

PRÉSENTATION DU CONTEXTE DU CAHIER DES CHARGES

1. Présentation du client et son contexte

Le client est un entrepreneur qui souhaite lancer sa startup digitale.

Il a trouvé dans son entourage un besoin non comblé et il souhaite être le premier à satisfaire ce besoin chez ses futurs clients.

Son marché est celui des salons d'esthétique. Il s'est rendu compte qu'il est extrêmement difficile d'accéder à des données à jour concernant le chiffre d'affaires des salons de beauté en France.

Le client n'a aucune infrastructure informatique. Il est prêt à payer pour un serveur privé virtuel (VPS) ainsi que pour l'achat du nom de domaine.

2. Expression du besoin

Le client souhaite que vous développiez la partie API *back-end* d'une application web qui permettra à un salon de beauté d'indiquer son chiffre d'affaires plus ou moins précisément, et d'avoir accès en échange à son positionnement parmi ses concurrents.

Le gérant du salon de beauté doit pouvoir se connecter à un espace personnel.

Les données doivent être reportées automatiquement.

2.1. Documentation de l'API

Vous intervenez sur tout le développement *back-end*, de la conceptualisation à la réalisation.

L'application doit comporter les fonctionnalités suivantes:

- ✓ Un système d'authentification JWT et d'inscription (*email + password*) avec confirmation d'email

→ Livrables:

- un *endpoint* "register" qui gère l'ajout d'un nouvel utilisateur en BDD
- un *endpoint* "login" qui authentifie l'utilisateur en renvoyant un *token* qui sera ré-utilisé pour authentifier chaque requête suivante
- un email envoyé lors de l'inscription - (BONUS: Plusieurs tentatives de connexion erronées mènent à l'envoi d'un mail de reset *password*)

- ✓ un espace personnel (profil du salon + accès à l'historique de saisie des mois passés + saisie du mois précédent)

→ Livrables:

- un *endpoint* "profil" qui gère l'accès et la modification des données du profil authentifié uniquement
- un *endpoint* "historique" qui permet l'accès aux CA saisis précédemment
- un *endpoint* "nouvelle saisie" qui gère la saisie du CA du mois précédent

- ✓ Un système de rappel par mail pour la saisie du mois passé

→ Livrable: Tâche CRON (ou autre scheduler) qui envoie un rappel par mail aux utilisateurs n'ayant pas saisi de CA pour le mois précédent (à partir du 5 du mois par exemple)

- ✓ Une base de données SQL

→ Livrable: une BDD SQL sur laquelle s'appuie l'API, les schémas sont à la discrétion des élèves avec une contrainte : pas de redondance d'information

- ✓ Mise à jour des statistiques du marché après chaque saisie (CA moyen France, CA moyen / Régions, CA moyen / Départements)

→ Livrable: les statistiques doivent être recalculées après chaque appel au *endpoint* "nouvelle saisie" uniquement pour la région et le département en question (Ex : si la saisie correspond à l'IDF, ne pas recalculer la moyenne de la Bretagne).

Le *back-end* est une API Rest, il doit permettre l'utilisation de n'importe quel *framework* front.

Le nom des *endpoints* ainsi que des fonctions n'est pas imposé mais doit être explicite.

Le code devra être documenté pour faciliter la maintenance par d'autres développeurs.
(des *docstrings* au format recommandé documentent chaque fonction)

Un fichier README.md devra présenter l'API avec des exemples d'utilisation.

Les mots de passe des utilisateurs doivent être stockés chiffrés.

Un utilisateur connecté n'a accès qu'à ses données.

Un utilisateur non connecté n'a accès qu'à la page d'accueil.

Vous utiliserez un outil de versioning collaboratif au choix.

Le client n'a aucune donnée, vous développerez un script de génération de fausses données.

2.2. Compléments



Mot de passe: Au moins 8 caractères, une majuscule, un chiffre et un caractère spécial. La réponse de l'api devra être différente en fonction du type de caractère manquant lors de l'enregistrement d'un nouvel utilisateur).

Champs du profil:

- Nom du salon
- Adresse du salon
- Date d'ouverture
- Nombre d'employés équivalent temps plein
- Nom du représentant/gérant
- Prénom du représentant/gérant

-- Fin du document --