

# Dossier Projet – Application de gestion de cours particuliers

---

## 1. Introduction

### 1.1 Contexte du projet

Dans le cadre d'une demande réelle émanant d'un professeur particulier, le projet vise à développer une application web permettant de gérer de manière centralisée les cours particuliers, de la réservation au paiement en ligne. Actuellement, la gestion des cours, des inscriptions et des paiements se fait de manière dispersée (échanges par téléphone, messages, virements manuels), ce qui entraîne des pertes de temps, un risque d'erreur et une expérience utilisateur peu fluide. L'objectif est donc de concevoir un outil simple et intuitif permettant :

. Aux élèves de s'inscrire, réserver un créneau disponible, effectuer un paiement sécurisé et consulter l'historique de leurs cours.

. Au professeur, également administrateur de la plateforme, de gérer son planning, ses tarifs, les inscriptions et la communication avec ses élèves à partir d'un espace unique.

Ce projet s'inscrit dans une démarche de digitalisation des services éducatifs, en offrant un gain de temps, une meilleure traçabilité et une expérience utilisateur moderne.

### 1.2 Objectifs généraux

- Permettre aux élèves de réserver et payer leurs cours en ligne.
- Centraliser la gestion des plannings pour le professeur.
- Faciliter la communication entre professeur et élèves.
- Permettre aux élèves de télécharger leurs factures à la demande

### 1.3 Public cible

- **Professeur/Administrateur** : gestion complète des cours, des paiements et des élèves.
- **Élèves** : inscription, réservation, paiement, suivi des cours.

### 1.4 Problématique

La gestion actuelle des cours particuliers repose sur des échanges éparés par téléphone, SMS ou e-mail, ainsi que sur des paiements manuels par virement ou espèces. Ce fonctionnement présente plusieurs inconvénients :

- Manque de visibilité sur les disponibilités du professeur.
- Risque de double réservation ou d'oubli d'un cours.
- Suivi administratif et financier chronophage.
- Communication dispersée entre différents canaux.

Ces contraintes nuisent à la fluidité de l'organisation, augmentent le risque d'erreurs et peuvent impacter la satisfaction des élèves. Il est donc nécessaire de mettre en place une application centralisée, offrant une vision claire du planning, une réservation simple, un paiement sécurisé et un historique consultable à tout moment, afin d'optimiser le temps du professeur et de faciliter la vie des élèves.

---

## 2. Présentation générale

### 2.1 Description fonctionnelle

L'application se présente sous la forme d'un site web ergonomique et intuitif, composé d'une page d'accueil (landing page), d'un profil public du professeur, et d'un espace utilisateur sécurisé appelé dashboard, qui regroupe l'ensemble des fonctionnalités interactives.

#### **Page d'accueil (Landing page)**

La page d'accueil, accessible à tous, présente brièvement le principe de fonctionnement de l'application et ses avantages. Elle intègre un menu de navigation permettant :

- D'accéder au profil public du professeur.
- De se connecter ou de s'inscrire pour accéder au dashboard (partie privée de l'application).

#### **Espace utilisateur (Dashboard)**

Une fois connecté, l'élève accède au tableau de bord. Par défaut, le menu principal est affiché à gauche et permet de naviguer entre les différentes sections.

1. Gestion des informations personnelles.
2. Historique des réservations.
3. Communication avec le professeur.
4. Calendrier interactif
5. Commandes et paiement

#### **Le calendrier interactif**

Pour les élèves, ce calendrier permet de consulter les créneaux disponibles du professeur, avec indication des prix et promotions éventuelles, d'afficher le planning sous différentes vues : jour, semaine ou mois et de sélectionner un ou plusieurs créneaux pour pré-réservation.

Lorsqu'un créneau est choisi, il est réservé temporairement (15 minutes) le temps de finaliser le paiement. Pendant cette période, il n'apparaît plus comme disponible pour les autres utilisateurs.

Pour le professeur, ce calendrier est le cœur de l'application, il permet de visualiser les créneaux, réserves, libres ou en cours de réservation, mais également, il permet l'ajout des nouveaux créneaux ou l'édition des créneaux libres. tel que le changement des prix, l'ajout de promotion ou la suppression.

#### **Historique et réservations**

Un onglet dédié permet de consulter :

- Les réservations à venir.
- Les réservations passées.

### **Profil élève**

L'onglet Profil regroupe :

- Les informations personnelles (nom, prénom, coordonnées...).
- Les informations de formation (utiles au professeur pour préparer les cours).
- Les adresses (facturation, domicile...).
- Une description personnelle libre.
- Les liens vers les réseaux sociaux (LinkedIn, GitHub...).

Ces informations sont visibles par le professeur afin d'adapter son enseignement.

### **Contact**

L'onglet Contact permet d'envoyer un message directement au professeur pour :

- Demander un remboursement.
- Poser une question.
- Obtenir des renseignements divers.

### **Notifications**

La page par défaut du dashboard est la page Notifications, affichant :

- Toutes les notifications par ordre chronologique.
- Un système de filtrage (par notifications vues / non vues).
- Un résumé synthétique de l'activité de la semaine.

Des extras information sont affihces pour le professeur

### **Onglet Utilisateurs (professeur)**

Le professeur, également administrateur, dispose de fonctionnalités supplémentaires :

Onglet Utilisateurs qui permet de lister tous les élèves, rechercher un profil, consulter leurs informations.

### **Personnalisation visuelle**

En bas de l'application, il propose un mode sombre et un mode clair, que l'utilisateur peut sélectionner selon ses préférences. Le choix est conservé tant qu'il n'est pas modifié.

## **2.2 Cas d'usage principaux**

- Réserver un cours.
- Payer en ligne.
- Consulter l'historique.
- Envoyer un message au professeur.

2.3 Acteurs et rôles

Acteur	Rôle principal
Professeur	Administrer et donner les cours
Élève	Réserver et suivre les cours

3. Fonctionnalités détaillées

- 1. Inscription et authentification.
- 2. Réservation de créneaux disponibles.
- 3. Paiement sécurisé.
- 4. Consultation de l'historique.
- 5. Messagerie.
- 6. Gestion des tarifs et disponibilités.

4. Architecture technique

4.1 Technologies utilisées

- **Frontend** : Angular 19
- **Backend** : .NET 8.0
- **Base de données** : Postgres 15
- **Paiement** : Stripe
- **Hébergement** : VPS chez Hostinger

4.2 Schéma d'architecture

*(Insérer un diagramme)*

4.3 Structure de la base de données

*(Lister les tables principales et relations)*

5. Conception

- Diagrammes UML : cas d'utilisation, séquence, classes.
- Maquettes écran.
- Modèle de données.

6. Sécurité

- Authentification sécurisée (hash mot de passe, JWT).
- HTTPS obligatoire.
- Conformité RGPD.
- Protection contre les injections SQL.

---

## 7. Déploiement

- **Environnement de développement** : Windows, Visual Studio, Angular CLI.
  - **Environnement de production** : hébergement web + base SQL.
  - Procédure de mise en ligne.
- 

## 8. Tests et assurance qualité

- Tests unitaires.
  - Tests d'intégration.
  - Tests end-to-end.
  - Validation fonctionnelle.
- 

## 9. CI/CD

- Utilisation de GitHub Actions ou Azure DevOps.
  - Automatisation des builds et tests.
  - Déploiement automatique sur l'environnement de test.
- 

## 10. Maintenance et évolutivité

- Correctifs de bugs.
  - Ajout futur de nouvelles fonctionnalités.
  - Amélioration de l'expérience utilisateur.
- 

## 11. Gestion de projet

- Méthodologie : Agile/Scrum.
  - Planning.
  - Suivi des tâches.
- 

## 12. Conclusion

- Bilan.
  - Limites.
  - Perspectives.
- 

## 13. Annexes

- Captures d'écran.
- Documentation API.
- Schéma BDD.