



CodeIguanas

Cahier des charges:

Site web HomeSkolar.

Introduction:

HomeSkolar, vise à créer une plateforme connectant des élèves en difficulté scolaire à des tuteurs bénévoles à distance, offrant ainsi un accès au soutien scolaire. Cette association ambitionne de faciliter la communication entre élèves et tuteurs, de planifier efficacement les rencontres éducatives, et d'assurer un suivi des tâches, tout en mettant l'accent sur une expérience utilisateur conviviale. À travers ce cahier des charges, nous détaillons les fonctionnalités clés, les spécifications techniques, et les objectifs de développement nécessaires à la réalisation de HomeSkolar.

1. Spécifications fonctionnelles

Le site permet aux élèves de s'inscrire, de se voir assigner un tuteur bénévole, de planifier des sessions de soutien avec communication en visioconférence, de gérer leurs tâches assignées, et de recevoir des notifications en temps réel. Les tuteurs peuvent s'inscrire, accepter/refuser les demandes de sessions, communiquer en visioconférence, gérer les tâches assignées, et voir les évaluations des élèves. Les visiteurs peuvent consulter des informations générales, comprendre le processus d'inscription, contacter le support, consulter des retours d'expérience, et s'inscrire en tant qu'élève ou tuteur directement depuis la page d'accueil.

Les cas d'utilisation détaillés en annexe du cahier des charges décrivent de manière exhaustive les interactions entre les utilisateurs et le système HomeSkolar. Ces scénarios mettent en lumière les fonctionnalités clés du site, de l'authentification à la planification des rencontres, fournissant ainsi une vision complète des expériences utilisateur prévues.

1.1 Authentification :

- Les utilisateurs (élèves et tuteurs bénévoles) doivent pouvoir s'inscrire, se connecter et gérer leur compte utilisateur.
- Les fonctionnalités de réinitialisation de mot de passe et de modification des données personnelles doivent être disponibles.

1.2 Communication :

- Un système de messagerie instantanée doit être mis en place pour permettre la communication entre élèves et tuteurs.
- Les utilisateurs peuvent épingler des messages pour un accès facile.
- Notifications pour les nouveaux messages non lus.

1.3 Rencontres élève/tuteur :

- Intégration d'un calendrier pour planifier les rencontres élève/tuteur.
- Possibilité d'ajouter, modifier et supprimer des événements sur le calendrier.
- les rencontres s'effectuent grâce à un outils de visioconférence intégré sur le site

1.4 Tâches :

- Les tâches doivent être générées à la fin de chaque rencontre et notifiées aux élèves.
- Fonctionnalité permettant aux utilisateurs de créer des tâches personnelles, comme des notes ou des mémos.

2. Veille technologique

2.1. Frontend :

- **Technologie Utilisée** : React

Justification : React est une bibliothèque JavaScript bien établie, maintenue par Facebook, qui offre une approche déclarative pour la création d'interfaces utilisateur réactives. Son architecture basée sur les composants facilite la modularité, améliorant ainsi la maintenance du code. La vaste communauté et l'écosystème riche de React garantissent un support solide et des opportunités d'innovation continues.

- **Alternative à React** : Vue.js

Justification : Vue.js est une bibliothèque JavaScript progressive et facile à intégrer, offrant une approche flexible pour la construction d'interfaces utilisateur réactives.

2.2 Backend :

- **Technologie Utilisée** : Python avec Flask

Justification : Python est choisi pour sa lisibilité, sa simplicité, et son adoption généralisée dans le domaine du développement web. Flask, en tant que micro-framework, offre une approche légère et flexible pour la création d'API RESTful, correspondant aux besoins spécifiques de HomeSkolar. La simplicité de Flask accélère le développement, tout en fournissant une structure modulaire et extensible.

- **Alternative à Flask (Python)** : Django (Python)

Justification : Django est un framework web robuste et complet pour Python, offrant une structure plus conventionnelle pour le développement rapide d'applications.

2.3 Base de Données:

- **Technologie Utilisée : MySQL (SQL)**

Justification : MySQL est une base de données relationnelle populaire avec une solide réputation en termes de performances et de fiabilité, offrant des fonctionnalités SQL complètes et une bonne intégration avec de nombreuses applications.

- **Alternative à MySQL (SQL): MongoDB (NoSQL) :**

Justification : MongoDB, une base de données NoSQL orientée documents, est privilégiée pour sa flexibilité dans le stockage de données non structurées. Son modèle NoSQL s'aligne bien avec la nature évolutive et variée des données de HomeSkolar, permettant une évolution libre de la structure des données en fonction des besoins du projet. La prise en charge native des données JSON simplifie la manipulation des données complexes.

3. Spécifications techniques

Le choix des technologies pour le développement du site web HomeSkolar a été effectué pour répondre aux besoins spécifiques du client. En utilisant un mélange technologique équilibré, le site combine des technologies frontend telles que HTML, CSS, et React pour assurer une interface utilisateur réactive et conviviale. Côté backend, l'utilisation de Python avec Flask et la bibliothèque Flask-SocketIO permet une gestion efficace de l'authentification, de la communication en temps réel, et de la planification des rencontres. La base de données MongoDB, une base de données NoSQL, offre une flexibilité optimale pour stocker et gérer des données non structurées, adaptée aux besoins évolutifs de HomeSkolar. L'intégration de la technologie WebRTC pour la visioconférence assure une expérience immersive lors des rencontres élève/tuteur. En combinant ces technologies, HomeSkolar offre une plateforme robuste, sécurisée et évolutive, garantissant une expérience d'apprentissage en ligne optimale pour les élèves en difficulté scolaire et leurs tuteurs bénévoles.

3.1 Authentification (Frontend et Backend) :

- Frontend : HTML, CSS, JavaScript (React pour une gestion de l'interface utilisateur réactive).
- Backend : Python avec Flask, Flask-Security pour la gestion de l'authentification.

3.2 Attribution Automatique de Tuteurs (Backend) :

- Python (Flask) : Flask sera utilisé pour développer la logique back-end du système, offrant une approche légère et flexible pour l'intégration de l'algorithme.
- Base de Données (SQL - MySQL) : Stockage des informations sur les tuteurs, les élèves, et les critères d'attribution pour une gestion efficace des données.

3.3 Communication (Messagerie) :

- Frontend : HTML, CSS, JavaScript (React pour la gestion de l'interface utilisateur).
- Backend : Python avec Flask-SocketIO pour la communication en temps réel.

3.4 Rencontres Élève/Tuteur (Calendrier) :

- Frontend : HTML, CSS, JavaScript (React pour une expérience interactive).
- Backend : Python avec Flask pour la gestion des événements et des rendez-vous.

3.5 Tâches (Gestion des Tâches) :

- Frontend : HTML, CSS, JavaScript (React pour la gestion de l'interface utilisateur).
- Backend : Python avec Flask pour la gestion des tâches.

3.6 Stockage de Données (Base de Données) :

- Base de Données : MySQL pour ses performances éprouvées, sa fiabilité, et son intégration transparente avec une variété d'applications.

3.7 Intégration de Visioconférence :

- WebRTC : Pour la communication vidéo en temps réel, pouvant être intégré dans le frontend avec JavaScript et Flask-SocketIO pour la gestion des connexions.

3.8 Sécurité :

- Flask-Security : Pour des fonctionnalités de sécurité dans le backend Flask, notamment la gestion de l'authentification.

3.9 Interface Utilisateur Réactive :

- React : Pour la construction d'une interface utilisateur réactive et dynamique.

3.10 Avantages et inconvénients:

Avantages :

- **Utilisation de Python** : Python est connu pour sa lisibilité, ce qui peut faciliter le développement et la maintenance du backend. Flask, étant un micro-framework, offre une simplicité appréciable.
- **Flexibilité avec MongoDB** : L'utilisation de MongoDB offre une flexibilité dans le stockage de données JSON, adapté aux besoins changeants du projet.
- **Richesse de l'écosystème Python** : Python dispose d'un écosystème riche en bibliothèques et frameworks, ce qui peut être bénéfique pour des tâches spécifiques.

Inconvénients :

- **Courbe d'Apprentissage de Flask** : Si l'équipe n'est pas familière avec Flask, une courbe d'apprentissage peut être nécessaire, bien que Flask soit généralement considéré comme facile à apprendre.
- **Performance** : Bien que Python soit performant, il peut être moins rapide que Node.js pour les opérations non bloquantes en raison de son modèle d'exécution bloquant.

4. Diagramme de classes

