**ÉVALUATION NODE JS**

**USE CASE:**

Au sein de votre agence web, vous êtes en charge de la création d’une petite application de gestion de projet qui permet au sein de votre entreprise de:

- créer des projets,

- créer des salariés,

- créer des clients,

- apposer des notes sur un projet,

- affecter des salariés à un projet,

- affecter un client sur un projet

**Un projet est caractérisé par :**

- Son nom,

- Une description

- Sa date de début,

- Sa date de fin,

- Un client,

- Un ou plusieurs salariés peuvent être affecté au projet,

- Le montant total du projet,

- son statut: annulé, terminé, en cours de réalisation, prospect

**Un salarié est caractérisé par :**

- Nom

- Prénom,

- username

- Date de naissance

- Adresse: rue + villle +code postal

- Téléphone,

- Mail

- poste: chef de projet, développeur, commercial, intégrateur

**Un client est caractérisé par:**

* + Nom de l’entreprise
  + Adresse: rue + ville + code postal
  + Contact référent: nom + prénom+ téléphone + mail
  + Secteur d’activité

**Arborescence de l’application :**

- **Home page:** Donne des stats globales sur l’activité de l’entreprise:

- Nombre de salariés dans l’agence,

- Nombre de projets annulés, terminés, en cours de réalisation, et en prospection

- Nombre de clients,

- Chiffre d’affaires total (somme des montants des projets terminés)

- **Page tous les Projets :** liste de tous les projets sous la forme d’un tableau avec call to action :voir, modifier, supprimer + bouton de création d’un projet

**- Page détail d’un projet:** vue d’ensemble de toutes les informations + call to action (supprimer, modifier), possibilité d’ajouter des commentaires

**- Page modifier un projet :** formulaire qui permet de modifier toutes les informations d’un projet.

- **Page création d’un projet:** formulaire de création d’un projet

**- Page tous les clients:** listes de tous les clients dans un tableau avec call to action ( voir modifier, supprimer) + bouton de création d’un client

-  **Page détail d’un client:** vue d’ensemble de toutes les informations + call to action (supprimer, modifier) + nom du projet sur lequel il est affecté

**- Page modifier un client :** formulaire qui permet de modifier toutes les informations signalétiques

**- Page création d’un client:** formulaire de création d’un projet

**- Page tous les salariés:** liste de tous les salariés sous la forme d’un tableau avec call to action :voir, modifier, supprimer + bouton de création d’un salarié

-  **Page détail d’un salariés:** vue d’ensemble de toutes les informations + call to action (supprimer, modifier) (en bonus les projets sur lesquels il est affecté)

**- Page modifier un salarié :** formulaire qui permet de modifier toutes les informations signalétiques

- **Page création d’un salarié:** formulaire de création d’un projet

**Merci de respecter les points suivants:**

- Un code claire, bien indenté et commenté

- Lorsque vous effectuez des actions de mise à jour ou suppression en front, de les soumettre dans des popups / modals

- Essayer au maximum que vos pages soient propres en front, penser à utiliser un framework css

- Utiliser des jointures entre les collections pour par exemple affecter un client à un projet, affecter des salariés à un projet etc… C’est une partie que nous n’avons pas vu en cours qui reste relativement simple à faire.

- Utiliser git : avec un git ignore

- Ne pas copier le code d’un autre, je le verrais tout de suite !!

**Technologies à utiliser:**

**- Base de données MangoDB (sur Mlab)**

**- Api Node js avec express JS + mongoose**

**- Front sur Angular**

**Ressources :**

- <https://expressjs.com/fr/>

- **<https://mongoosejs.com/>**

- <https://angular.io/>

**- <https://help.github.com/>**

- Jointures : **<https://mongoosejs.com/docs/populate.html>**

**Délais :**

Deadline au 24/10. Possibilité de faire ce projet par groupe de 2.

**Rendu : sur un repo Git, (penser à utiliser gitignore).**

Bien mentionner dans un fichier readme votre nom et prénom

**Bonus:**

**-** Créer une page de login et gérer les authentifications (au moins 1)

- Faire des contrôles sur formulaires (ex: utilisateurs existants, nom de projet qui existe déjà, format de saisi etc…).

- Ajout de toasters ou alerts validation ou erreur à la soumission d’un formulaire

- L’utilisateur peut uploader / downloader des documents sur un projet.