

Site Web + API

Documentation du Portail Professeurs

Table des matières

1. Introduction
2. Architecture technique
3. Structure du projet
4. Fonctionnalités principales
5. Guide d'utilisation
6. API et endpoints
7. Base de données
8. Sécurité
9. Déploiement
10. Dépannage

1. Introduction

Le Portail Professeurs est une application web conçue pour aider les enseignants à gérer efficacement leurs classes, les absences des élèves, les réservations de salles, et à communiquer avec les autres membres de l'établissement. Cette plateforme centralisée offre une interface intuitive et responsive qui s'adapte à tous les appareils.

Objectifs du projet

- Simplifier la gestion administrative pour les enseignants
- Améliorer le suivi des absences des élèves
- Faciliter la réservation des salles

- Offrir des outils de communication entre les membres de l'équipe éducative
- Fournir des statistiques et des rapports sur les performances et l'assiduité

2. Architecture technique

Frontend

- **Framework** : Next.js 14 (App Router)
- **Langage** : TypeScript
- **UI/UX** : Tailwind CSS avec shadcn/ui
- **État client** : React Hooks
- **Requêtes API** : Fetch API

Backend

- **Framework** : ASP.NET Core 9
- **Langage** : C#
- **Architecture** : API RESTful
- **Authentification** : JWT (JSON Web Tokens)
- **WebSockets** : Pour la messagerie en temps réel

Base de données

- **SGBD** : MySQL
- **ORM** : Entity Framework Core

3. Structure du projet

Frontend (Next.js)

```
/app                # Dossier principal des routes (App Router)
/absences           # Gestion des absences
/classes            # Gestion des classes
/login              # Page de connexion
/messages           # Système de messagerie
```

```
/notifications      # Centre de notifications
/parametres         # Paramètres utilisateur
/points             # Système de points
/rapports           # Génération de rapports
/reservations       # Réservation de salles
/salles             # Gestion des salles
/statistiques       # Tableaux de bord et statistiques
/layout.tsx         # Layout principal
/globals.css        # Styles globaux
/page.tsx           # Page d'accueil

/components         # Composants réutilisables
/ui                # Composants UI (shadcn)
/header.tsx        # En-tête de l'application
/sidebar.tsx       # Barre latérale de navigation
/auth-guard.tsx    # Protection des routes authentifiées

/lib               # Services et utilitaires
/api-client.ts     # Client API centralisé
/auth-service.ts   # Service d'authentification
/absence-service.ts # Service de gestion des absences
/class-service.ts  # Service de gestion des classes
/message-service.ts # Service de messagerie
/points-service.ts # Service de gestion des points
/room-service.ts   # Service de gestion des salles
/settings-service.ts # Service de gestion des paramètres

/public            # Fichiers statiques
```

Backend (ASP.NET Core)

```
/Controllers       # Contrôleurs API
/AuthController.cs  # Authentification
/Teacher           # Contrôleurs spécifiques aux enseignants
/TeacherAbsenceController.cs
/TeacherClassController.cs
/TeacherProfileController.cs
/TeacherRoomController.cs
```

```
/TeacherSettingsController.cs
/TeacherStatController.cs
/Message          # Contrôleurs de messagerie
/MessageController.cs
/Points           # Contrôleurs de gestion des points
/PointsController.cs

/Models           # Modèles de données
/Absence.cs
/Class.cs
/Message.cs
/Notification.cs
/PointsConfig.cs
/PointsHistory.cs
/Reservation.cs
/Room.cs
/Schedule.cs
/Settings.cs
/Student.cs
/StudentPoints.cs
/Teacher.cs
/User.cs
/GestionAbsencesContext.cs # Contexte EF Core

/DTO              # Objets de transfert de données
/AbsenceDTO.cs
/AuthDTO.cs
/ClassDTO.cs
/Message
/MessageDTO.cs
/ConversationDTO.cs
/Points
/StudentPointsDTO.cs
/PointsHistoryDTO.cs
/Teacher
/TeacherProfileDTO.cs

/Services         # Services métier
/Teacher
```

```
/TeacherAbsenceService.cs
/TeacherClassService.cs
/TeacherProfileService.cs
/TeacherRoomService.cs
/TeacherSettingsService.cs
/TeacherStatService.cs
/Message
/MessageService.cs
/Points
/PointsService.cs

/WebSocket          # Gestion des WebSockets
/WebSocketHandler.cs
/WebSocketMiddleware.cs
/WebSocketExtensions.cs

/Utils              # Utilitaires
/DateOnlyJsonConverter.cs
/DateUtils.cs

/Middleware          # Middlewares personnalisés
/BearerPrefixMiddleware.cs
/RequestResponseLoggingMiddleware.cs

/Scripts            # Scripts SQL
```

4. Fonctionnalités principales

Authentification et sécurité

- Connexion sécurisée avec JWT
- Protection des routes authentifiées
- Gestion des rôles (professeur, admin)
- Déconnexion automatique après inactivité

Gestion des classes

- Liste des classes assignées à l'enseignant
- Détails de chaque classe avec liste des élèves
- Statistiques de présence par classe
- Recherche et filtrage des classes

Gestion des absences

- Saisie des absences par classe ou par élève
- Historique des absences
- Justification des absences
- Statistiques d'assiduité
- Exportation des données d'absence

Système de points

- Attribution de points aux élèves
- Classement des élèves par points
- Historique des points attribués
- Configuration des règles d'attribution

Réservation de salles

- Consultation des salles disponibles
- Réservation de créneaux horaires
- Gestion des caractéristiques des salles
- Calendrier des réservations

Messagerie

- Communication en temps réel via WebSockets
- Conversations privées entre utilisateurs
- Notifications de nouveaux messages
- Statut des messages (envoyé, livré, lu)

Rapports et statistiques

- Génération de rapports personnalisés
- Tableaux de bord avec indicateurs clés
- Graphiques d'assiduité et de performance

- Exportation des rapports en différents formats

Paramètres utilisateur

- Personnalisation du profil
- Choix du thème (clair/sombre)
- Préférences de notification
- Gestion du mot de passe

5. Guide d'utilisation

Connexion au portail

1. Accédez à la page de connexion
2. Entrez les identifiants suivants :
 - Email :
 - Mot de passe :
3. Cliquez sur "Se connecter"

Navigation dans l'application

- Utilisez la barre latérale pour accéder aux différentes sections
- Le menu supérieur contient les notifications et l'accès au profil
- Le tableau de bord principal affiche un résumé des informations importantes

Gestion des classes

1. Accédez à la section "Classes" depuis la barre latérale
2. Consultez la liste de vos classes
3. Cliquez sur une classe pour voir les détails et la liste des élèves
4. Utilisez la barre de recherche pour filtrer les classes

Saisie des absences

1. Accédez à la section "Absences" depuis la barre latérale
2. Cliquez sur "Saisir des absences"
3. Sélectionnez la classe concernée

4. Cochez les élèves absents
5. Renseignez la date et le motif si connu
6. Validez la saisie

Réservation d'une salle

1. Accédez à la section "Salles" depuis la barre latérale
2. Consultez les salles disponibles
3. Cliquez sur "Nouvelle réservation"
4. Sélectionnez la salle, la date et l'horaire
5. Indiquez le motif de la réservation
6. Validez la réservation

Utilisation du système de points

1. Accédez à la section "Points" depuis la barre latérale
2. Consultez le classement des élèves
3. Pour attribuer des points :
 - Accédez à la fiche d'un élève
 - Cliquez sur "Ajouter des points"
 - Saisissez le nombre de points et le motif
 - Validez l'attribution

Envoi d'un message

1. Accédez à la section "Messages" depuis la barre latérale
2. Cliquez sur "Nouveau message"
3. Sélectionnez le destinataire
4. Rédigez votre message
5. Envoyez le message

Génération d'un rapport

1. Accédez à la section "Rapports" depuis la barre latérale
2. Sélectionnez le type de rapport
3. Définissez les paramètres (période, classe, etc.)
4. Générez le rapport
5. Consultez-le en ligne ou exportez-le

Modification des paramètres

1. Accédez à la section "Paramètres" depuis la barre latérale
2. Modifiez votre profil, thème ou préférences
3. Enregistrez les modifications

6. API et endpoints

Authentification

- `POST /api/auth/login` - Connexion utilisateur
- `GET /api/auth/test` - Test d'authentification
- `GET /api/auth/test-professeur` - Test d'autorisation professeur

Gestion des classes

- `GET /api/teacher/classes` - Liste des classes
- `GET /api/teacher/classes/{id}` - Détails d'une classe

Gestion des absences

- `GET /api/teacher/absences` - Liste des absences
- `GET /api/teacher/absences/{id}` - Détails d'une absence
- `POST /api/teacher/absences` - Création d'une absence
- `PUT /api/teacher/absences/{id}` - Mise à jour d'une absence
- `DELETE /api/teacher/absences/{id}` - Suppression d'une absence

Gestion des salles

- `GET /api/teacher/rooms` - Liste des salles
- `GET /api/teacher/rooms/{id}` - Détails d'une salle
- `GET /api/teacher/rooms/{id}/reservations` - Réservations d'une salle

Réservations

- `GET /api/teacher/reservations` - Liste des réservations
- `GET /api/teacher/reservations/{id}` - Détails d'une réservation
- `POST /api/teacher/reservations` - Création d'une réservation
- `PUT /api/teacher/reservations/{id}` - Mise à jour d'une réservation

- `DELETE /api/teacher/reservations/{id}` - Suppression d'une réservation

Messagerie

- `GET /api/messages` - Liste des conversations
- `GET /api/messages/{userId}` - Messages avec un utilisateur
- `POST /api/messages` - Envoi d'un message
- `PUT /api/messages/{id}/read` - Marquer un message comme lu
- `WebSocket /ws` - Communication en temps réel

Système de points

- `GET /api/points/students` - Points des élèves
- `GET /api/points/students/{id}` - Points d'un élève
- `GET /api/points/ranking` - Classement des élèves
- `POST /api/points/add` - Attribution de points
- `GET /api/points/history/{studentId}` - Historique des points

Profil et paramètres

- `GET /api/teacher/profile` - Profil de l'enseignant
- `PUT /api/teacher/profile` - Mise à jour du profil
- `GET /api/teacher/settings` - Paramètres utilisateur
- `PUT /api/teacher/settings` - Mise à jour des paramètres

7. Base de données

Schéma de la base de données

La base de données est organisée autour des entités principales suivantes :

- **Users** : Utilisateurs du système (enseignants, administrateurs)
- **Teachers** : Informations spécifiques aux enseignants
- **Students** : Informations sur les élèves
- **Classes** : Classes et leurs associations avec les enseignants
- **Absences** : Enregistrements des absences des élèves
- **Rooms** : Salles disponibles pour réservation
- **Reservations** : Réservations des salles

- **Messages** : Messages échangés entre utilisateurs
- **Notifications** : Notifications système
- **Settings** : Paramètres utilisateur
- **StudentPoints** : Points attribués aux élèves
- **PointsHistory** : Historique des attributions de points
- **PointsConfig** : Configuration du système de points

Relations principales

- Un enseignant peut avoir plusieurs classes
- Une classe peut avoir plusieurs élèves
- Un élève peut avoir plusieurs absences
- Un utilisateur peut avoir plusieurs réservations
- Un utilisateur peut envoyer/recevoir plusieurs messages
- Un élève peut avoir plusieurs entrées dans l'historique des points

8. Sécurité

Authentification

- Utilisation de JWT (JSON Web Tokens) pour l'authentification
- Stockage sécurisé des tokens côté client
- Expiration automatique des tokens
- Vérification des rôles pour l'accès aux ressources

Protection des données

- Hachage des mots de passe avec SHA-256
- Validation des entrées côté client et serveur
- Protection contre les attaques CSRF
- Utilisation de HTTPS pour toutes les communications

Autorisations

- Système de rôles (professeur, admin)
- Vérification des autorisations pour chaque action
- Isolation des données par utilisateur
- Journalisation des actions sensibles

9. Déploiement

Prérequis

- Node.js 18+ pour le frontend
- .NET 9 SDK pour le backend
- MySQL 8+ pour la base de données
- Serveur web (Nginx recommandé)
- Proxy Reverse Caddy

Étapes de déploiement

1. Base de données

- Créer une base de données MySQL
- Exécuter les scripts de migration

2. Backend

- Compiler l'application ASP.NET Core
- Configurer les variables d'environnement
- Déployer sur un serveur compatible .NET

3. Frontend

- Construire l'application Next.js (`next build`)
- Déployer les fichiers statiques
- Configurer le serveur web pour servir l'application

Configuration

- Fichier `.env` pour les variables d'environnement frontend
- Fichier `appsettings.json` pour la configuration backend
- Configuration de la base de données dans `GestionAbsencesContext.cs`

10. Dépannage

Problèmes courants

Erreur de connexion

- Vérifier les identifiants (`teacher@teacher.fr` / `admin123`)
- Vérifier que le serveur backend est accessible
- Vérifier les logs du serveur pour des erreurs d'authentification

Données non affichées

- Vérifier la connexion au serveur backend
- Vérifier les requêtes dans la console du navigateur
- Vérifier que l'utilisateur a les permissions nécessaires

Erreurs WebSocket

- Vérifier que le serveur WebSocket est en cours d'exécution
- Vérifier la configuration du proxy si applicable
- Vérifier les logs du serveur pour des erreurs de connexion

Problèmes de performance

- Vérifier la charge du serveur
- Optimiser les requêtes à la base de données
- Vérifier les index de la base de données

Logs et monitoring

- Les logs frontend sont disponibles dans la console du navigateur
- Les logs backend sont stockés dans le dossier `/logs`
- Utiliser les outils de développement du navigateur pour déboguer les problèmes frontend
- Utiliser les outils de débogage de Visual Studio pour le backend

Revision #3

Created 16 May 2025 10:27:38 by qoyri

Updated 16 May 2025 10:47:55 by qoyri