



# CodeIguanas

## Cahier des charges

### Application HomeSkolar

## Spécifications Fonctionnelles

### 1. Authentification

- ⑩ Les users peuvent s'inscrire en renseignant leur nom, leur adresse e-mail et un mot de passe.
- ⑩ Lors de l'inscription, ils ont le choix entre un compte "élève" ou "tuteur bénévole".
- ⑩ La connexion se fait via l'adresse e-mail et le mot de passe.
- ⑩ Chaque utilisateur ou utilisatrice peut modifier ses informations personnelles ainsi que son mot de passe.
- ⑩ Dès leur inscription, chaque élève est automatiquement associé à un tuteur.

### 2. Communication

- ⑩ Les utilisateurs et utilisatrices ont la possibilité d'envoyer, recevoir et archiver des messages.
- ⑩ Une notification est alors envoyée à chaque réception d'un nouveau message.
- ⑩ Les élèves peuvent épingler certains messages pour y accéder plus facilement.

### 3. Organisation des rendez-vous

- ⑩ Les élèves ainsi que leurs tuteurs peuvent planifier leurs rencontres directement via l'application.
- ⑩ Un calendrier intégré permet d'afficher les rendez-vous à venir.

### 4. Gestion des tâches

- ⑩ À la fin de chaque session, une liste de tâches est attribuée aux élèves en vue du prochain rendez-vous.
- ⑩ Les utilisateurs peuvent également créer des tâches personnelles, telles que des notes ou des mémos.
- ⑩ Les élèves reçoivent une notification pour chaque nouvelle tâche ajoutée.

# Spécifications Techniques

## Backend (Serveur Web et Base de Données)

Le site repose sur **Flask**, un framework web léger et flexible, et **MySQL** pour le stockage des données. Flask gère les requêtes HTTP, organise les routes et structure les vues, tandis que MySQL assure une gestion sécurisée et performante des informations via SQLAlchemy.

## Frontend (Interface Utilisateur)

L'interface est construite avec **Jinja**, le moteur de templates de Flask. Il permet d'afficher dynamiquement les données et de créer une navigation fluide et intuitive.

## Communication en Temps Réel

Pour faciliter les échanges entre élèves et tuteurs, **Flask-SocketIO** est utilisé afin de gérer les WebSockets. Cela permet d'avoir une messagerie instantanée fluide et réactive.

## Authentification

La gestion des comptes est assurée par **Flask-Login**, qui prend en charge l'inscription, la connexion et la gestion des sessions pour garantir un accès sécurisé à la plateforme.

## Calendrier

Un système de gestion de calendrier est intégré via **FullCalendar** couplé à Flask. Il permet aux utilisateurs de planifier leurs rendez-vous et de suivre les événements importants.

## Notifications

Pour tenir les utilisateurs informés des événements clés, **Flask-Flash** et **WebSockets** sont utilisés afin d'afficher des messages instantanés ou d'envoyer des alertes par e-mail selon les actions effectuées sur la plateforme.

# Diagramme de Classe

outil utilisé : <https://www.plantuml.com/>

