**Cahier des charges – Plateforme de gestion des projets d'étudiantes**

**1. Titre du projet :**

**EduProjects** :

Plateforme web de gestion et de suivi des projets des étudiants universitaires

**2. Objectif général :**

Créer une application web interactive permettant aux étudiants de soumettre, suivre et gérer leurs projets de fin d’études tout en facilitant le travail d’encadrement et de validation pour les professeurs et administrateurs.

**3. Objectifs spécifiques :**

* Offrir aux étudiants une interface intuitive pour créer et soumettre leurs projets.
* Permettre aux encadrants de suivre l’évolution des projets et d’ajouter leurs remarques.
* Faciliter l’échange d'informations et de documents autour des projets.

**4. Utilisateurs concernés :**

* **Étudiants** : Création et suivi des projets
* **Encadrants (Professeurs)** : Encadrement, validation, notation
* **Administrateurs** : Gestion de la plateforme, utilisateurs, projets

**5. Technologies utilisées (MERN Stack)**

**Backend :**

* **Node.js + Express**
* **MongoDB** (base de données NoSQL)
* **JWT** (authentification)
* **Multer** (gestion des fichiers)

**Frontend :**

* **Next.js**

**6.Contraintes techniques du projet :**

**Technologies imposées / utilisées**

* **Backend** : laravel
* **Frontend** : Next.js (ou React.js si besoin)
* **Authentification** : JWT (JSON Web Tokens)
* **Gestion des fichiers** : Multer pour upload (PDF, Word...)

**Base de données prévue**

* **MongoDB** (
  + Stockage des utilisateurs, projets,fichiers...

**Hébergement envisagé :**

* **Backend** :
* **Frontend** :
* **Base de données** : MongoDB Atlas

**Mobilité / Responsiveness :**

* **Interface responsive** compatible avec :
  + **PC (desktop)**
  + **Tablettes**
  + **Smartphones**

**Tableau des utilisateurs et rôles :**

| **Nom de l’utilisateur** | **Description (Rôle réel)** | **Rôles et permissions dans l’application** | **Implémentation technique** |
| --- | --- | --- | --- |
| **Étudiant** | Étudiant réalisant un projet de fin d’études | - S’inscrire / se connecter - Créer un projet - Soumettre des fichiers (rapports, documents...) - Consulter les remarques des encadrants - Modifier son projet (tant qu’il n’est pas validé) - Voir l’état d’avancement du projet | **Laravel** : Middleware role:etudiant, policies sur les projets **Next.js** : Routes protégées avec vérification du rôle "étudiant" via JWT |
| **Encadrant** | Professeur encadrant un ou plusieurs étudiants | - Se connecter - Voir les projets qu’il encadre - Laisser des remarques et feedbacks - Télécharger les fichiers soumis - Valider ou refuser une étape - Noter le projet final | **Laravel** : Middleware role:encadrant, policies pour projets assignés **Next.js** : Affichage conditionnel des interfaces d’évaluation selon le rôle |
| **Administrateur** | Membre de l’administration (ex. chef de département) | - Gérer tous les comptes utilisateurs - Assigner les encadrants aux projets - Créer / modifier / supprimer des projets - Voir toutes les statistiques - Exporter les données - Gérer les cycles, les filières | **Laravel** : Middleware role:admin, accès complet via policies **Next.js** : Accès complet au dashboard avec vérification du rôle "admin" via JWT |

**Liste des fonctionnalités attendues – EduProjects**

**1. Présentation générale**

**But de l’application** :

EduProjects est une plateforme web permettant aux étudiants de gérer et suivre leurs projets de fin d’études, tout en facilitant l’encadrement, la validation et le suivi par les enseignants et les administrateurs.

**Utilisateurs cibles** :

* Étudiants
* Encadrants (enseignants)
* Administrateurs

**2. Liste des fonctionnalités par type d’utilisateur**

**Étudiant**

* Créer un compte et se connecter
* Créer un projet de fin d’étude
* Ajouter des informations au projet (titre, description, domaine…)
* Soumettre des rapports ou fichiers (PDF, Word…)
* Suivre l’évolution de son projet
* Consulter les remarques de l’encadrant
* Modifier son projet (tant qu’il n’est pas validé)
* Voir le statut du projet (en cours, validé, refusé)

**Encadrant**

* Se connecter à son espace
* Consulter les projets assignés
* Laisser des commentaires ou remarques
* Valider ou refuser une étape du projet
* Télécharger les rapports des étudiants
* Noter les projets
* Envoyer un retour aux étudiants

**Administrateur**

* Gérer les comptes utilisateurs (ajout, modification, suppression)
* Assigner les encadrants aux projets
* Gérer les filières / années / cycles
* Exporter des données (rapports, résultats)

**3. User Stories**

* **Étudiant** :  
  *En tant qu’étudiant, je veux soumettre mon rapport final, afin que mon encadrant puisse le corriger.*
* **Encadrant** :  
  *En tant qu’encadrant, je veux laisser des commentaires sur un projet, afin de guider l’étudiant dans son travail.*
* **Administrateur** :  
  *En tant qu’administrateur, je veux assigner un encadrant à un projet, afin d’assurer le suivi pédagogique.*

**4. Priorisation des fonctionnalités**

| **Fonctionnalité** | **Rôle concerné** | **Priorité** |
| --- | --- | --- |
| Authentification JWT | Tous | **Essentielle (MVP)** |
| Création et gestion des projets | Étudiant | **Essentielle (MVP)** |
| Upload de fichiers (Multer) | Étudiant | **Essentielle (MVP)** |
| Commentaires/Feedback | Encadrant | **Essentielle (MVP)** |
| Validation et notation | Encadrant | **Essentielle (MVP)** |
| Tableau de bord avec stats | Administrateur | **Secondaire** |
| Export de données | Administrateur | **Secondaire** |
| Notifications par mail | Tous | **Bonus** |
| Recherche par mots-clés / filtres | Tous | **Bonus** |

**planification du projet : EduProjects**

**1. Liste des tâches à réaliser**

**Phase 1 : Analyse & Spécifications**

* Définition des user stories
* Élaboration du modèle de données (MongoDB)
* Rédaction des schémas de use case

**Phase 2 : Conception**

* Création de maquettes UI/UX (Figma)
* Définition des routes frontend (Next.js)
* Définition des endpoints backend (Laravel API REST)
* Schéma de base de données MongoDB

**Phase 3 : Développement Backend (Laravel + MongoDB)**

* Configuration du projet Laravel avec MongoDB
* Authentification JWT (étudiant, encadrant, admin)
* CRUD projets (étudiant)
* Attribution des projets (admin)
* Commentaires et validation (encadrant)
* Upload de fichiers via Multer
* Protection des routes par rôles

**Phase 4 : Développement Frontend (Next.js)**

* Authentification et gestion du JWT côté client
* Interfaces par rôle :
  + Espace étudiant (création, suivi projet, upload)
  + Espace encadrant (feedback, validation)
  + Espace admin (gestion utilisateurs, attribution)
* Tableau de bord responsive
* Intégration avec l’API Laravel

**Phase 5 : Tests**

* Tests de routes protégées frontend
* Tests fonctionnels avec Postman
* Validation responsive (PC, tablette, mobile)

**Phase 6 : Finalisation & Présentation**

* Génération des rapports (PDF)
* Préparation d’une démo fonctionnelle
* Rédaction de la documentation utilisateur et technique

**2. Estimation de la durée**

| **Phase** | **Durée estimée** |
| --- | --- |
| Analyse & Spécifications | 2 jours |
| Conception | 2 jours |
| Dev Backend | 8 jours |
| Dev Frontend | 6 jours |
| Tests | 2 jours |
|  |  |
| Finalisation & Présentation | 2 jours |
| **Total estimé** | **22 jours** |

**3. Estimation des coûts**

| **Ressource** | **Coût estimé** |
| --- | --- |
| Hébergement Laravel (Render) | Gratuit (tiers gratuit) |
| MongoDB Atlas | Gratuit (plan Free Tier) |
|  |  |
| Outils de design (Figma) | Gratuit |
| GitHub (version gratuite) | Gratuit |

**4. Outils nécessaires au développement**

* **Backend** : Laravel, MongoDB, JWT, Multer
* **Frontend** : Next.js, (Tailwind CSS ou bootstrap)
* **IDE** : VS Code
* **Versioning** : GitHub
* **Tests** : Postman
* **Design** : Figma