

Headless CMS - Rendu Airtable

Contexte

Vous allez développer un outil de gestion de clients en utilisant Airtable comme back-end (Headless CMS) pour gérer et visualiser les données des clients. Le but est de créer un système simple mais fonctionnel avec deux interfaces web et une automatisation des processus dans Airtable.

Objectifs

Le client devra avoir 6 champs : Prénom (firstname), Nom (lastname), Email (email), Numéro de Téléphone (phoneNumber), Notes (notes), Date de soumission (createdAt) et Statut (status) avec pour choix "Not contacted", "Contacted" et "Contact in future"

1. Page Admin

- Créer une page permettant d'afficher la liste des clients
- Intégrer des fonctionnalités pour modifier et supprimer des clients depuis cette interface
 - Pour la modification, il sera possible de modifier seulement les champs *Notes* et *Statut*

2. Page Formulaire Client

- Créer un formulaire permettant aux clients d'envoyer leurs informations (nom, prénom, email, numéro de téléphone).
- Lors de la soumission du formulaire, une automatisation devra se déclencher dans Airtable :
 - a. Envoi d'un email de confirmation au client pour l'informer que sa demande a bien été reçu
- Le statut du client devra être mis sur 'Not Contacted' au moment de sa création

3. Statistiques Clients

- Utiliser les fonctionnalités d'Airtable pour créer une interface de visualisation sous forme de graphiques. Ces graphiques devront afficher des statistiques sur les clients (Nombre total, Nombre en fonction du statut, etc ...).

Livrables attendus

- Une application React réalisé en Typescript, correctement typé et structuré.
- Deux pages web :
 - **Page admin** avec gestion des clients (lister, modifier, supprimer).
 - **Formulaire client** avec intégration de l'automatisation Airtable.
- Interface Airtable avec les graphiques pour la visualisation des statistiques des clients.
- Un style correct. Possibilité d'utiliser une librairie externe pour gérer le style (ex: Bootstrap, Tailwind, MUI, etc ...)

Critères d'évaluation

- Fonctionnalité complète des deux pages web. (8 points)
- Bonne intégration d'Airtable en tant que back-end. (5 points)
- Automatisations Airtable correctement configurées et déclenchées. (3 points)
- Qualité de l'interface de visualisation des statistiques. (2 points)
- Propreté et organisation du code (clarté, structure, types). (2 points)
- **Bonus : Style de l'application (1 point)**

Rendu

Le rendu du code se fera sur un repository Github dont il faudra m'envoyer le lien. Pour Airtable, il faudra me partager votre base en invitant l'adresse mail: gillettheo1@gmail.com

Deadline

Lundi 25 Novembre à 23h59

Documentation

- Documentation Airtable : <https://airtable.com/developers/web/api/introduction>
- Repository Github de référence : <https://github.com/theo-code33/airtable-courses>