1. Présentation

Application Vérification Information Entreprise (VIE), les images qui suive peuvent changer au niveau de la présentation mais le principe de l’application reste le même

Lien du projet (repos git) : <http://p-m132:8080/tfs/TFS2017Collection/_git/VerifEntreprise>

Le projet a été réalisé sur Visual studio 2022 il est donc conseillé de l’utiliser.

Lors du lancement, vous arriver sur une page vous demandant de vous connecter.

Une image contenant texte, ligne, capture d’écran

Le contenu généré par l’IA peut être incorrect.

Après vous être connecter, vous arrivez sur une nouvelle page avec une barre de recherche, sur celle-ci plusieurs options sont possibles soit vous écrivez un Siret d’entreprise (cas 1) soit vous entrez le nom d’une entreprise (cas 2)

Une image contenant texte, capture d’écran, Police, blanc

Le contenu généré par l’IA peut être incorrect.

Cas 1 :

Sur la nouvelle page vous avez la possibilité de choisir l’année, l’API parmi 7 chacune est différentes et possède une présentation propre. Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Cas 2 :

Si vous entrez le nom d’une entreprise vous verez apparaître toutes les entreprise avec ce nom, il y a plusieurs pages et 10 entreprises par page. Un système de tri géographique est aussi proposé. De plus vous pouvez cliquez sur « Voir les détails » pour arriver dans le premier cas.

Une image contenant texte, capture d’écran, logiciel, Page web

Le contenu généré par l’IA peut être incorrect.

Attention : Il ne faut pas appuyer sur les boutons trop vite cela peut provoquer une surcharge de rêquête de l’API. Si cela arrive attendez et refaîte l’action si nécessaire.

1. Fonctionnement du code

Le code utilise la méthode MVC, les langages utilisés sont le C#, le Javascript, l’HTML et le CSS.

Exemple de Controller :

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Le contenu généré par l’IA peut être incorrect.

Le Controller permet d’envoyer une requête GET à l’API qui demande une authentification avec un Bearer Token (caché ici). La réponse JSON est lue et désérialisé dans un objet .NET, on remplit un model avec les données et on retourne la vue partielle correspondante.

Tous les Controllers suivent le même principe.

Exemple de Model :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Exemple de Vue :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

La vue appelle une fonction javascript sur le Json qui le met en forme

1. Modification du code

Dans cette partie on va voir comment modifier le code selon des cas précis.

1. Erreur Mauvaise valeur afficher sur le site

Dans cette partie nous prendrons pour exemple Liasse fiscale parce qu’elle le sujet à erreur. En effet dans Liasse fiscale les valeurs n’apparaissent pas toutes dans le Json ce qui pose un problème au moment de remplir le tableau.

Pour l’exemple imaginons que nous voulons ajouter la valeur « Concession, brevet et droits similaires, brut » qui est associé au code AF. Une image contenant texte, capture d’écran, ligne, nombre

Le contenu généré par l’IA peut être incorrect.

Lorsque l’on appelle l’API, on reçoit un Json de la forme suivante :

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

On remarque plusieurs points : on a le numéro de l’annexe sur laquelle les valeurs se trouvent (rouge). Puis il y a un tableau nommé « donnees » dont la première valeur est la valeur qui nous intéresse. Chaque valeur possède 4 propriétées intéressantes. Le code\_nref, la valeur de la cellule, son intitulé et son code.

Une image contenant texte, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

Pour modifier ou ajouter une cellule le code fonctionne de la manière suivante : On récupère la déclaration dont on a besoin (rouge), ici c’est la 2050, dans la déclaration il y a plusieurs valeurs toutes regroupées dans un tableau, on récupère donc ce tableau (jaune).

Ensuite nous avons deux fonctions qui sont très semblable, elles ont pour but de récupérer la valeur grâce à l’intitulé ou le code de la cellule (vert).

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Nous avons ensuite une partie où on construit le tableau et faisons de la mise en forme. Pour modifier une valeur il faut juste modifier l’intérieur de la fonction valeurs avec le bon intitulé (rouge).

Voici le rendu final sur le l’application.

Une image contenant texte, capture d’écran, Police, nombre

Le contenu généré par l’IA peut être incorrect.

1. Ajout d’une API

Pour ajouter une nouvelle API, il faut ajouter un modèle, une vue et un controller (plus un fichier javascript pour la mise en forme) qui sont similaire aux autres sauf quelque valeur à changer. Et il faut modifier mapage.cshtml.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

En effet juste après cette portion de code il faut rajouter la même chose en changeant le type et l’intérieur du fetch en fonction de votre controller.

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Ne pas oublier de rajouter un bouton.