

# ForsaTaalim

## 1. Introduction

### Nom du projet

**ForsaTaalim**

### Objectif du projet

Le projet **ForsaTaalim** a pour objectif de créer une plateforme éducative innovante qui met en relation des jeunes chercheurs d'emploi, souvent enseignants qualifiés, avec des parents à la recherche de professeurs pour leurs enfants. Cette plateforme vise à simplifier la recherche de cours particuliers et à rendre l'éducation plus accessible. En permettant aux utilisateurs de réserver des cours en ligne ou en présentiel, et en intégrant des outils de suivi des progrès, de gestion de paiements sécurisés, ainsi que des évaluations et des avis, \*ForsaTaalim\* cherche à répondre aux besoins croissants de flexibilité et d'accessibilité dans le domaine de l'enseignement privé.

### Technologies utilisées

Le projet utilise plusieurs technologies pour garantir une expérience fluide, sécurisée, et performante. Les principales technologies utilisées sont :

- **Backend** : Laravel (PHP), un framework robuste et sécurisé pour la gestion du backend.
- **Frontend** : ReactJS, un framework moderne pour construire des interfaces utilisateurs interactives et réactives.
- **Base de données** : MySQL, une base de données relationnelle puissante et fiable pour gérer les informations des utilisateurs, les cours, et les paiements.
- **Hébergement** : Cloud AWS ou DigitalOcean, offrant une infrastructure cloud sécurisée et scalable pour héberger la plateforme.
- **Sécurité** : SSL/TLS pour le chiffrement des données, authentification à deux facteurs, et protection contre les attaques DDoS.

## Problématique

Dans un contexte où la demande pour des cours particuliers et un enseignement personnalisé ne cesse de croître, plusieurs défis demeurent :

- **Accès difficile à des enseignants qualifiés** : De nombreux parents ont du mal à trouver des enseignants compétents et disponibles pour répondre aux besoins spécifiques de leurs enfants.
- **Manque de transparence** : Il n'existe pas toujours de mécanismes fiables pour évaluer les compétences des enseignants et la qualité de l'enseignement, ce qui crée de l'incertitude pour les parents.
- **Système de réservation et paiement complexe** : La réservation des cours et la gestion des paiements peuvent être complexes et peu sécurisées, rendant l'expérience utilisateur moins fluide.
- **Manque de flexibilité** : Les parents ont souvent des horaires stricts, et la recherche de solutions d'enseignement adaptées à leurs contraintes de temps est limitée.
- **Sécurité des données personnelles** : La protection des données personnelles et des informations de paiement est un enjeu majeur, nécessitant des solutions de sécurité robustes.

## Structure hiérarchique des acteurs

La **structure hiérarchique des acteurs** de la plateforme \*ForsaTaalim\* peut être décrite comme suit :

1. **Administrateur (Admin) - Rôle** : Supervise et gère l'ensemble de la plateforme. L'administrateur a accès à toutes les fonctionnalités de gestion, y compris la gestion des utilisateurs, des paiements, des évaluations et des rapports d'activité. - **Responsabilités** :

- Gérer les comptes utilisateurs (professeurs, étudiants, parents).
- Superviser les transactions et les paiements.
- Modérer et valider les évaluations.
- Générer des rapports de performance et de statut de la plateforme.
- Assurer la sécurité et la conformité de la plateforme.

2. **Professeur (Teacher) - Rôle** : Propose des cours aux étudiants. Le professeur crée et gère son profil, met à jour ses disponibilités et accepte ou refuse les réservations. - **Responsabilités** :

- Créer et mettre à jour son profil.
- Proposer des cours, y compris les matières, les horaires et les tarifs.

- Gérer ses réservations de cours et son emploi du temps.
- Recevoir les paiements pour les cours dispensés.
- Gérer les évaluations et retours des étudiants.

3. **Étudiant (Student) - Rôle :** Recherche et réserve des cours proposés par les professeurs. L'étudiant gère son propre profil et effectue les paiements nécessaires pour les cours réservés. - **Responsabilités :**

- Créer et gérer son profil personnel.
- Rechercher des professeurs en fonction des matières, des horaires et des prix.
- Réserver des cours et gérer son emploi du temps.
- Effectuer des paiements sécurisés en ligne.
- Laisser des évaluations sur les cours et les professeurs.

4. **Parent (Parent) - Rôle :** Rechercher des professeurs qualifiés pour leurs enfants. Les parents suivent les progrès de leurs enfants dans leurs études grâce à l'interaction avec les professeurs et la plateforme. - **Responsabilités :**

- Rechercher un professeur en fonction des besoins et des matières spécifiques de leurs enfants.
- Gérer les réservations de cours pour leurs enfants.
- Suivre les progrès de leurs enfants en consultation avec les professeurs.
- Suivre les paiements et gérer les transactions liées aux cours.

Chaque acteur a des responsabilités distinctes, mais ils interagissent ensemble pour faire fonctionner la plateforme de manière fluide et efficace. L'administrateur étant au sommet de cette hiérarchie, il a la responsabilité ultime de la gestion et de la surveillance de toutes les activités.

## 2. Vue fonctionnelle de la plateforme

### Besoins fonctionnels pour l'acteur

La plateforme répond à différents besoins fonctionnels pour chaque acteur :

### **Besoins fonctionnels de l'administration**

L'administration de la plateforme doit pouvoir :

- Gérer les comptes utilisateurs (professeurs, étudiants, parents).
- Superviser les paiements et les transactions.
- Mettre en place un système de gestion des cours et des horaires.
- Modérer les évaluations et les avis des utilisateurs.
- Générer des rapports sur les performances de la plateforme.

### **Besoins fonctionnels pour le professeur**

Les professeurs doivent pouvoir :

- Créer et gérer leur profil.
- Publier des annonces de cours avec des détails sur la matière, l'horaire et le tarif.
- Consulter les demandes de cours et accepter ou refuser les réservations.
- Gérer leur emploi du temps.
- Suivre les paiements et les évaluations des étudiants.

### **Besoins fonctionnels pour les étudiants**

Les étudiants doivent pouvoir :

- Créer et gérer leur profil.
- Rechercher et réserver des cours en fonction des matières et des horaires.
- Effectuer des paiements en ligne de manière sécurisée.
- Laisser des évaluations et des commentaires sur les professeurs.
- Suivre leurs progrès et recevoir des retours de leur professeur.

### **Besoins non fonctionnels de sécurité**

Les besoins non fonctionnels de sécurité pour la plateforme incluent :

- **Protection contre les attaques courantes :** Mise en place de pare-feu, protection DDoS et chiffrement des communications.
- **Sécurisation des données :** Chiffrement des données sensibles (paiements, informations personnelles) et utilisation de protocoles de sécurité comme SSL/TLS.

- **Monitoring et audits** : Surveillance en temps réel de l'activité de la plateforme pour détecter des comportements suspects, et audits réguliers pour assurer la conformité.
- **Résilience** : Mise en place de mécanismes de sauvegarde régulière, de redondance et de tolérance aux pannes pour garantir la continuité des services.

## Liste des Fonctions par Rôle

### Étudiant (Student)

#### 1. inscrireSeConnecter()

Permet à l'utilisateur de créer un compte ou de se connecter à la plateforme.

#### 2. rechercherProfesseur(criteres)

Permet de rechercher des tuteurs en fonction de critères personnalisés (matière, localisation, tarif...).

#### 3. consulterProfil(idTuteur)

Affiche les informations détaillées d'un tuteur (compétences, expérience, avis, etc.).

#### 4. reserverCours(idTuteur, date)

Permet aux étudiants de réserver un cours avec un tuteur.

#### 5. laisserAvis(idTuteur, note, commentaire)

L'étudiant peut donner un avis et une note après un cours.

#### 6. payerCours(idReservation, modePaielement)

Effectue le paiement sécurisé pour un cours réservé.

#### 7. suivreNotifications()

Permet de suivre les réservations et de recevoir des notifications sur l'état des cours.

#### 8. consulterCoursDisponibles()

Affiche la liste des cours actuellement disponibles.

### **9. ajouterPreferences(preferences)**

L'étudiant peut configurer ses préférences pour trouver des tuteurs correspondant à ses attentes.

### **10. consulterHistoriqueReservations()**

Affiche la liste des réservations passées avec leurs détails.

## **Professeur (Teacher)**

### **1. inscrireSeConnecter()**

Permet au professeur de s'inscrire ou de se connecter à la plateforme.

### **2. creerMettreAJourProfil()**

Le professeur peut créer ou mettre à jour son profil avec ses compétences, diplômes et expérience.

### **3. accepterRefuserReservations(idReservation)**

Gère les demandes de réservation en les acceptant ou refusant.

### **4. laisserAvis(idEtudiant, note, commentaire)**

Le professeur peut évaluer un étudiant après un cours.

### **5. gererDisponibilite(creneaux)**

Ajoute ou modifie les créneaux horaires de disponibilité du professeur.

### **6. recevoirPaiement(idReservation)**

Permet au professeur de recevoir le paiement pour les cours dispensés.

### **7. ajouterQualifications(certifications)**

Le professeur peut ajouter des diplômes et certifications supplémentaires à son profil.

### **8. gererRetoursExperience()**

Consulte et répond aux avis laissés par les étudiants.

### **9. organiserCoursGroupe(listeEtudiants)**

Le professeur peut organiser des cours en groupe avec plusieurs étudiants.

#### **10. proposerCoursEnLigneOuPresentiel(mode)**

Choix entre l'organisation de cours en ligne ou en présentiel.

### **Administrateur (Administrator)**

#### **1. gererUtilisateurs()**

Permet de gérer les comptes des étudiants et des professeurs.

#### **2. gererPaielements()**

Supervise et valide les paiements effectués sur la plateforme.

#### **3. surveillerAvis()**

Modère et surveille les avis laissés par les utilisateurs.

#### **4. consulterStatistiques()**

Affiche des statistiques sur l'utilisation de la plateforme et les réservations.

#### **5. gererCategoriesCours()**

Ajoute, modifie ou supprime des catégories de cours.

#### **6. approuverProfilsProfesseurs(idProfesseur)**

Examine et valide les profils des professeurs avant qu'ils ne puissent proposer des cours.

#### **7. genererRapports()**

Produit des rapports d'activité et de performance sur la plateforme.