



CAHIER DES CHARGES APPLICATION HOMESKOLAR

1. Présentation d'ensemble du projet

1.1 Présentation de l'entreprise :

- HomeSkolar, association qui met en relation des enfants en difficulté scolaire avec des tuteurs bénévoles, nous à commandé une application web permettant la mise en contact des ces derniers
- Cette application facilitera la mise en relation des élèves et tuteurs.

1.2 Les objectifs du site :

Ce site aura pour but de créer une plateforme accessible de mise en contact et d'organisation :

- Permettre un accès simple au site (Gestion de compte et informations personnels)
- Décider du soutien scolaire le plus adapté aux élèves
- Permettre une communication accessible et efficace entre élèves et tuteurs.
- Planification des rencontres
- L'organisation et le suivis de l'élève via des tâches et ressources
- L'organisation via notes personnelles

1.3 La cible adressée par le site :

- Les élèves en difficultés scolaire
- Professeurs bénévoles

1.4 Périmètre du projet :

- Le site sera en Français
- Ce projet n'adaptera pas le site web en application mobile



2. Spécifications Fonctionnelles

2.1. Authentification :

- Inscription:
 - Tuteurs et élèves peuvent s'inscrire grâce à un mail et un mot de passe
 - Inscription via Google
 - Inscription via Facebook
 - Questionnaire pour l'orientation des élèves
 - Technologie pour la création et gestion de compte et BDD
- Connexion
 - tuteur et élève peuvent se connecter après inscription
 - Connexion via Google
 - Connexion via Facebook

2.2. Profil :

- Gestion des données personnelles
 - Les élèves pourront présenter leurs lacunes, et leurs demandes
 - Les tuteurs pourront donner leurs compétences et champs de connaissances
- Mise en relation
 - Un questionnaire sera demandé lors de l'inscription
 - Suite au profil, un tuteur sera assigné à un élève
- Photo de profil
 - Élèves et tuteur pourront uploader une photo

2.3. Chat :

- Création de fil de discussion
 - Elève ou tuteur pourront créer un fil de discussion
 - Les messages seront organisés comme un chat classique
 - Élèves et tuteurs pourront épingler des messages en tête de discussion
 - il pourront aussi les exporter dans la section mémos

2.4. Calendrier :

- Le tuteur pourra créer des rendez vous
- Les échéances des tâches y seront épinglés



2.5. Tâches

- Le tuteur pourra créer des tâches à réaliser pour l'élève
 - Ces tâches pourront être ajoutés au calendrier si une échéance est nécessaire
- Le tuteur pourra aussi y ajouter des liens "ressources" pour aider l'élève dans ses exercices
- L'élève pourra demander une validation de ses tâches effectuer
 - upload des exercices.

2.6. Notes personnelle :

- Élèves et tuteurs pourront se créer des notes personnelle dans cette section

2.7. Notifications

- Élèves et tuteurs recevront une notification et un mail lors de la mise en relation
- Ils recevront aussi une notification lors de message non lu, ou d'un nouveau RDV mis en place
- Lors d'une création de tâches ou demande de validation
- Ils pourront choisir d'être averti par mail ou SMS



3. Veille Technologique

Cette veille Technologique peut être suivie via Wakelet, une application web permettant le regroupement en “collections” de plusieurs sources pertinentes à cette veille.

→ <https://wakelet.com/wake/xSRn31tQCnWtGwRNNNl>

3.1 Framework Django

Sources :

- Documentation Django :
<https://docs.djangoproject.com/fr/5.0/>
- Suivis de l'évolution et des nouvelles utilisations via Reddit :
<https://www.reddit.com/r/django/>

3.2 Les websockets pour le système de messagerie

Sources :

- Documentation implémentation Django :
<https://websockets.readthedocs.io/en/stable/howto/django.html>
- Avantage des websockets dans la création d'un système de messagerie :
<https://appmaster.io/fr/blog/comment-websocket-peut-il-ameliorer-les-applications-de-chat-en-temps-reel>
- Suivis de l'évolution via Reddit :
<https://www.reddit.com/r/websocket/>

3.3 Authentification et gestions données utilisateur via Google et Facebook

Sources :

- Implémentation authentification Google avec django :
<https://www.mindbrowser.com/implementation-of-social-authentication-by-using-firebase-with-django-rest-framework/>
- Firebase authentication Google :
https://firebase.google.com/docs/auth/web/start?hl=fr#web-modular-api*
- Documentation Authentification Facebook :
<https://developers.facebook.com/docs/facebook-login/>
- Implémentation authentification Facebook avec django :
<https://medium.com/geekculture/django-social-authentication-sign-in-with-facebook-eb52c384e1d>



4. Spécifications techniques

4.1 Language

Python, associé au framework Django, offre de nombreux avantages pour le développement web :

- Sa syntaxe claire et concise facilite la lecture et la maintenance du code, ce qui accélère le processus de développement.
- Django fournit également une architecture solide et bien structurée, favorisant la création rapide d'applications web robustes et sécurisées.
- De plus, sa vaste communauté offre un support étendu ainsi qu'une multitude de packages et d'extensions prêts à l'emploi, permettant de répondre efficacement à divers besoins.

4.2 Front-End

- **Templates intégrés** : Django propose un système de templates intégré qui permet de générer dynamiquement du contenu HTML.
- **Gestion des URL** : Django facilite l'organisation des URLs, ce qui rend la navigation entre les différentes pages du site plus fluide et intuitive pour les utilisateurs.
- **Intégration avec des frameworks front end** : Bien que Django soit principalement un framework back end, il peut être associé à des frameworks front end comme React, Vue.js ou Angular pour créer des applications web modernes et interactives.

4.3 Back-End

- **ORM intégré** : Django propose un Object-Relational Mapping intégré qui simplifie l'interaction avec la base de données.
- **Administration automatique** : Django fournit une interface d'administration générée automatiquement et accessible pour une gestion simple des données.
- **Gestion des formulaires** : Django propose des outils pour la validation et le traitement des formulaires côté serveur, ce qui simplifie la collecte et la validation des données entrées par les utilisateurs.
- **Sécurité intégrée** : Django intègre des fonctionnalités de sécurité telles que la protection contre les attaques par injection SQL ainsi que la gestion de mots de passe sécurisés. Cela permet de développer des applications web robustes et sécurisées dès le départ.
- **Extensibilité et réutilisabilité** : Django favorise la réutilisabilité du code grâce à son architecture modulaire et à son système de packages.



4.4 Technologie supplémentaire

- **Websockets :**
 - Permettent une communication bidirectionnelle entre le client et le serveur, ce qui permet aux applications web d'échanger des données en temps réel sans avoir besoin de recharger la page.
 - Permettent des mises à jour instantanées de l'interface utilisateur sans besoin de rafraîchir la page
 - Offrent une manière efficace et élégante d'établir des communications en temps réel entre les clients et les serveurs, ce qui ouvre la voie à des applications web interactives et réactives.
- **Firebase authentication :**
 - Permet l'authentification(Inscription et connexion) Via les services google
 - Offre une authentification simple, sécurisée et transparente aux utilisateurs

5. Diagramme UML



