# Veille technologique – Application HomeSkolar

## Objectif

Proposer un socle technologique fiable, maintenable et simple, tout en garantissant la satisfaction des besoins métier et la scalabilité du projet.

## 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](https://www.acseo.fr/developpement/6-raisons-daimer-reactjs/)

## 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 ?](https://www.mobiskill.fr/blog-posts/pourquoi-utiliser-django-pour-du-developpement-web#:~:text=Django%20%3A%20ORM&text=Les%20d%C3%A9veloppeurs%20n'ont%20pas,des%20modifications%20minimales%20du%20code.)

## 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](https://www.d-booker.fr/content/72-postgresql)

## 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.