- HTML5 et CSS3 pour la structure et le style des pages
- Bootstrap pour la mise en page réactive et les composants UI

Backend :

- PHP avec architecture MVC pour organiser le code
- Routes et contrôleurs modulaires : fichiers séparés pour l'authentification, les classes, les utilisateurs, etc.
- Gestion des responsabilités via des fichiers Controller .php dédiés

Base de données :

MySQL relationnelle avec :

- table user (enseignants et élèves distingués par rôle)
- table class (sujets gérés par les enseignants)
- table lesson(sujets gérés par les enseignants)
- table lesson_students (liens entre élèves et classes)
- table homeworks (devoirs créés par les enseignants, liés aux classes et aux élèves)
- Clés étrangères pour l'intégrité des données
- Identifiants personnalisés générés au format S25-XXXXXX ou T25-XXXXXX

Sécurité :

- Gestion des rôles : distinction entre enseignants et élèves
- Contrôle d'accès au backend selon les permissions
- Stockage sécurisé des mots de passe : les mots de passe sont hachés avant d'être enregistrés dans la base de données

Structure du projet :

- **Architecture MVC claire** : séparation des models, controllers et routes
- **Dossiers utils et config** : centralisation des fonctions réutilisables et des constantes
- **Frontend** : PHP, HTML, CSS et Bootstrap organisés par components, pages et assets
- **Backend** : contrôleurs, repository, serveur et configuration
- **Database** : scripts MySQL pour la création des tables user, class, class_students et homeworks
- **Documentation** : README.md complète du proje

4. Versions

Version 1 – Fonctionnalités

- Les utilisateurs peuvent s'inscrire et se connecter avec un rôle enseignant ou élève.
- Les enseignants peuvent créer des classes, et lessons
- Ajouter des élèves à leurs lessons,
- Et envoyer des devoirs à ces élèves.
- Les élèves peuvent ensuite consulter et suivre les devoirs qui leur ont été attribués.

Version 2 – Fonctionnalités supplémentaires

- Les enseignants peuvent désormais attribuer des notes aux élèves.
- Ils peuvent également publier des annonces à destination de leurs classes ou de certains élèves.
- Les élèves peuvent consulter ces notes et annonces depuis leur espace personnel.

Version 3 – Amélioration du compte utilisateur

- Les utilisateurs peuvent désormais mettre à jour leurs informations personnelles (nom, e-mail, mot de passe, etc.).
- Une option pour ajouter ou modifier la photo de profil a été ajoutée.
- La sécurité du compte et l'expérience utilisateur ont été améliorées.

5. Modèle de données simplifié

Tables minimales :

- user(id, e-mail, mot de passe, nom, prénom, role)
- class (id , nom, teacherId)
- lesson (id , nom, classId, teacherId)
- lessonStudent (id, nom de l'étudiant, nom de famille de l'étudiant, email, identifiant de l'étudiant, lessonId, classId, teacherId)

6. Fonctionnalités à implémenter

- Profil indiquant les informations relatives à l'utilisateur
- Menu de gauche avec des composants basés sur les rôles (profil, leçon, devoirs...)

7. Contraintes techniques

Le site doit : - fonctionner localement (localhost), - être hébergé proprement (organisation des fichiers), - utiliser Git pour la gestion de version, - proposer un code clair, commenté et fonctionnel.

8. Livrables attendus

- Code source sur GitHub - Base de données exportée (.sql) - README.md clair avec consignes d'installation - Documentation technique (PDF/Word) avec captures et explications - Présentation orale avec démonstration