

Student Absence Tracker

Application de Gestion des Absences avec .NET et React

Saif Eddine Jlassi

Chaima Zbedi

INFO 3

18 décembre 2024

Résumé

Le Student Absence Tracker est une application web moderne développée pour répondre aux besoins de gestion des absences dans le milieu éducatif. Cette solution, construite avec .NET Core et React, offre une plateforme complète et intuitive pour le suivi des présences des étudiants.

Points clés du projet :

- Interface administrative complète pour la gestion des données
- Système de suivi des absences en temps réel
- Architecture moderne avec API REST et frontend responsive
- Gestion efficace des enseignants, étudiants, classes et matières
- Sécurité renforcée avec système d'authentification intégré
- Génération de rapports et analyses des données d'absence

Cette solution vise à simplifier et optimiser le processus de gestion des absences, offrant aux établissements éducatifs un outil moderne et efficace pour le suivi de l'assiduité des étudiants.

Table des matières

1	Introduction	4
1.1	Contexte du Projet	4
1.2	Objectifs du Projet	4
1.3	Portée du Projet	4
2	Analyse des Besoins	5
2.1	Besoins Fonctionnels	5
2.1.1	Détection des Émotions	5
2.1.2	Réponses Adaptatives	5
2.1.3	Interface Utilisateur Dynamique	5
2.1.4	Authentification et Gestion des Utilisateurs	5
2.1.5	Sessions de Chat Multiples	5
2.1.6	Visualisation des Émotions	5
2.2	Besoins Non Fonctionnels	5
2.2.1	Performance	6
2.2.2	Scalabilité	6
2.2.3	Sécurité	6
2.2.4	Fiabilité	6
2.2.5	Compatibilité	6
2.2.6	Maintenabilité	6
3	Solution Proposée	7
3.1	Architecture Globale	7
3.2	Composants Clés	7
3.2.1	Interface Utilisateur Dynamique	7
3.2.2	Système de Détection des Émotions	7
3.2.3	Génération de Réponses Contextuelles	7
3.2.4	Gestion des Sessions et Authentification	7
3.2.5	Visualisation des Données Émotionnelles	7
3.3	Avantages de la Solution	7
4	Conception	8
4.1	Architecture Globale	8
4.1.1	Backend (.NET Core)	8
4.1.2	Frontend (React)	8
4.2	Modèle de Données	8
5	Implémentation du Backend	9
5.1	Structure de l'API	9
5.1.1	Contrôleurs Principaux	9
5.2	Modèles de Données	9
5.3	Entity Framework Core	9
5.4	Sécurité et Performance	9

6	Emotion Detection Model	10
6.1	Transformer Model	10
6.2	Model Training and Fine-tuning	10
6.3	Integration with Flask Backend	10
7	Implémentation Frontend	11
7.1	Structure des Composants	11
7.2	Composants Réutilisables	11
7.3	Gestion d'État	11
7.4	Communication avec l'API	11
8	Tests et Validation	12
8.1	Tests Backend	12
8.2	Tests Frontend	12
8.3	Assurance Qualité	12
8.4	Performance	12
9	Interface Utilisateur	13
9.1	Page de Connexion	13
9.2	Tableau de Bord	13
9.3	Gestion des Absences	13
9.4	Interfaces Administratives	13
9.5	Responsive Design	13
10	Conclusion	14
10.1	Réalisations	14
10.2	Perspectives d'Évolution	14
10.3	Impact et Bénéfices	14

1 Introduction

1.1 Contexte du Projet

Dans le contexte éducatif actuel, la gestion efficace des présences est devenue un enjeu majeur pour les établissements d'enseignement. Le suivi manuel des absences est chronophage et sujet aux erreurs, nécessitant une solution numérique moderne et fiable.

1.2 Problématique

Les établissements font face à plusieurs défis :

- Difficulté de suivi en temps réel des absences
- Risque d'erreurs dans la saisie manuelle
- Manque de centralisation des données
- Complexité dans la génération de rapports
- Besoin d'un accès rapide à l'historique des absences

1.3 Objectifs du Projet

Le Student Absence Tracker vise à :

- Automatiser le processus de gestion des absences
- Fournir une interface intuitive pour les enseignants
- Centraliser les données d'absence
- Faciliter le suivi et l'analyse des tendances
- Améliorer la communication entre les acteurs

1.4 Portée du Projet

Le projet englobe :

- Développement d'une API REST avec .NET Core
- Création d'une interface utilisateur avec React
- Mise en place d'une base de données SQL Server
- Implémentation d'un système d'authentification
- Développement de fonctionnalités de reporting
- Tests et documentation complète

2 Analyse des Besoins

2.1 Besoins Fonctionnels

Le Student Absence Tracker vise à offrir une solution complète pour la gestion des absences. Les besoins fonctionnels suivants ont été identifiés :

2.1.1 Gestion des Absences

- Enregistrer les absences des étudiants par cours
- Suivre les présences en temps réel
- Générer des rapports d'absences
- Visualiser l'historique des absences par étudiant

2.1.2 Gestion des Utilisateurs

- Authentification des enseignants
- Gestion des profils étudiants
- Administration des classes et matières
- Attribution des matières aux enseignants

2.2 Besoins Non Fonctionnels

2.2.1 Performance

- Temps de réponse rapide (< 2 secondes)
- Gestion simultanée de multiples utilisateurs
- Optimisation des requêtes de base de données

2.2.2 Sécurité

- Authentification sécurisée
- Protection des données personnelles
- Journalisation des actions importantes

2.2.3 Fiabilité

- Disponibilité du système de 99,9
- Sauvegarde régulière des données
- Gestion des erreurs appropriée

2.2.4 Utilisabilité

- Interface intuitive et responsive
- Support multilingue (français/anglais)
- Accessibilité WCAG 2.1 niveau AA

3 Solution Proposée

Pour répondre aux besoins identifiés, le Student Absence Tracker propose une solution moderne combinant plusieurs technologies :

3.1 Architecture Globale

L'application est conçue selon une architecture client-serveur :

- **Frontend** : Interface utilisateur développée avec React et TypeScript
- **Backend** : API REST construite avec ASP.NET Core
- **Base de données** : SQL Server pour le stockage persistant
- **Authentication** : Système d'authentification intégré

3.2 Composants Clés

3.2.1 Interface Administrative

Interface complète pour la gestion des données de base :

- Gestion des enseignants
- Gestion des étudiants
- Configuration des classes
- Définition des matières

3.2.2 Système de Suivi des Absences

- Interface de saisie rapide des absences
- Visualisation des données par classe/matière
- Génération de rapports
- Historique détaillé

3.3 Avantages de la Solution

- **Efficacité** : Processus de suivi des absences simplifié
- **Évolutivité** : Architecture modulaire extensible
- **Performance** : Technologies modernes et optimisées
- **Sécurité** : Authentification et autorisation robustes
- **Maintenance** : Code structuré et bien documenté

4 Conception

4.1 Architecture Globale

L'architecture du Student Absence Tracker est basée sur une approche moderne et modulaire :

4.1.1 Backend (.NET Core)

- API RESTful avec ASP.NET Core
- Entity Framework Core pour l'ORM
- Architecture en couches (Controllers, Services, Models)
- Gestion des migrations de base de données

4.1.2 Frontend (React)

- Interface utilisateur avec React et TypeScript
- Gestion d'état avec Context API
- Composants réutilisables avec Tailwind CSS
- Routing avec React Router

4.2 Modèle de Données

Description détaillée des entités principales :

- Teachers (Enseignants)
- Students (Étudiants)
- Classes
- Subjects (Matières)
- Absences
- TeacherSubjectClass (Relations)

5 Implémentation du Backend

Le backend du Student Absence Tracker, développé avec .NET Core, constitue le cœur du système de gestion des absences.

5.1 Structure de l'API

L'API REST est organisée selon une architecture modulaire :

5.1.1 Contrôleurs Principaux

- `TeachersController` : Gestion des enseignants et authentification
- `StudentsController` : Gestion des étudiants
- `ClassesController` : Gestion des classes
- `SubjectsController` : Gestion des matières
- `AbsencesController` : Gestion des absences

5.2 Modèles de Données

- `Teacher` : Informations sur les enseignants
- `Student` : Données des étudiants
- `Class` : Configuration des classes
- `Subject` : Définition des matières
- `Absence` : Enregistrement des absences
- `TeacherSubjectClass` : Relations entre enseignants, matières et classes

5.3 Entity Framework Core

- Configuration du contexte de base de données
- Migrations pour la gestion du schéma
- Relations entre les entités
- Requêtes optimisées avec LINQ

5.4 Sécurité et Performance

- Authentification basée sur les noms d'utilisateur
- Validation des données entrantes
- Gestion des erreurs globale
- Optimisation des requêtes

6 Implémentation Frontend

Le frontend du Student Absence Tracker est développé avec React et TypeScript, offrant une interface utilisateur moderne et responsive.

6.1 Structure des Composants

- **Layout** : Structure globale de l'application
 - Header : Barre de navigation supérieure
 - Sidebar : Menu latéral pour la navigation
 - Main Content : Zone principale de contenu
- **Pages** : Composants principaux
 - Dashboard : Vue d'ensemble
 - Teachers : Gestion des enseignants
 - Students : Gestion des étudiants
 - Classes : Gestion des classes
 - Subjects : Gestion des matières
 - MarkAbsences : Saisie des absences

6.2 Composants Réutilisables

- **Button** : Boutons stylisés avec états
- **Input** : Champs de saisie avec validation
- **Select** : Listes déroulantes
- **Table** : Tableaux de données
- **Modal** : Fenêtres modales
- **DatePicker** : Sélecteur de date

6.3 Gestion d'État

- Context API pour l'état global
- Hooks personnalisés pour la logique réutilisable
- État local avec useState pour les composants

6.4 Communication avec l'API

- Service API centralisé
- Gestion des requêtes HTTP avec Fetch
- Gestion des erreurs et chargement
- Types TypeScript pour la sécurité des données

7 Tests et Validation

7.1 Tests Backend

- Tests unitaires des contrôleurs
- Tests d'intégration de l'API
- Tests de la couche de données
- Validation des modèles

7.2 Tests Frontend

- Tests unitaires des composants React
- Tests d'intégration des formulaires
- Tests de navigation
- Validation des entrées utilisateur

7.3 Assurance Qualité

- Revue de code
- Analyse statique du code
- Documentation du code
- Standards de codage

7.4 Performance

- Tests de charge de l'API
- Optimisation des requêtes
- Temps de réponse du frontend
- Optimisation du bundle React

8 Interface Utilisateur

Le Student Absence Tracker propose une interface utilisateur intuitive et fonctionnelle, adaptée aux besoins des enseignants et administrateurs.

8.1 Page de Connexion

FIGURE 1 – Interface de connexion

La page de connexion offre une interface épurée permettant aux enseignants de s'authentifier avec leur nom d'utilisateur (prénom) et mot de passe (nom).

8.2 Tableau de Bord

FIGURE 2 – Tableau de bord principal

Le tableau de bord présente :

- Statistiques globales (nombre d'étudiants, classes, absences)
- Vue d'ensemble des absences récentes
- Accès rapide aux fonctionnalités principales

8.3 Gestion des Absences

- Interface de saisie des absences par classe et matière
- Sélection de la date et de la session (matin/après-midi)
- Liste des étudiants avec cases à cocher
- Historique des absences consultable

8.4 Interfaces Administratives

- Gestion des enseignants : ajout, modification, suppression
- Gestion des étudiants : inscription, affectation aux classes
- Configuration des classes et matières
- Attribution des matières aux enseignants

8.5 Responsive Design

L'interface s'adapte automatiquement aux différents appareils :

- Version desktop pour une utilisation administrative
- Version tablette pour une utilisation en classe
- Navigation simplifiée sur mobile

9 Conclusion

Le Student Absence Tracker représente une solution moderne et efficace pour la gestion des absences en milieu éducatif. Le projet démontre l'utilisation réussie des technologies .NET Core et React pour créer une application web performante et conviviale.

9.1 Réalisations

- Développement d'une API REST complète avec .NET Core
- Création d'une interface utilisateur intuitive avec React
- Implémentation d'un système de gestion des données robuste
- Mise en place d'une architecture évolutive et maintenable

9.2 Perspectives d'Évolution

Le projet pourrait être enrichi avec :

- Génération de rapports d'absences détaillés
- Notifications automatiques aux responsables
- Interface mobile native
- Intégration avec d'autres systèmes de gestion scolaire
- Analyse statistique avancée des données d'absence

9.3 Impact et Bénéfices

- Simplification du processus de suivi des absences
- Réduction des erreurs administratives
- Meilleure visibilité sur l'assiduité des étudiants
- Gain de temps significatif pour le personnel enseignant

Cette solution démontre comment les technologies modernes peuvent être utilisées pour améliorer significativement les processus éducatifs quotidiens, tout en offrant une expérience utilisateur optimale pour tous les acteurs concernés.