Contact Pro Manager

Explication

Afin de stocker vos contacts professionnels vous aller devoir réaliser une application Web qui permettra de voir les contacts, d'ajouter un contact et de voir la fiche d'un contact sélectionné à partir de la liste.

L'accès à cette application se fera à partir d'une page de connexion. Seul un utilisateur authentifié avec son email et son mot de passe pourra accéder aux pages permettant de gérer les contacts.

L'application fournira également une API pour manipuler les données.

Énoncé

Chaque user aura les informations suivantes :

- Nom
- Prénom
- Adresse mail
- Mot de passe
- Tasks

Chaque contact aura les informations suivantes :

- Nom
- Prénom
- Société
- Adresse
- Tel
- Adresse mail
- Secteur (Industrie, Informatique, Santé, Education, ...)
- User

Il faudra structurer la base de données MongoDB avec ces informations.

Vous pouvez vous baser sur ce tuto https://www.bezkoder.com/mongoose-one-to-many-relationship/ afin de mettre en place la relation entre user et contact.

On structurera l'application avec un dossier views dans lequel on stockera toutes les vues à venir, un dossier routes ou on aura toutes les routes nécessaires pour faire fonctionner l'application. Afin d'améliorer l'architecture de l'application, vous devez séparer la déclaration des routes, des traitements nécessaires pour construire les pages dans des contrôleurs. Ces contrôleurs seront seront stocké dans un dossier controllers. On va également séparer les manipulations des données dans avec des services qui seront stockées dans un dossier services.

L'application possédera une API pour pouvoir manipuler les contacts et les utilisateurs sans passer par l'interface utilisateur.

	CONNEXION	
erreurs		
Nom		
Nom		
Prénom		
Prénom		
	CONNEXION	

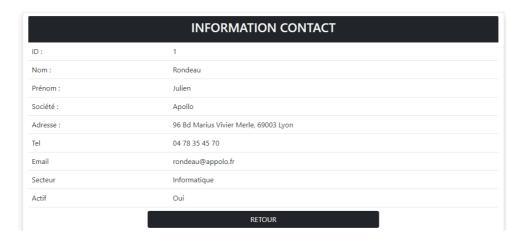
L'application possèdera un formulaire de connexion permettant à un utilisateur de se connecter et d'accéder à la page d'accueil affichant la liste des contacts. L'utilisateur connecté sera enregistré dans la session. On vérifiera si un utilisateur est enregistré dans la session avant d'autoriser l'accès à une page de l'application, dans le cas contraire il sera redirigé sur la page de connexion.

Vous pouvez créer un <u>middleware</u> afin d'effectuer cette vérification et de rediriger sur la page de connexion si l'utilisateur n'est pas connecté.

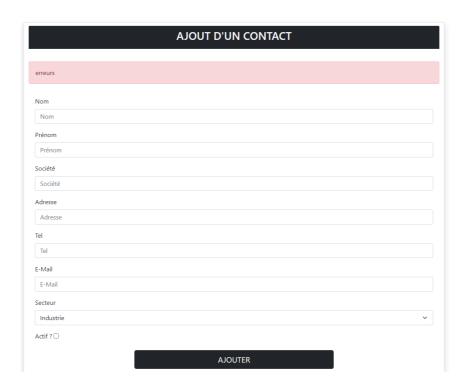


L'application possédera une page principale qui affichera la liste des contacts contenus dans la base de données et permettra l'ajout d'un contact, de voir un contact, de modifier un contact et de supprimer un contact.

Lors de la suppression d'un contact l'application demandera à l'utilisateur de confirmer la suppression.



L'application possèdera une page qui permettra de voir les informations d'un contact.



L'application possédera un formulaire permettant l'ajout d'un nouveau contact. La vue enverra les informations au processus principal qui ajoutera le contact dans la base de données et redirigera sur la page d'accueil pour afficher la liste des contacts. Un message d'erreurs sera affiché pour chaque champ qui n'est pas renseigné.

L'application possédera un formulaire permettant la modification d'un contact. La vue enverra les informations au processus principal qui modifiera le contact dans la base de données et redirigera sur la page d'accueil pour afficher la liste des contacts. Un message d'erreurs sera affiché pour chaque champ qui n'est pas renseigné. A noter que l'action de ce formulaire devra contenir l'id du contact à modifier dans l'url.