

# 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'un application education (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 :

- **React.js** avec composants fonctionnels et hooks (useState, useEffect, useContext)
- **React Router DOM** pour la navigation client-side
- **React Toastify** pour les messages de notification utilisateur
- **CSS modules** et Bootstrap pour le style

### Backend :

- **Node.js + Express.js** avec architecture RESTful
- **Gestion des routes modulaire** : routes séparées pour l'authentification, les classes, les utilisateurs, etc.
- **Contrôleurs dédiés** : séparation des responsabilités via des fichiers controller.js
- **Gestion centralisée des erreurs** (fichier errorUtils.js)

### Base de données :

- **MySQL** relationnelle avec :
  - table user (enseignants et élèves distingués par rôle)
  - table class (sujets gérés par les enseignants)
  - table class\_students (liens entre élèves et classes)
- **Clés étrangères** pour l'intégrité des données
- **Identifiants personnalisés** générés au format S25-XXXXXX ou T25-XXXXXX

## Sécurité :

- **Gestion des rôles** (enseignant / élève)
- **Contrôle d'accès au backend** selon les permissions
- **Stockage sécurisé des mots de passe** (hash à ajouter en production)

## Structure du projet :

- Architecture **MVC** claire (models, controllers, routes)
- **Dossiers utils/config** pour centraliser les fonctions réutilisables et constantes

## Outils de développement :

- **Git / GitHub** pour la gestion de version
- **Dotenv** pour les variables d'environnement
- **Postman** utilisé pour les tests d'API

## 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**,
- **Ajouter des élèves** à leurs classes,
- 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 :

- utilisateur (id, e-mail, mot de passe, nom, prénom)
- classe ( id , nom, créateur )
- classStudent (id, nom de l'étudiant, nom de famille de l'étudiant, email, identifiant de l'étudiant)

## 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