**apport de Planification et d'Exécution du Projet : Portail Administratif Scolaire V1.0**

**Nom du Rapport :** Plan d'Exécution Polyvalent & DevOps Intégré - Portail Administratif Scolaire

**Équipe de Développement :** Beni, Ouly, The Tidiane

**Période de Développement :** 10 Juillet - 26 Juillet 2025

**En-tête du Projet : Modules et Modèle de Base de Données**

**Modules Fonctionnels Clés à Développer :**

1. **Gestion des Utilisateurs et des Rôles** (Administrateur, Enseignant, Élève/Parent)
2. **Gestion des Élèves** (inscription, affectation aux classes, gestion des documents, identifiants uniques)
3. **Gestion des Enseignants et des Matières** (création de profils, affectation aux matières/classes, création de matières avec coefficients)
4. **Saisie des Notes et Bulletins** (saisie par enseignant, calcul automatique des moyennes, mentions, rangs, génération de bulletins PDF)
5. **Portail Élève / Parent** (consultation et téléchargement PDF des bulletins, historique optionnel)
6. **Tableau de Bord Administratif** (vue d'ensemble, statistiques, suivi de saisie des notes)
7. **Fonctionnalités Bonus :**
   * Notifications par e-mail pour la disponibilité de nouveaux bulletins.
   * Téléchargement des bulletins par groupe (pour l'administration).

**Modèle de Base de Données de Base (Inférence du Rapport) :**

Ce modèle représente les entités principales et leurs relations nécessaires pour supporter les modules fonctionnels.

* **users**
  + id (PK)
  + nom\_utilisateur (unique)
  + mot\_de\_passe (haché)
  + email (unique, optionnel)
  + role\_id (FK vers roles)
  + date\_creation
  + date\_mise\_a\_jour
* **roles**
  + id (PK)
  + nom (ex: Administrateur, Enseignant, Élève, Parent) (unique)
* **etudiants**
  + id (PK)
  + user\_id (FK vers users, nullable si pas d'accès direct)
  + nom
  + prenom
  + date\_naissance
  + classe\_id (FK vers classes)
  + identifiant\_unique\_portail (pour élève/parent)
  + date\_inscription
  + date\_mise\_a\_jour
* **enseignants**
  + id (PK)
  + user\_id (FK vers users)
  + nom
  + prenom
  + specialite
  + date\_mise\_a\_jour
* **classes**
  + id (PK)
  + nom\_classe (ex: "Terminal S1")
  + niveau\_id (FK vers niveaux - optionnel, pour structure niveaux si besoin)
  + annee\_academique\_id (FK vers annees\_academiques)
  + date\_mise\_a\_jour
* **matieres**
  + id (PK)
  + nom\_matiere
  + coefficient
  + date\_mise\_a\_jour
* **enseignant\_matiere\_classe** (table pivot pour les affectations)
  + enseignant\_id (FK vers enseignants)
  + matiere\_id (FK vers matieres)
  + classe\_id (FK vers classes)
  + (PK composite sur les trois FK)
* **notes**
  + id (PK)
  + etudiant\_id (FK vers etudiants)
  + matiere\_id (FK vers matieres)
  + enseignant\_id (FK vers enseignants)
  + periode (ex: "Trimestre 1", "Semestre 1")
  + score
  + date\_saisie
  + date\_mise\_a\_jour
* **bulletins**
  + id (PK)
  + etudiant\_id (FK vers etudiants)
  + annee\_academique\_id (FK vers annees\_academiques)
  + periode
  + date\_generation
  + chemin\_pdf (chemin vers le fichier PDF généré)
  + date\_mise\_a\_jour
* **annees\_academiques** (optionnel pour la gestion historique)
  + id (PK)
  + annee\_debut
  + annee\_fin
  + est\_courante (booléen)

**Répartition Ordonnée des Tâches avec Rotation des Responsabilités & Intégration DevOps**

Compte tenu de la période de développement du 10 au 26 juillet, la rotation des rôles de lead sera répartie sur trois semaines pour permettre à chaque membre d'endosser un rôle de leadership sur le Backend (Laravel), le Frontend Angular et le Frontend Flutter, tout en intégrant des aspects DevOps essentiels à chaque étape.

**Jour 1 (10 juillet) : Lancement du Projet & Approfondissement (3h chacun)**

* **Groupe :** Examiner ce plan détaillé. Finaliser la compréhension de toutes les exigences et contraintes. S'accorder sur les canaux de communication (Discord/Slack, stand-ups quotidiens). Mettre en place la structure du dépôt GitHub (dépôts backend, angular, flutter).
* **Beni (Lead Backend) :** Configuration du projet Laravel (composer create-project). Installation de nwidart/laravel-modules et configuration de la structure modulaire de base.
* **Ouly (Lead Angular) :** Configuration du projet Angular (ng new). Familiarisation avec Angular CLI et la structure de composants de base.
* **The Tidiane (Lead Flutter) :** Configuration du projet Flutter (flutter create). Familiarisation avec les bases de Flutter et l'arbre de widgets.

**Jour 2 (11 juillet) : Base de Données & Module Utilisateur (3h chacun)**

* **Groupe :** Conception collaborative du schéma de base de données (en utilisant le Tableau Détaillé du Schéma de Base de Données). Finaliser les tables users, roles, students. Définir les points d'accès API initiaux pour l'authentification utilisateur et la gestion des utilisateurs/rôles.
* **Beni (Lead Backend) :** Implémentation des migrations et modèles roles et users. Configuration de Laravel Sanctum pour l'authentification API. Développement du module Utilisateur/Rôle (création, lecture des rôles).
* **Ouly (Lead Angular) :** Création du module Angular Auth et du composant de connexion. Conception de la mise en page de base du tableau de bord admin (statique).
* **The Tidiane (Lead Flutter) :** Création du module Flutter Auth et de l'écran de connexion (UI uniquement). Exploration du routage Flutter pour la navigation du portail.

**Jour 3 (12 juillet) : Authentification & Module Élève (3h chacun)**

* **Groupe :** Brève synchronisation sur les points d'accès API pour la gestion des élèves.
* **Beni (Lead Backend) :** Implémentation de la logique d'authentification Laravel (connexion, déconnexion, inscription via API). Développement du module students (migrations, modèle, API CRUD de base).
* **Ouly (Lead Angular) :** Intégration de la connexion Angular avec l'API Laravel. Développement du composant Angular de gestion des élèves (liste des élèves, formulaire de base).
* **The Tidiane (Lead Flutter) :** Intégration de la connexion Flutter avec l'API Laravel. Début de la conception du tableau de bord du portail Élève/Parent (contenu statique).

**Jour 4 (13 juillet) : Module Enseignant & Matière (3h chacun)**

* **Groupe :** Brève synchronisation sur les points d'accès API pour la gestion des enseignants/matières.
* **Beni (Lead Backend) :** Développement des modules teachers et subjects (migrations, modèles, API CRUD). Implémentation de la table pivot teacher\_subject\_class.
* **Ouly (Lead Angular) :** Développement des composants Angular de gestion des enseignants et des matières (formulaires, listes).
* **The Tidiane (Lead Flutter) :** Travail sur l'UI Flutter pour l'affichage du profil de base de l'élève (données maquettes). Recherche sur les options d'affichage PDF avec Flutter.

**Jour 5 (14 juillet) : Module Saisie des Notes (3h chacun)**

* **Groupe :** Définition des points d'accès API pour la saisie des notes.
* **Beni (Lead Backend) :** Développement du module grades (migrations, modèle, API pour la saisie des notes par l'enseignant). Implémentation de l'autorisation initiale pour les enseignants (uniquement leurs matières/classes).
* **Ouly (Lead Angular) :** Développement du composant Angular de saisie des notes pour les enseignants (sélection de la classe/matière, saisie des notes).
* **The Tidiane (Lead Flutter) :** Poursuite de l'intégration de la visualisation PDF avec Flutter (contenu de remplacement). Exploration des considérations de performance de Flutter web.

**Jour 6 (15 juillet) : Tableau de Bord Admin & Affinage API (3h chacun)**

* **Groupe :** Révision de la cohérence globale de l'API.
* **Beni (Lead Backend) :** Développement des points d'accès API pour les statistiques du tableau de bord Admin (comptes, moyennes générales). Implémentation de la limitation de taux API de base (throttle global).
* **Ouly (Lead Angular) :** Remplissage du tableau de bord Admin Angular avec les données de l'API de Beni. Affinage de l'UI/UX pour les composants Angular existants.
* **The Tidiane (Lead Flutter) :** Travail sur la vue de liste des "Bulletins" de Flutter (données maquettes, UI). Recherche sur l'optimisation de la taille de l'application Flutter.

**Jour 7 (16 juillet) : Révision Mi-Projet & Rattrapage (3h chacun)**

* **Groupe :** Synchronisation complète de l'équipe : examen des progrès, identification des blocages, repriorisation si nécessaire. S'assurer que les fonctionnalités clés sont en bonne voie.
* **Individuel (Tous) :** Rattrapage des tâches en suspens de la semaine. Refactoring du code, écriture de tests unitaires de base pour les composants critiques.

**Semaine 2 : Bulletins, Sécurité, DevOps & Déploiement (17 juillet - 26 juillet)**

**Objectif Principal :** Finalisation des fonctionnalités clés (notes, bulletins, tableau de bord), intégration et focus sur les aspects de sécurité avancée, Dockerisation, CI/CD, déploiement et tests finaux.

**Jour 8 (17 juillet) : Génération des Bulletins (3h chacun)**

* **Groupe :** Discussion sur la stratégie de génération PDF (par exemple, package Laravel pour PDF).
* **Beni (Lead Flutter) :** Développement du composant Flutter pour afficher les bulletins générés et permettre le téléchargement PDF.
* **Ouly (Lead Backend) :** Développement du module report\_cards (migrations, modèle). Implémentation de la logique de calcul automatique des moyennes, mentions, rangs, appréciations. Intégration d'une bibliothèque de génération PDF (par exemple, Dompdf) pour créer les PDF des bulletins.
* **The Tidiane (Lead Angular) :** Développement du composant Angular pour l'Administrateur afin de déclencher la génération des bulletins.

**Jour 9 (18 juillet) : Fonctionnalités Bonus & Améliorations de Sécurité (3h chacun)**

* **Groupe :** Discussion sur l'implémentation des notifications par e-mail et des téléchargements groupés.
* **Beni (Lead Flutter) :** Implémentation de l'UI Flutter pour la saisie OTP lors de la connexion. Affinage de l'UI/UX Flutter pour le portail étudiant.
* **Ouly (Lead Backend) :** Implémentation de l'API pour le téléchargement groupé des bulletins (bonus). Début de l'implémentation de la logique OTP pour la connexion Administrateur (partie backend).
* **The Tidiane (Lead Angular) :** Développement de l'UI Angular pour le téléchargement groupé des bulletins (admin). Intégration du champ de saisie OTP dans le flux de connexion Angular.

**Jour 10 (19 juillet) : DevOps - Dockerisation (3h chacun)**

* **Groupe :** Révision de la stratégie Docker.
* **Beni (Lead Flutter) :** Assistance à The Tidiane pour le Dockerfile et docker-compose pour le backend Laravel.
* **Ouly (Lead Backend) :** Focus sur les optimisations de performance Angular (OnPush, lazy loading si applicable).
* **The Tidiane (Lead Angular) :** Création du Dockerfile pour le backend Laravel. Création du docker-compose.yml pour le développement local Laravel (app, db). Test de la configuration Docker en local.

**Jour 11 (20 juillet) : DevOps - CI/CD & SonarQube (3h chacun)**

* **Groupe :** Discussion sur le workflow CI/CD.
* **Beni (Lead Flutter) :** Direction de l'intégration de SonarQube pour le backend Laravel. Configuration du scanner SonarQube dans GitHub Actions.
* **Ouly (Lead Backend) :** Finalisation de l'UI Angular et assurance de l'intégration complète de l'API.
* **The Tidiane (Lead Angular) :** Mise en place du workflow GitHub Actions de base pour Laravel (construction de l'image Docker, exécution des tests). Assistance à Beni pour l'intégration de SonarQube.

**Jour 12 (21 juillet) : DevOps - Terraform & Préparation Déploiement Backend (3h chacun)**

* **Groupe :** Sélection du service d'hébergement backend final à faible budget.
* **Beni (Lead Flutter) :** Préparation de Laravel pour le déploiement en production (mise en cache de la configuration, des routes).
* **Ouly (Lead Backend) :** Préparation d'Angular pour la construction en production (ng build --configuration production).
* **The Tidiane (Lead Angular) :** Recherche et mise en place de la configuration Terraform de base pour le service backend choisi. Préparation des scripts/commandes de déploiement pour le backend.

**Jour 13 (22 juillet) : Déploiement - Angular & Backend (3h chacun)**

* **Groupe :** Coordination des déploiements.
* **Beni (Lead Flutter) :** Assistance à The Tidiane pour le déploiement backend.
* **Ouly (Lead Backend) :** Déploiement du frontend Angular sur Vercel. Test en direct.
* **The Tidiane (Lead Angular) :** Exécution de Terraform pour provisionner l'infrastructure backend (si applicable). Déploiement du backend Laravel conteneurisé sur le service à faible budget choisi. Test des points d'accès API en direct.

**Jour 14 (23 juillet) : Tests d'Intégration & Correction de Bugs (3h chacun)**

* **Groupe :** Tests d'intégration de bout en bout de tous les modules (flux Admin, Enseignant, Élève/Parent). Identification et priorisation des bugs critiques.
* **Tous :** Correction collaborative des bugs.

**Jour 15 (24 juillet) : Correction Finale de Bugs & Polissage (3h chacun)**

* **Groupe :** Résolution des bugs restants. Révision de l'UI/UX pour la cohérence. Préparation de la démo.
* **Tous :** Révision finale du code et nettoyage.

**Jour 16 (25 juillet) : Documentation & Préparation de la Démo (3h chacun)**

* **Groupe :** Préparation du script de démo et de la présentation. Création d'une courte vidéo de démo (si nécessaire et si le temps le permet).
* **Beni (Lead Flutter) :** Documentation des points d'accès API backend et de leur utilisation.
* **Ouly (Lead Backend) :** Documentation de la configuration Angular et des composants clés.
* **The Tidiane (Lead Angular) :** Documentation de la configuration Flutter et des étapes de déploiement. S'assurer que le dépôt GitHub est bien organisé avec des READMEs clairs.

**26 juillet : Soumission du Projet.**

**Principes de Collaboration Continue pour tous (y compris DevOps) :**

* **Pair Programming Fréquent :** Chaque "lead" de la semaine devrait organiser des sessions de pair programming, incluant les aspects DevOps pertinents, avec les autres membres pour les tâches importantes de sa pile. Cela favorisera le transfert de compétences et la compréhension mutuelle des environnements.
* **Revues de Code Croisées :** Tous les membres sont encouragés à revoir le code des autres dans toutes les stacks (backend, Angular, Flutter), y compris les configurations et scripts DevOps, pour assurer la qualité, la cohérence et le transfert de connaissances.
* **Stand-ups Quotidiens Renforcés :** Les réunions quotidiennes seront essentielles pour partager les progrès, les défis rencontrés (y compris les aspects DevOps), coordonner les efforts d'intégration entre les différentes piles technologiques et ajuster les priorités.
* **Documentation Vivante :** Chaque membre contribuera à la documentation des parties qu'il développe ou dont il est "lead" pour la semaine. Cette documentation inclura non seulement le code, mais aussi les procédures et les configurations DevOps mises en place.