**Objectif**

Créer un système CRUD (Créer, Lire, Mettre à jour, Supprimer) type-safe pour un modèle de données générique en TypeScript.

**Instructions**

**Définir un Modèle Générique :**

Créer une interface générique `Model` qui représente une entité avec un identifiant de type `number`.

**Implémenter un Répertoire Générique :**

Créer une classe `Repository` qui accepte un type générique `T` étendant `Model`.

Implémenter les méthodes `create`, `read`, `update` et `delete` pour le répertoire.

Assurer la sécurité des types et l'immuabilité lorsque cela est applicable.

**Définir des Modèles Spécifiques :**

Créer deux modèles spécifiques `User` et `Product` qui étendent l'interface `Model`.

Le modèle `User` doit avoir des propriétés supplémentaires `name` et `email`.

Le modèle `Product` doit avoir des propriétés supplémentaires `name` et `price`.

**Créer des Répertoires pour des Modèles Spécifiques :**

Instancier des répertoires pour `User` et `Product`.

Effectuer des opérations CRUD sur les deux répertoires pour démontrer la fonctionnalité.

**Implémenter la Sécurité des Types et Types Utilitaires :**

Assurer que les méthodes du répertoire acceptent et retournent uniquement des types appropriés.

Utiliser des types utilitaires pour rendre certaines propriétés en lecture seule lorsque cela est applicable.