



# CodeIguanas

## Cahier des charges

### Application HomeSkolar

## Spécifications fonctionnelles

### 1. Présentation du projet

HomeSkolar est une association qui met en relation des enfants en difficulté scolaire avec des tuteurs bénévoles. L'objectif principal est de permettre à tout élève, où qu'il soit, d'accéder à un soutien scolaire adapté à ses besoins.

L'application web doit faciliter les interactions entre les élèves et les tuteurs en proposant des fonctionnalités de gestion des comptes, de communication, de planification et de suivi des tâches.

### 2. Fonctionnalités principales

#### 2.1. Gestion des Comptes Utilisateurs

Les élèves et les tuteurs doivent pouvoir gérer leur compte personnel.

- Inscription via un formulaire (nom, prénom, email, mot de passe, statut élève/tuteur).
- Connexion sécurisée avec un email et un mot de passe.
- Réinitialisation de mot de passe en cas d'oubli.
- Mise à jour des informations personnelles.

#### 2.2. Communication entre Élèves et Tuteurs

Un système de messagerie interne permet aux utilisateurs de communiquer facilement.

- Envoi et réception de messages privés.
- Possibilité d'épingler des messages importants.
- Notifications des nouveaux messages non lus.

## 2.3. Planification des Rencontres

La gestion des rendez-vous entre élèves et tuteurs se fait via un calendrier interactif.

- Affichage des rendez-vous programmés.
- Prise de rendez-vous entre un élève et son tuteur.
- Modification et annulation des rendez-vous.
- Notifications pour rappeler les rendez-vous à venir.

## 2.4. Suivi des Tâches

Pour assurer un meilleur suivi des progrès des élèves, un système de gestion des tâches sera mis en place.

- Ajout automatique d'une liste de tâches à la fin de chaque session.
- Ajout manuel de tâches personnelles.
- Possibilité de marquer une tâche comme complétée.
- Notifications pour rappeler les tâches en attente.

# Spécifications techniques

Partie du projet	Technologie retenue	Alternatives envisagées	Justification du choix et rôle dans le projet
Front-end (interface utilisateur)	React.js	Vue.js, Angular	React.js est une technologie très populaire pour créer des interfaces modernes, rapides et interactives. Elle permet de concevoir des écrans intuitifs et agréables à utiliser pour les élèves et les tuteurs. Vue.js aurait pu être utilisé mais dispose d'une communauté plus réduite, tandis qu'Angular est plus complexe à mettre en œuvre rapidement.
Back-end (gestion des données et logique métier)	Django (Python)	Node.js (Express), Laravel (PHP)	Django est un framework Python robuste, sécurisé et facile à maintenir dans la durée. Nous l'avons choisi car nous utilisons principalement Python, ce qui facilitera grandement la réalisation du projet. Django permet de gérer efficacement les utilisateurs, les rendez-vous, les messages et les tâches avec une sécurité intégrée. Node.js aurait pu être envisagé mais nécessite plus de configuration initiale, tandis que Laravel implique un changement de langage vers PHP.
Base de données	PostgreSQL	MongoDB, MySQL	PostgreSQL est une base de données relationnelle fiable qui s'intègre parfaitement avec Django. Elle permet d'organiser clairement les données structurées nécessaires au projet (utilisateurs, rendez-vous, tâches). MongoDB aurait été adapté pour des données moins structurées, tandis que MySQL offre moins de flexibilité sur les requêtes complexes nécessaires ici.

Authentification utilisateurs	Authentification intégrée Django	Firebase Auth, OAuth2 (Auth0)	L'authentification intégrée à Django répond simplement aux besoins du client : inscription rapide, connexion sécurisée et gestion facile des comptes utilisateurs sans dépendance externe supplémentaire. Les autres solutions auraient ajouté une complexité inutile à ce stade du projet.
Messagerie instantanée	WebSockets (Django Channels)	API REST polling, Firebase Realtime Database	Les WebSockets permettent une communication instantanée entre élèves et tuteurs avec notifications immédiates des nouveaux messages ou messages épinglés importants. Cette solution répond précisément au besoin client d'une messagerie simple mais efficace. Les autres solutions sont soit moins performantes soit plus difficiles à maintenir dans le temps.
Calendrier / Planification	FullCalendar.js(intégré avec React)	Google Calendar API, Day.js	FullCalendar.js permet d'afficher facilement les rendez-vous entre élèves et tuteurs sur un calendrier interactif clair et simple à utiliser côté utilisateur. Cette solution répond directement au besoin client sans dépendre d'un service externe comme Google Calendar qui serait plus complexe techniquement.
Notifications internes	Django Celery + Redis	Cron Jobs simples, Firebase Cloud Messaging (FCM)	Django Celery associé à Redis permet de gérer efficacement les notifications internes : rappels de tâches ou messages non lus pour assurer un bon suivi par l'élève. Cette solution est simple à intégrer avec Django tout en étant fiable sur le long terme. Les alternatives sont soit trop limitées soit introduisent une dépendance externe non nécessaire au projet actuel.

# Diagramme de classes

