**Les principes SOLID, DRY et KISS**

Les principes SOLID, DRY et KISS sont des principes de conception logicielle largement utilisés pour créer des systèmes logiciels robustes, maintenables et évolutifs. Voici une brève description de l’utilisation de chaque principe dans le projet:

1. SOLID : Le projet a été développé dans différents modules et composants grâce à l’utilisation de la bibliothèque de React et dans le cas du serveur API, on a divisé chaque composant dans différents dossiers distincts pour différencier les responsabilités de chaque composant. SOLID est un acronyme pour cinq principes de conception orientée objet (MIcrosoft Learn, 2023) :
   1. Single Responsibility Principle: Chaque composant et module crée ont une seule responsabilité clairement définie.
   2. Open-Closed Principle : Si le client veut ajouter des nouvelles fonctionnalités à la page, c’est juste de créer nouveaux composants et les ajouter sans modifier le site.
   3. Liskov Substitution Principle : Les sous-composants juste ajoutent des fonctionnalités, ils ne les altèrent pas.
   4. Interface Segregation Principle : le design du site web a été développé en plusieurs interfaces qui donnent des fonctionnalités spécifiques par exemple la page de recherche spécifique, la page de catalogue, la page de vendre voiture, etc.
   5. Dependency Inversion Principle : si le client décide d’effacer une composant de bas niveau, il pourra le faire parce que ces composants ne dépendent pas des composants autres de haut niveau.
2. DRY : chaque module et composant n’existe qu’une seule fois dans le projet, c’est-à-dire on évite la duplication de code pour éviter les répétitions. De plus, la structure de dossier en couche permet de suivre le principe DRY (MIcrosoft Learn, 2023).

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

Description générée automatiquement

1. KISS : le code crée adopte le principe de la simplicité pour éviter des solutions surdimensionnées.

# Bibliographie

MIcrosoft Learn. (2023, 05 17). *Common web application architectures*. Récupéré sur https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures#solid-principles