

# Cahier des Charges

**Projet :** Développement d'une application de gestion des projets intégrés

## 1. Contexte du projet

Le développement de l'application de gestion des projets intégrés vise à fournir une plateforme centralisée permettant de gérer les projets d'étudiants au sein d'une institution académique. L'application permettra d'ajouter des sujets de projet, d'affecter des équipes et des tuteurs, de suivre l'avancement des projets, et d'évaluer les résultats.

L'objectif est de simplifier le suivi de chaque projet en permettant une gestion claire et accessible pour tous les intervenants : étudiants, tuteurs et administrateurs.

## 2. Objectifs du projet

- **Ajouter des sujets de projet :** L'application permettra aux administrateurs d'ajouter de nouveaux sujets de projets avec des informations détaillées (titre, description, compétences requises, etc.).
- **Affectation des équipes et tuteurs :** Les équipes d'étudiants seront créées et affectées à des tuteurs de manière automatisée ou manuelle par les administrateurs.
- **Suivi d'avancement :** Chaque équipe pourra soumettre des rapports d'avancement de leur projet. Les tuteurs pourront suivre ces rapports et donner des retours.
- **Évaluation des projets :** À la fin du projet, les tuteurs pourront évaluer le travail des équipes, en fonction des critères définis.

## 3. Périmètre du projet

**Fonctionnalités principales :**

### 1. Gestion des utilisateurs :

- **Administrateurs :** Ajouter/modifier/supprimer des utilisateurs, gérer les rôles (étudiants, tuteurs).
- **Étudiants :** S'inscrire à des projets, suivre l'avancement, soumettre des rapports.
- **Tuteurs :** Assigner des projets aux étudiants, suivre l'avancement, évaluer les projets.

### 2. Gestion des projets :

- **Ajout de sujets de projet** par les administrateurs : titre, description, compétences, durée estimée, etc.
- **Création et gestion des équipes d'étudiants** : affectation manuelle ou automatique aux sujets de projet.

- **Suivi de l'avancement :** Les étudiants soumettent des rapports d'avancement, et les tuteurs donnent des retours.

### 3. Affectation des tuteurs :

- **Affectation manuelle ou automatique** des tuteurs aux projets.
- Les tuteurs peuvent visualiser l'avancement des projets de leurs équipes et apporter des feedbacks.

### 4. Évaluation des projets :

- Les tuteurs évaluent les projets sur la base de critères prédéfinis (qualité du travail, respect des délais, innovation, etc.).
- Les évaluations sont enregistrées et accessibles pour les administrateurs et les étudiants.

## 4. Rôles des utilisateurs

### • Administrateur :

- Gère l'ensemble des utilisateurs (ajout, modification, suppression).
- Ajoute et modifie les sujets de projet.
- Gère l'affectation des tuteurs et des équipes.
- Visualise les évaluations finales.

### • Tuteur :

- Gère et suit les projets des étudiants affectés.
- Donne des retours sur les rapports d'avancement.
- Évalue les projets à la fin de leur développement.

### • Étudiant :

- Crée une équipe ou se joint à une équipe existante.
- Soumet des rapports d'avancement.
- Reçoit des retours et évaluations de son tuteur.

## 5. Exigences fonctionnelles

### Gestion des projets :

- L'administrateur peut ajouter, modifier et supprimer des sujets de projet.
- Les sujets doivent comporter des informations telles que : titre, description, compétences requises, etc.

### Création des équipes :

- Les étudiants peuvent être affectés à une équipe manuellement par l'administrateur ou automatiquement selon des critères définis (ex : compétences, disponibilité).

#### Suivi d'avancement :

- Les étudiants soumettent des rapports d'avancement à intervalles réguliers (hebdomadaires, mensuels).
- Les tuteurs peuvent commenter ces rapports et donner des retours.

#### Évaluation :

- Les tuteurs peuvent attribuer des notes ou des évaluations qualitatives basées sur des critères définis.
- Les évaluations sont accessibles pour les administrateurs et les étudiants.

### 6. Exigences techniques

- **Plateforme** : L'application doit être web-based et compatible avec les navigateurs modernes (Chrome, Firefox, Safari, Edge).
- **Langages et technologies** :
  - **Frontend** : HTML, CSS, JavaScript, React.js ou Angular
  - **Backend** : Node.js avec Express ou Django en Python.
  - **Base de données** : MySQL ou PostgreSQL.
  - **Authentification** : Authentification par email/mot de passe, avec possibilité d'intégration LDAP pour l'institution.
- **Sécurité** :
  - Les données doivent être sécurisées par SSL.
  - Authentification et autorisation strictes selon les rôles.
- **Hébergement** : L'application doit être hébergée sur un serveur dédié ou cloud (AWS, Azure, etc.).

---

### 7. Délais et livrables

Phase	Livrables	Délai
Phase 1 : Analyse des besoins	Cahier des charges, spécifications fonctionnelles	2 semaines
Phase 2 : Conception	Diagrammes UML, architecture du système	2 semaines

Phase	Livrables	Délai
<b>Phase 3 : Développement initial</b>	Maquettes, prototype fonctionnel	4 semaines
<b>Phase 4 : Développement complet</b>	Application fonctionnelle complète avec tests	10 semaines
<b>Phase 5 : Validation et tests</b>	Test final, rapport de tests, documentation utilisateur	2 semaines

## 8. Critères d'acceptation

L'application sera considérée comme terminée et validée si :

- Toutes les fonctionnalités décrites dans ce cahier des charges sont implémentées et fonctionnelles.
- Les tests de validation ont été réalisés avec succès.
- La sécurité et la performance répondent aux exigences spécifiées.
- La documentation technique et utilisateur est complète.