



CodeIguanas

Cahier des charges

Application HomeSkolar

Spécifications fonctionnelles

Objectifs de l'application

L'application HomeSkolar a pour objectif de permettre à des élèves en difficulté scolaire d'être accompagnés à distance par des tuteurs bénévoles.

Elle doit faciliter la mise en relation, la communication, la planification des rendez-vous et le suivi du travail entre élèves et tuteurs, dans un environnement sécurisé et intuitif.

Fonctionnalités principales

1. Gestion des comptes utilisateurs

- Création de compte : Inscription des élèves et des tuteurs via un formulaire en ligne.
- Connexion : Authentification sécurisée par identifiant et mot de passe.
- Gestion du profil : Modification des informations personnelles.
- Gestion des droits : Distinction entre les rôles "élève" et "tuteur bénévole".

2. Système de communication

- Messagerie interne : Échanges de messages entre élèves et tuteurs.
- Épinglage de messages : Marquer un message important.
- Notifications : Alerte pour tout message non lu.

3. Planification des rendez-vous

- Calendrier partagé : Affichage des rendez-vous.
- Prise de rendez-vous : Planification des créneaux d'accompagnement.
- Gestion des disponibilités : Déclaration des plages horaires du tuteur.

4. Suivi pédagogique et gestion des tâches

- Attribution de tâches : Liste de devoirs ou d'objectifs.
- Consultation des tâches : Espace personnel de l'élève.
- Création de mémos personnels : Notes ou rappels personnels.

Cohérence avec les objectifs

Toutes ces fonctionnalités répondent aux besoins de HomeSkolar :

- Fluidifier les échanges élèves/tuteurs.
- Organiser le suivi pédagogique hebdomadaire.
- Offrir une solution ergonomique et accessible aux utilisateurs.

Veille technologique

1. Front-end : React.js

Analyse : Bibliothèque JavaScript développée par Meta pour créer des interfaces utilisateur réactives.

Avantages :

- Immense communauté et forte popularité.
- Documentation riche : react.dev.
- Composants réutilisables, productivité accrue.
- Intégration naturelle avec les API REST.

Inconvénients :

- Courbe d'apprentissage initiale.
- Nécessité de respecter des bonnes pratiques.

Décision : Retenu pour sa maturité, communauté et ergonomie.

Source : [6 raisons d'aimer et d'utiliser ReactJS - ACSEO](#)

2. Back-end : Django (Python)

Analyse : Framework web open source basé sur Python, reconnu pour sa simplicité et sa rapidité de développement.

Avantages :

- Très riche en fonctionnalités intégrées (ORM, sécurité, authentification).
- Documentation exceptionnelle : docs.djangoproject.com.
- Grande communauté active.
- Excellente structure pour le développement rapide d'applications sécurisées.

Inconvénients :

- Moins performant que certains frameworks pour des applications à très forte charge.
- Courbe d'apprentissage pour maîtriser l'ensemble de l'écosystème Django.

Décision : Retenu pour sa robustesse, sa simplicité de mise en œuvre et son intégration fluide avec PostgreSQL.

Source :

- [Pourquoi utiliser Django pour du développement web ?](#)
- [Le parfait combo Django/Python pour développer une application web](#)
- [Django : Tout sur le framework de développement web en Python](#)

3. Base de données : PostgreSQL 17

Analyse : Système de gestion de base de données relationnelle (SGBDR) open source puissant et mature.

Avantages :

- Très forte intégrité des données.
- Fonctionnalités avancées : transactions, contraintes, indexation.
- Haute scalabilité et performance.
- Très bonne documentation et communauté active.

Inconvénients :

- Courbe d'apprentissage légèrement supérieure à celle d'une base NoSQL.
- Moins flexible que NoSQL pour des structures de données très variées.

Décision : Retenu pour sa fiabilité, sa robustesse et sa conformité aux standards des bases de données relationnelles.

Source : [PostgreSQL : Robuste, performant, stable et open-source](#)

Conclusion

Les choix s'inscrivent dans une stratégie :

- Simplicité avant complexité (PEP20).
- Technologies matures, documentées, avec forte communauté.
- Évolutivité garantie pour accompagner la croissance.

Ce socle technique est fiable, pérenne et offre un coût de maintenance maîtrisé.

Spécifications techniques

Cette partie décrit les technologies choisies pour le développement de l'application HomeSkolar. Elle explique leur rôle et en quoi elles répondent aux besoins exprimés par le client, notamment en matière de gestion des comptes, de messagerie, de calendrier et de suivi des tâches.

1. Front-end – React.js

React.js est une bibliothèque JavaScript moderne permettant de construire des interfaces utilisateur réactives et modulaires. Elle repose sur un système de composants réutilisables et un DOM virtuel qui optimise les performances d'affichage.

Pourquoi React.js ?

- Réactivité : mise à jour dynamique de l'interface sans recharger la page.
- Modulation : composants réutilisables pour structurer efficacement les vues (inscription, messagerie, tableau de bord).
- Communauté : forte adoption, documentation riche et nombreux plugins disponibles.
- Accessibilité : facilité d'intégration avec des bibliothèques UI comme Chakra UI pour créer une interface accessible.

Rôle dans le projet :

React.js permet d'offrir une expérience fluide aux utilisateurs (élèves et tuteurs), avec une navigation rapide, des mises à jour en temps réel du calendrier, des messages instantanés, et une interface claire pour la gestion des tâches.

2. Back-end – Django

Django est un framework web Python structuré et complet, qui inclut par défaut un ORM, un système d'authentification, un moteur d'admin, et une organisation modulaire des applications.

Pourquoi Django ?

- Rapidité de développement : structure 'batteries-included' avec beaucoup d'outils intégrés.
- Sécurité : protections natives contre les failles courantes (CSRF, XSS, injections SQL).
- Extensibilité : parfait pour une architecture modulaire avec des apps indépendantes (utilisateurs, séances, messages).
- Compatibilité avec Django REST Framework (DRF) pour exposer des API REST modernes.

Rôle dans le projet :

Django gère toute la logique métier de l'application : authentification, planification des séances, création et gestion des messages, des tâches, ainsi que l'interaction avec la base de données via l'ORM.

3. Base de données – PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle open-source reconnu pour sa robustesse et sa richesse fonctionnelle.

Pourquoi PostgreSQL ?

- Fiabilité : réplication, transactions ACID, gestion robuste des erreurs.
- JSONB : permet de stocker des données semi-structurées tout en profitant du SQL.
- Indexation : performances optimisées sur les requêtes complexes (notamment avec GIN/BTree).
- Communauté active et documentation complète.

Rôle dans le projet :

PostgreSQL stocke toutes les données persistantes (comptes, messages, séances, tâches). Son schéma relationnel permet de modéliser précisément les relations entre élèves, tuteurs et contenus.

Conclusion

Le trio React.js + Django + PostgreSQL forme une stack moderne, fiable et adaptée aux objectifs de HomeScolar. Il permet de répondre efficacement aux besoins du client tout en garantissant évolutivité, maintenance aisée et sécurité.

Diagramme de classes

