

# Cahier des Charges

## Application HomeSkolar

### 1. Spécifications Fonctionnelles

Les fonctionnalités principales de l'application HomeSkolar :

#### 1.1. Authentification

- **Inscription** : Les élèves et les tuteurs doivent pouvoir s'inscrire sur la plateforme en fournissant des informations personnelles (nom, email, mot de passe).
- **Connexion** : Les utilisateurs doivent pouvoir se connecter à leur compte en utilisant leur email et mot de passe.
- **Gestion du compte** : Les utilisateurs doivent pouvoir gérer leurs informations personnelles et modifier leur mot de passe.

#### 1.2. Communication

- **Messagerie** : Les élèves et les tuteurs doivent pouvoir échanger des messages via une interface de messagerie intégrée.
- **Épingler des messages** : Les utilisateurs doivent pouvoir épingler certains messages pour les retrouver facilement.
- **Notifications** : Les utilisateurs doivent être notifiés pour chaque message non lu.

#### 1.3. Rencontres Élève/Tuteur

- **Planification des rencontres** : Les élèves et les tuteurs doivent pouvoir planifier des rendez-vous via un calendrier intégré.
- **Affichage des événements** : Les événements et rendez-vous doivent être affichés sur une page de calendrier classique.

#### 1.4. Tâches

- **Liste de tâches** : À la fin de chaque rencontre, les élèves doivent recevoir une liste de tâches à réaliser pour la prochaine rencontre.

- **Notifications des tâches** : Les tâches doivent pouvoir être notifiées et validées dans l'application pour un meilleur suivi.
- **Création de tâches personnelles** : Les élèves et les tuteurs doivent pouvoir créer des tâches personnelles (notes, mémos)

## 2. Veille Technologique

### 2.1. Front-end

- **Framework : React.js**
  - **Justification** : React.js est une bibliothèque JavaScript populaire et largement utilisée pour la construction d'interfaces utilisateur interactives et dynamiques. Son approche déclarative et sa réactivité en font un choix idéal pour fournir une expérience utilisateur fluide et optimale.
  - **Avantages** :
    - Réactivité et performance élevées grâce à la virtual DOM.
    - Large écosystème de bibliothèques et composants disponibles.
    - Approche déclarative qui simplifie le développement.
  - **Inconvénients** :
    - Courbe d'apprentissage initiale pour les nouveaux développeurs.
    - Complexité accrue pour les applications très dynamiques.

- **Langages : HTML5, CSS3, JavaScript**

- **Justification :** HTML5, CSS3 et JavaScript sont des langages standards de l'industrie pour le développement web. Ils garantissent une compatibilité maximale et une expérience utilisateur cohérente sur différents navigateurs et appareils.

- **Avantages :**

- Compatibilité universelle avec tous les navigateurs modernes.
- Large communauté et ressources disponibles pour le support.
- Flexibilité pour créer des interfaces riches et interactives.

- **Inconvénients :**

- Peut nécessiter des frameworks supplémentaires pour gérer des applications complexes.
- HTML et CSS peuvent devenir difficiles à gérer dans des projets de grande envergure sans structure adéquate.

## 2.2. Back-end

- **Framework : Django**

- **Justification :** Django est un framework web Python complet, sécurisé et bien documenté. Il est idéal pour des applications nécessitant une gestion complexe des utilisateurs et des interactions. En utilisant Django avec MySQL, nous bénéficions de la robustesse et de la performance de MySQL combinées à la productivité de développement de Django.

- **Avantages :**

- Framework web complet avec de nombreuses fonctionnalités intégrées.
- Sécurité renforcée grâce à la gestion automatique des failles de sécurité.

- **Inconvénients :**

- Apprentissage initial complexe en raison de sa richesse en fonctionnalités.
- Plus adapté aux applications de taille moyenne à grande.

- **Base de données : MySQL**

- **Justification :** MySQL est un système de gestion de base de données pour sa fiabilité, sa performance et sa compatibilité avec de nombreux langages. MySQL offre une solution solide pour le stockage et la gestion des données de l'application HomeSkolar.
- **Avantages :**
  - Haute performance et fiabilité.
  - Large communauté et support commercial.
  - Compatibilité avec de nombreux langages de programmation.
- **Inconvénients :**
  - Moins de fonctionnalités avancées par rapport à certaines alternatives.
  - Gestion des transactions moins robuste dans certains scénarios complexes.

## 2.3. API

- **Architecture : RESTful API**

- **Justification :** Les API RESTful sont une norme de facto pour la conception d'API web. Elles offrent une architecture simple, standardisée et bien comprise pour la communication entre le frontend et le backend. En utilisant une API RESTful, nous assurons une interopérabilité facile avec différentes technologies et une maintenance simplifiée de l'API.
- **Avantages :**
  - Simplicité et standardisation.
  - Large adoption et compatibilité avec de nombreux clients et serveurs.
- **Inconvénients :**
  - Moins flexible pour des requêtes complexes par rapport à GraphQL.

### 3. Spécifications Techniques

Les spécifications techniques présenteront en détail comment chaque technologie sera mise en œuvre pour répondre aux besoins du projet HomeSkolar, en se basant sur la veille technologique réalisée.

- **Front-end :**
  - Utilisation de React.js pour construire des interfaces utilisateur réactives et dynamiques.
  - HTML5 et CSS3 pour structurer et styliser le contenu web.
  - JavaScript pour la logique et les interactions dynamiques.
  
- **Back-end :**
  - Développement du serveur avec Django, offrant un environnement sécurisé et robuste.
  - MySQL comme base de données principale pour stocker et gérer les informations des utilisateurs et autres données.
  
- **API :**
  - Mise en œuvre d'une API RESTful pour permettre la communication entre le frontend et le backend.

### Sources Pertinentes

- "React - A JavaScript library for building user interfaces" - Official React Documentation (<https://reactjs.org/>)
- "Why React.js is so popular and why you should learn it" - Medium.com
- "The benefits of using Django for web development" - DjangoProject.com
- "MySQL vs PostgreSQL: Which is Better?" - TowardsDataScience.com
- "Introduction to Django" - Official Django Documentation (<https://docs.djangoproject.com/en/stable/intro/>)

### 3. Diagramme de classes

