**Fiche d'Exercices : Angular - Standalone Components et Directives (CRM Fournisseurs)**

**Objectifs :**

1. Créer un composant standalone et le charger en lazy loading via le système de routage d’Angular.
2. Utiliser les directives natives d’Angular, notamment \*ngFor, \*ngIf, et [class] pour manipuler l’affichage dynamique des données.
3. Construire une application de gestion des fournisseurs pour un CRM.

**Exercice 1 : Créer un Composant Standalone en Lazy Loading**

1. Utilisez une commande CLI pour créer un composant standalone nommé SupplierListComponent :
2. Ajoutez une propriété message au composant et affichez-la dans son template avec une simple interpolation :
3. <h2>{{ message }}</h2>
4. Configurez le lazy loading pour ce composant dans le fichier **app-routing.module.ts**, et testez son fonctionnement en naviguant à l'URL /suppliers.

**Exercice 2 : Utiliser \*ngFor pour Générer une Liste Dynamique**

1. Créer la class Supplier dans le dossier models :

public id: number,

public name: string,   
public company: string   
public phone: string  
private isVerified: boolean = false // Statut de vérification (par défaut : non vérifié)

1. isVerified est une propriété privée et peut être modifiée par une méthode et accédée par unt autre méthode.
2. Ajoutez une liste statique de fournisseurs dans SupplierListComponent.
3. Utilisez la directive \*ngFor dans le template pour afficher chaque fournisseur sous forme de liste dans un composant SupplierListElementComponent.

**Exercice 3 : Afficher un Message Conditionnel avec \*ngIf**

1. Modifiez le template pour afficher un message si la liste de fournisseurs est vide.
2. Utilisez la directive \*ngIf pour conditionner l’affichage.

**Exercice 4 : Appliquer des Classes Dynamiques avec [class]**

1. Utilisez [class] pour appliquer une classe CSS qui change la couleur du fournisseur en fonction de son statut (verified ou non).

**Exercice 5 : Créer un Composant Enfant pour Afficher un Fournisseur**

1. Créez un composant SupplierCardComponent standalone pour afficher un fournisseur.
2. Passez chaque fournisseur du composant parent (SupplierListComponent) au composant enfant via un @Input.
3. Modifiez le composant parent pour utiliser le composant enfant avec \*ngFor.

**Exercice 6 : Ajouter des Boutons pour Modifier l’État**

1. Ajoutez un bouton à chaque carte fournisseur pour marquer un fournisseur comme vérifié.
2. Modifiez dynamiquement la propriété isVerified via un (click) binding.
3. Utilisez [class] pour appliquer une classe CSS qui change la couleur du fournisseur en fonction de son statut (verified ou non).

**Exercice 8 : Créer une Directive Personnalisée**

1. Créez une directive qui souligne le texte d’un élément lorsqu’il est survolé.
2. Appliquez cette directive aux cartes fournisseurs dans le template.