Une image contenant texte, Police, graphisme, Graphique

Le contenu généré par l’IA peut être incorrect.

**# ANALYSEETCONCEPTIONDANSLEPROCESSUSDUGENIELOGICIEL**

Projet Tutoré



**PROJET**

**Système de Gestion de la Scolarité**

*Encadreur Pédagogique :*

**Dr. (MC) KOUSSOUBE**

Présenter par :

**Georges Christian RAPONTCHOMBO**

**Odette Fidèle Gossi LOKOSSOU**

**Table des matières**

[Introduction 3](#_Toc198309195)

[**I.** **Présentation de l'équipe** 4](#_Toc198309196)

[**1.** **Constitution de l’équipe Scrum** 4](#_Toc198309197)

[**2.** **Organisation du travail** 4](#_Toc198309198)

[**3.** **Choix technologiques et Architecture** 5](#_Toc198309199)

[**4.** **Cadrage initial et conception préliminaire** 9](#_Toc198309200)

[**II.** **Product Backlog** 10](#_Toc198309201)

[**1.** **Sprint 2 – Module2 : Gestion des élèves et des classes** 11](#_Toc198309202)

[**2.** **Sprint 3 – Module3 : Saisie des notes & génération de bulletin** 12](#_Toc198309203)

[**3.** **Sprint 4 – Module4 : Communication & tableau de bord** 12](#_Toc198309204)

[**III. Sprint Backlog** 13](#_Toc198309205)

[**III.** **Planification des sprints** 14](#_Toc198309206)

[**Diagrammes** 16](#_Toc198309207)

[**1.** **Diagramme de cas d’utilisation (Sprint 1)** 16](#_Toc198309208)

[**Concusion** 19](#_Toc198309209)

# 

# **Introduction**

**Nom du projet : EduConnect Smart**

**Contexte général**

Le système éducatif actuel, notamment dans de nombreuses écoles primaires et secondaires, est encore largement basé sur des processus manuels pour la gestion de la scolarité : inscriptions sur papier, bulletins physiques, communications lentes entre les enseignants, les parents et l’administration. Cela entraîne des retards, des erreurs et une surcharge administrative.

Avec la transformation digitale en cours dans de nombreux secteurs, le **besoin d’un système numérique intelligent** de gestion de la scolarité devient crucial pour moderniser et fiabiliser l’ensemble du cycle éducatif.

**Domaine d’application**

**EduConnect Smart** est un projet de développement logiciel appartenant au domaine de l’**éducation numérique (EdTech)**. Il vise à concevoir une **plateforme de gestion scolaire** centralisée et évolutive, destinée à être utilisée par :

* Les **établissements scolaires** (administration, direction),
* Les **enseignants**,
* Les **élèves**,
* Et les **parents d’élèves**.

**Besoins identifiés**

Le projet répond à plusieurs besoins majeurs observés dans les établissements éducatifs :

* Digitaliser l’inscription, la gestion des dossiers élève et les bulletins scolaires.
* Faciliter le suivi du parcours académique des élèves en temps réel.
* Améliorer la communication entre les différents acteurs (administration, enseignants, parents).
* Réduire les erreurs humaines liées à la gestion manuelle.
* Offrir un tableau de bord intelligent à chaque utilisateur selon son profil.

**Pourquoi ce projet est important ?**

**EduConnect Smart** permettra de :

* **Automatiser** les processus administratifs chronophages.
* **Fluidifier** la circulation des informations scolaires.
* **Offrir une traçabilité complète** des données académiques.
* **Renforcer l'implication des parents** dans le suivi scolaire des enfants.
* Servir de **base évolutive** pour d'autres modules (gestion des absences, paiements en ligne, messagerie, etc.).
  1. **Présentation de l'équipe**

1. **Constitution de l’équipe Scrum**

Le projet **EduConnect Smart** est réalisé par une équipe de deux étudiants, tous deux développeurs, sous la supervision directe de l’enseignant qui endosse le rôle de **Product Owner**.

|  |  |  |
| --- | --- | --- |
| Nom | Rôle | Description |
| Christian RAPONTCHOMBO | **Scrum Master & Développeur** | Il assure la coordination de l’équipe, veille au respect de la méthodologie Scrum et participe activement au développement du projet. Il organise les sprints, facilite la communication et aide à résoudre les obstacles rencontrés. |
| Fidèle LOKOSSOU | **Développeur** | Il collabore à l’analyse, la conception et la réalisation technique du projet. Il contribue à l’implémentation des fonctionnalités définies dans le backlog. |
| Pr. (MC)KOUSSOUBE | **Product Owner** | Il représente le client et définit les priorités fonctionnelles. Il est garant de la vision produit, valide les user stories du backlog, oriente les choix fonctionnels et donne son retour à chaque sprint. |

1. **Organisation du travail**

* **Réunions** : points réguliers en présentiel ou à distance pour suivre l’avancement
* **Outil de gestion de projet** : Trello pour visualiser le backlog et suivre les tâches par sprint
* **Outil de versioning** : GitHub ou GitLab pour stocker et collaborer sur le code source
* **Communication interne** : Slack ou Google Meet pour les échanges rapides entre les membres

## **Choix technologiques et Architecture**

1. **Stack technique sélectionnée**

Le projet **EduConnect Smart** repose sur une stack technologique moderne, orientée web, qui favorise la modularité, l’évolutivité et une bonne séparation des responsabilités. Voici les technologies envisagées :

**Frontend (Interface utilisateur)**

**Framework** : Vue.js



**Vue.js** est un **Framework JavaScript progressif** utilisé pour **construire des interfaces utilisateur interactives**. Il se concentre principalement sur la **vue (View)** dans le modèle **MVC** (Modèle-Vue-Contrôleur), mais peut aussi s’étendre avec des bibliothèques ou outils pour créer des applications web complètes.

* **Langage** : TypeScript



**TypeScript** est un **superset de JavaScript** développé par **Microsoft**, qui ajoute des **types statiques** au langage. Il permet d’écrire du code **plus structuré, fiable et maintenable**, en détectant les erreurs **avant l’exécution**, grâce à la **vérification de type à la compilation**.

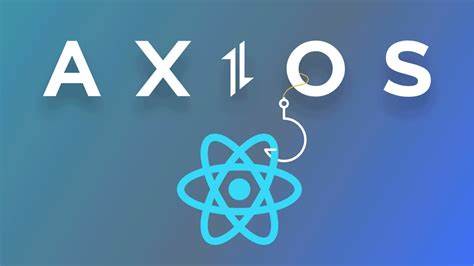
* **Gestion des routes** : Vue Router



**Vue Router** est la **bibliothèque officielle de routage** pour Vue.js. Elle permet de gérer la **navigation entre les pages ou les vues** d’une application **monopage (SPA)** sans recharger la page entière. Elle permet également de **définir des routes dynamiques**, de gérer des **paramètres d’URL**, des **niveaux d’accès** (guard), ou encore des **routes imbriquées**.

* **Appels API** : Axios

**Axios** est une **bibliothèque JavaScript** basée sur les **promesses** qui permet de **faire des requêtes HTTP** (GET, POST, PUT, DELETE, etc.) vers un serveur web depuis le navigateur ou depuis un environnement Node.js. C’est l’une des bibliothèques les plus populaires pour communiquer avec des **API REST**



* **Design System** : Tailwind CSS

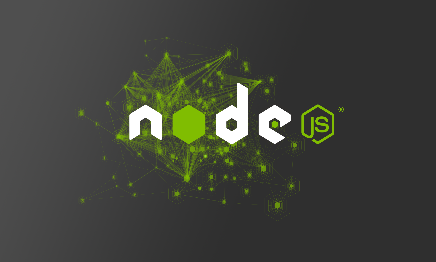


**Tailwind CSS** est un **Framework CSS utilitaire-first** qui permet de **concevoir des interfaces rapidement** en utilisant directement des **classes utilitaires** dans le HTML. Contrairement à Bootstrap qui fournit des composants prédéfinis, Tailwind vous donne les **briques de base** pour **créer vos propres designs sur mesure**.

**Backend (Serveur et logique métier)**

* **Langage** : Node.js

**Node.js** est un **environnement d’exécution JavaScript côté serveur**, construit sur le moteur **V8 de Google Chrome**. Il permet d’exécuter du code JavaScript **en dehors du navigateur**, ce qui en fait un excellent choix pour créer des **applications web back-end, des API REST, des scripts serveur et des outils CLI**.



* **ORM** : Prisma

**Prisma** est un **ORM (Object-Relational Mapping)** moderne pour **Node.js et TypeScript**, conçu pour faciliter la **connexion, l’écriture et la lecture dans une base de données relationnelle** (comme **PostgreSQL**, **MySQL**, **SQLite**, etc.).

Une image contenant Police, Graphique, logo, blanc

Le contenu généré par l’IA peut être incorrect.

* **ORM** : Express.js

**Express.js** est un **Framework web minimaliste et flexible** pour **Node.js** qui permet de **créer des applications web et des API REST** rapidement et facilement. Il fournit un ensemble de fonctionnalités essentielles pour gérer les **routes, les requêtes HTTP, les middlewares**, et l’organisation du serveur. C’est l’un des outils les plus utilisés dans l’écosystème JavaScript côté back-end.

****

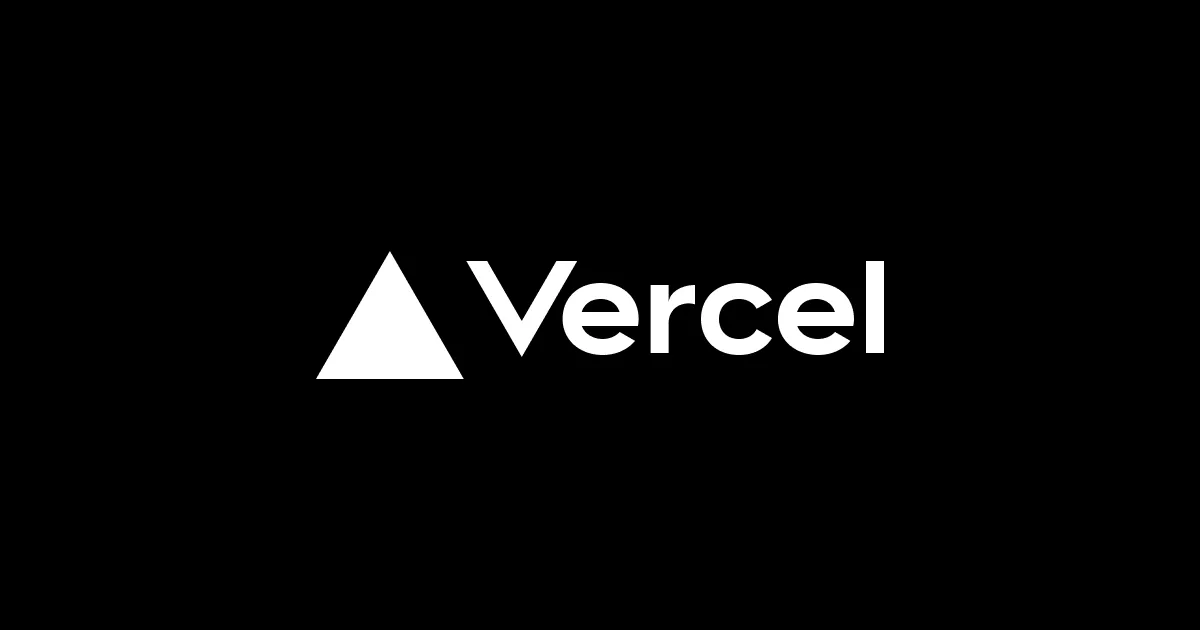
**Base de données**

* **Système** : MYSQL



**MySQL** est un **Système de Gestion de Base de Données Relationnelle (SGBDR)** **open-source**, basé sur le langage **SQL (Structured Query Language)**. Il est utilisé pour **stocker, organiser, gérer et interroger des données structurées** dans des bases relationnelles. C’est l’un des SGBD les plus populaires au monde, souvent utilisé avec des applications web.

* **Hébergement distant** : Vercel

****

**Vercel** est une **plateforme de déploiement cloud** spécialisée dans les **applications front-end modernes**. Elle permet de **déployer automatiquement** des sites statiques ou dynamiques créés avec des frameworks comme **Next.js**, **React**, **Vue.js**, **Nuxt**, **Svelte**, etc.

**Environnement de développement**

* **IDE** : WebStorm

**WebStorm** est un **Environnement de Développement Intégré (IDE)** puissant et moderne, développé par **JetBrains**, spécialement conçu pour le **développement JavaScript, TypeScript, HTML, CSS**, et leurs frameworks associés (comme **Vue.js, React, Angular**, etc.).

C’est un outil payant, mais très complet, utilisé par les développeurs professionnels pour **créer, gérer, tester et déboguer des applications web**.



* **Contrôle de version** : Git + GitHub

1. **Architecture de l’application**

L'application **EduConnect Smart** adopte une architecture **client-serveur** avec séparation claire entre le frontend et le backend, selon le modèle MVC étendu :

[Utilisateur]

↓

[Interface Web (Vue.js)]

(Axios)

↓

[API REST (Express.js)]

↓

[Base de données PostgreSQL]

**C. Caractéristiques de l’architecture**

* Architecture **modulaire** permettant de créer facilement de nouveaux modules (ex. : messagerie, gestion des paiements).
* Application conçue pour être **responsive** et compatible mobile/tablette.
* Découplage clair entre **affichage**, **logique métier** et **stockage des données**.
* Possibilité d’intégration future avec des services externes (e-mail, paiements en ligne, etc.)

# **Cadrage initial et conception préliminaire**

1. **Problématique**

Dans de nombreuses écoles primaires et secondaires, la gestion de la scolarité repose encore sur des processus manuels : inscriptions papier, bulletins imprimés, suivi académique approximatif, communication lente entre les parents et les enseignants. Cette organisation provoque :

* Des **retards dans la transmission d’information**,
* Des **erreurs fréquentes de saisie et d’archivage**,
* Une **difficulté à assurer un suivi individualisé** de chaque élève.

Le projet **EduConnect Smart** vise à **numériser ces processus** pour offrir une plateforme centralisée, moderne et accessible à tous les acteurs éducatifs.

1. **Objectifs du projet**

Le système **EduConnect Smart** a pour finalités de :

1. **Digitaliser la gestion scolaire** : inscriptions, emplois du temps, bulletins.
2. **Améliorer la communication** entre enseignants et parents via un espace sécurisé.
3. **Permettre un accès personnalisé** aux données scolaires (selon le profil : administrateur, enseignant, parent).
4. **Offrir une solution évolutive** pouvant intégrer d’autres modules à l’avenir (absences, paiements, évaluations en ligne).
5. **Garantir la sécurité et la confidentialité** des données scolaires.
6. **Conception préliminaire avancée**

Voici une première vue d’ensemble de la conception fonctionnelle du projet :

**Modules de base identifiés :**

* **Module Authentification** : Connexion sécurisée, gestion des rôles (admin, enseignant, parent).
* **Module Élève / Classe** : Inscription, consultation des dossiers scolaires, affectation aux classes.
* **Module Enseignant** : Création des bulletins, attribution des notes, messagerie interne.
* **Module Parent** : Consultation des bulletins, réception de notifications de l’école.
* **Module Admin** : Gestion complète des utilisateurs, accès aux statistiques et aux archives.

**Navigation prévue :**

* Un **tableau de bord** adapté à chaque rôle.
* Un **menu latéral** avec accès aux modules selon les droits.
* Une **interface responsive**, fluide et intuitive.
  1. **Product Backlog**

**Structure du backlog :**

Chaque user story est présentée ainsi :

* **ID**
* **User Story (format agile)**
* **Critères d’acceptation**
* **Priorité (MoSCoW)** :
  + **M**ust have (obligatoire)
  + **S**hould have (important)
  + **C**ould have (optionnel)
  + **W**on’t have now (ne sera pas inclus)

**Sprint 1 – Connexion, Préinscription, Dashboards, Génération PDF**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Rôle | User Story (format agile) | Critères d’acceptation | Priorité | Fonction concernée |
| US01 | Visiteur | En tant que visiteur, je veux accéder à un formulaire de préinscription pour m’inscrire | Formulaire complet avec validation côté client (nom, prénom, date, domaine, etc.) | Must | Préinscription |
| US02 | Étudiant | En tant qu’étudiant, je veux accéder à un dashboard personnalisé pour consulter mes infos | Accès à un tableau de bord avec ses informations de préinscription | Must | Dashboard étudiant |
| US03 | Étudiant | En tant qu’étudiant validé, je veux obtenir une attestation de préinscription au format PDF | Génération automatique d’un PDF contenant les informations saisies | Must | Génération PDF |
| US04 | Utilisateur | En tant qu’utilisateur, je veux pouvoir me connecter via un formulaire sécurisé | Système de connexion simple avec vérification d’identifiants | Must | Authentification / Connexion |

1. **Sprint 2 – Module2 : Gestion des élèves et des classes**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | ID | User Story | Critères d’acceptation | Priorité | | US05 | En tant qu’**admin**, je veux **ajouter, modifier, supprimer des classes**, afin d’organiser les élèves | Interface CRUD pour les classes | M | | US06 | En tant qu’**admin**, je veux **affecter des élèves à des classes**, afin de structurer les groupes | Sélection des élèves inscrits + affectation par classe | M | | US07 | En tant qu’enseignant, je veux **voir la liste des élèves par classe**, afin de préparer mes cours | Vue filtrée par classe avec nom/prénom/âge | S | |
|  |

1. **Sprint 3 – Module3 : Saisie des notes & génération de bulletin**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Critères d’acceptation | Priorité |
| US08 | En tant qu’**enseignant**, je veux **saisir les notes des élèves**, afin de produire un bulletin scolaire | Interface de saisie par matière et par élève | M |
| US09 | En tant qu’**admin**, je veux **générer un bulletin PDF pour chaque élève**, afin de les imprimer ou les envoyer | Génération PDF avec logo, nom, matières, moyennes, appréciations | M |
| US10 | En tant qu’**élève ou parent**, je veux **voir ou télécharger le bulletin**, afin de suivre ma progression | Interface avec lien de téléchargement sécurisé | S |

1. **Sprint 4 – Module4 : Communication & tableau de bord**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Critères d’acceptation | Priorité |
| US11 | En tant qu’**admin**, je veux **envoyer un message ou une annonce à un groupe d’élèves ou parents**, afin de partager des informations | Système de messagerie simple ou de notification interne | S |
| US12 | En tant qu’utilisateur, je veux **voir un tableau de bord clair avec mes informations clés**, afin d’avoir un aperçu global | Dashboard personnalisé selon le rôle (admin, enseignant, parent) | C |

# **III. Sprint Backlog**

**1. Objectif du Sprint 1**

L’objectif de ce premier sprint est de mettre en œuvre les bases fonctionnelles du système de gestion des admissions, en se concentrant sur les éléments suivants :

* 📥 **Préinscription** en ligne des étudiants
* 🔐 **Connexion** sécurisée des utilisateurs
* 📊 **Dashboards** personnalisés pour les étudiants et les administrateurs
* 📄 **Génération automatique de documents PDF** (attestation de préinscription)

**📋 2. User Stories sélectionnées – Sprint 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Rôle | User Story (format agile) | Critères d’acceptation | Priorité | Fonction concernée |
| US01 | Visiteur | En tant que visiteur, je veux accéder à un formulaire de préinscription pour m’inscrire | Formulaire complet avec validation des champs côté client et serveur (nom, prénom, date, domaine, etc.) | Must have | Préinscription |
| US02 | Étudiant | En tant qu’étudiant, je veux accéder à un dashboard personnalisé pour consulter mes infos | Tableau de bord affichant les informations de préinscription et le statut | Must have | Dashboard étudiant |
| US03 | Étudiant | En tant qu’étudiant validé, je veux obtenir une attestation de préinscription au format PDF | Génération automatique d’un PDF contenant les informations validées, téléchargeable | Must have | Génération PDF |
| US04 | Utilisateur | En tant qu’utilisateur, je veux pouvoir me connecter via un formulaire sécurisé | Formulaire de connexion avec vérification des identifiants et gestion des erreurs | Must have | Authentification |

**🧩 3. Modules techniques à implémenter**

|  |  |
| --- | --- |
| Module | Fonctionnalités à inclure |
| Dashboard | Création du tableau de bord étudiant et administrateur, avec informations contextualisées |
| Préinscription | Formulaire de saisie, enregistrement en base, interface admin pour validation |
| Génération PDF | Attestation de préinscription générée automatiquement avec mise en page officielle |
| Authentification | Formulaire de connexion avec vérification, redirection selon le rôle, messages d’erreur |

# **Planification des sprints**

**Sprint 1 – Connexion, Préinscription, Dashboards, Génération PDF**

* **Durée estimée** : 1 à 3 jours
* **Objectif** : Mise en place du module de gestion des admissions
* **Fonctionnalités développées** :
  + Accès au formulaire de préinscription
  + Enregistrement des données
  + Génération automatique de l’attestation PDF
  + Prévisualisation / téléchargement du PDF
  + Connexion sécurisée (étudiant/admin)
  + Dashboards simplifiés pour étudiant et admin
* **Modules liés** :
  + Étudiant (modèle + formulaire)
  + Préinscription
  + PDF Génération
  + Authentification
* **Livrables attendus** :
  + Diagramme de classes (Étudiant, Dossier, Document)
  + Cas d’utilisation (Préinscription, Validation)
  + Génération de PDF fonctionnelle avec données réelles

**🟨 Sprint 2 – Gestion des comptes et sécurité**

* **Durée estimée** : 5 à 7 jours
* **Objectif** : Mise en place du système d’authentification avancée et gestion des utilisateurs
* **Fonctionnalités** :
  + Attribution des rôles et permissions (Admin, Étudiant, Enseignant)
  + Authentification renforcée (tokens, gestion des sessions)
  + Sécurisation des données sensibles en base
  + Début de la sécurisation des fichiers uploadés (ex : PDF, justificatifs)

**🟦 Sprint 3 – Paiement en ligne & gestion des frais**

* **Durée estimée** : 7 à 10 jours
* **Objectif** : Intégrer la gestion financière
* **Fonctionnalités** :
  + Intégration Mobile Money / Visa
  + Calcul automatique des frais
  + Paiement en plusieurs tranches
  + Reçus PDF générés et envoyés
  + Suivi des paiements dans le tableau de bord

**🟧 Sprint 4 – Notifications, finalisation et tests globaux**

* **Durée estimée** : 5 jours
* **Objectif** : Polissage final du MVP
* **Fonctionnalités** :
  + Notifications email (validation de dossier, reçu de paiement)
  + Optimisation UI/UX responsive
  + Tests fonctionnels et de performance
  + Corrections finales et ajustements
  + Préparation de la démo de fin de projet

**🗺️ Synthèse de la Roadmap Agile (Corrigée)**

|  |  |  |
| --- | --- | --- |
| Sprint | Objectif principal | Modules principaux |
| Sprint 0 | Cadrage + Backlog | Préparation projet, architecture initiale |
| Sprint 1 | Admissions (MVP initial) | Connexion, Préinscription, Dashboards, PDF |
| Sprint 2 | Comptes & Sécurité | Gestion des rôles, Authentification avancée |
| Sprint 3 | Paiement | Frais, paiements, reçus PDF |
| Sprint 4 | Notifications & Finalisation | Emails, UI/UX, tests, démo |

# **Diagrammes**

## **Diagramme de cas d’utilisation (Sprint 1)**

**Une image contenant ligne, diagramme, Tracé, capture d’écran

Le contenu généré par l’IA peut être incorrect.**

**Diagramme de classe (Sprint 1)**

Une image contenant texte, diagramme, Parallèle, Plan

Le contenu généré par l’IA peut être incorrect.

**Concusion**

Le projet **EduConnect Smart** s'inscrit dans une démarche résolument moderne, visant à digitaliser la gestion scolaire pour répondre aux besoins des établissements éducatifs d'aujourd'hui. Grâce à une approche agile et à l'utilisation de technologies robustes et évolut ives, cette plateforme offrira une gestion centralisée, sécurisée et efficace des données académiques.

En favorisant l'automatisation des processus administratifs, l'amélioration de la communication entre les différents acteurs et la traçabilité des informations scolaires, **EduConnect Smart** contribuera à moderniser le système éducatif et à impliquer davantage les parents dans le suivi pédagogique. Ce projet, porté par une équipe engagée et guidé par une vision claire, se veut une réponse innovante aux défis de la transformation numérique dans le secteur de l'éducation.