

## PROJET 2

---

### Pondération et échéance

- ☐ Ce projet est divisé en 3 parties et compte pour 4 % de la note finale du bulletin.
- ☐ Date de remise de l'énoncé : **Lundi, 26 octobre 2020**
- ☐ Date de remise de la solution : **Lundi, 16 novembre 2020 en fin de journée**

Une pénalité de 10 % par jour de retard, incluant les jours de congé, sera imposée, et ce, jusqu'à concurrence de 5 jours. (Après ce délai, la note attribuée est zéro.)

### Barème de correction

Exécution de l'application (l'application fait ce qu'elle a à faire !)	100 %
Respect des spécifications	jusqu'à -100 %
Interface utilisateur (présentation et navigation)	jusqu'à -20 %
Disposition du code source	jusqu'à -20 %

Vous serez évalués sur le produit final. (Une application ne produisant pas des résultats tangibles ne vaut rien.) Mieux vaut remettre une application incomplète, mais fonctionnelle qu'une qui (semble) complète, mais qui ne produit pas les bons résultats.

## Énoncé du problème

On souhaite créer une application qui permet de gérer des autos selon leurs marques et leurs couleurs. Chaque auto est définie par son identifiant, son nom, son modèle, l'identifiant de sa marque et l'identifiant de sa couleur. Chaque marque est définie par son identifiant et son nom de marque.

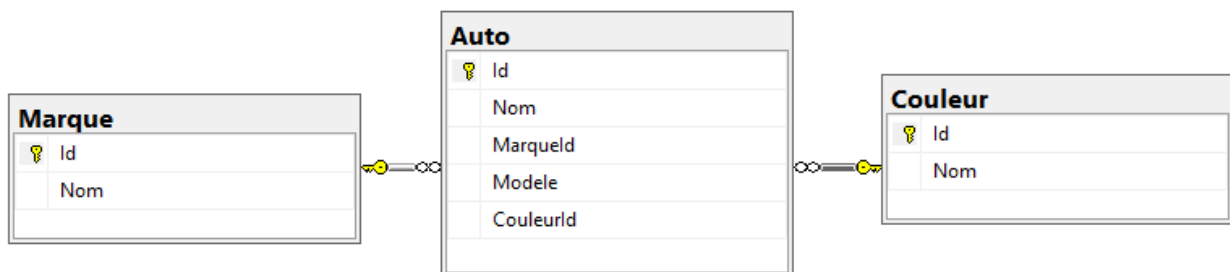
L'application doit permettre de :

- ✓ Afficher toutes les autos.
- ✓ Afficher les détails d'une auto.
- ✓ Saisir et ajouter une nouvelle auto.
- ✓ Éditer et modifier une auto.
- ✓ Supprimer une auto.
- ✓ Afficher toutes les marques.
- ✓ Afficher les détails d'une marque.
- ✓ Saisir et ajouter une nouvelle marque.
- ✓ Éditer et modifier une nouvelle marque.
- ✓ Supprimer une nouvelle marque.
- ✓ Afficher toutes les couleurs.
- ✓ Afficher les détails d'une couleur.
- ✓ Saisir et ajouter une nouvelle couleur.
- ✓ Éditer et modifier une nouvelle couleur.
- ✓ Supprimer une nouvelle couleur.

Les autos, les couleurs et les marques d'autos seront stockés dans une base de données qui va être créée par l'approche Code First d'Entity Framework Core. Le nom de la base de données : **BDW56Projet2VotrePrénom** et doit contenir trois tables :

- Table Auto
- Table Marque
- Table Couleur

La base de données à utiliser dans ce projet est illustrée dans le digramme suivant :



## Partie 2 du projet

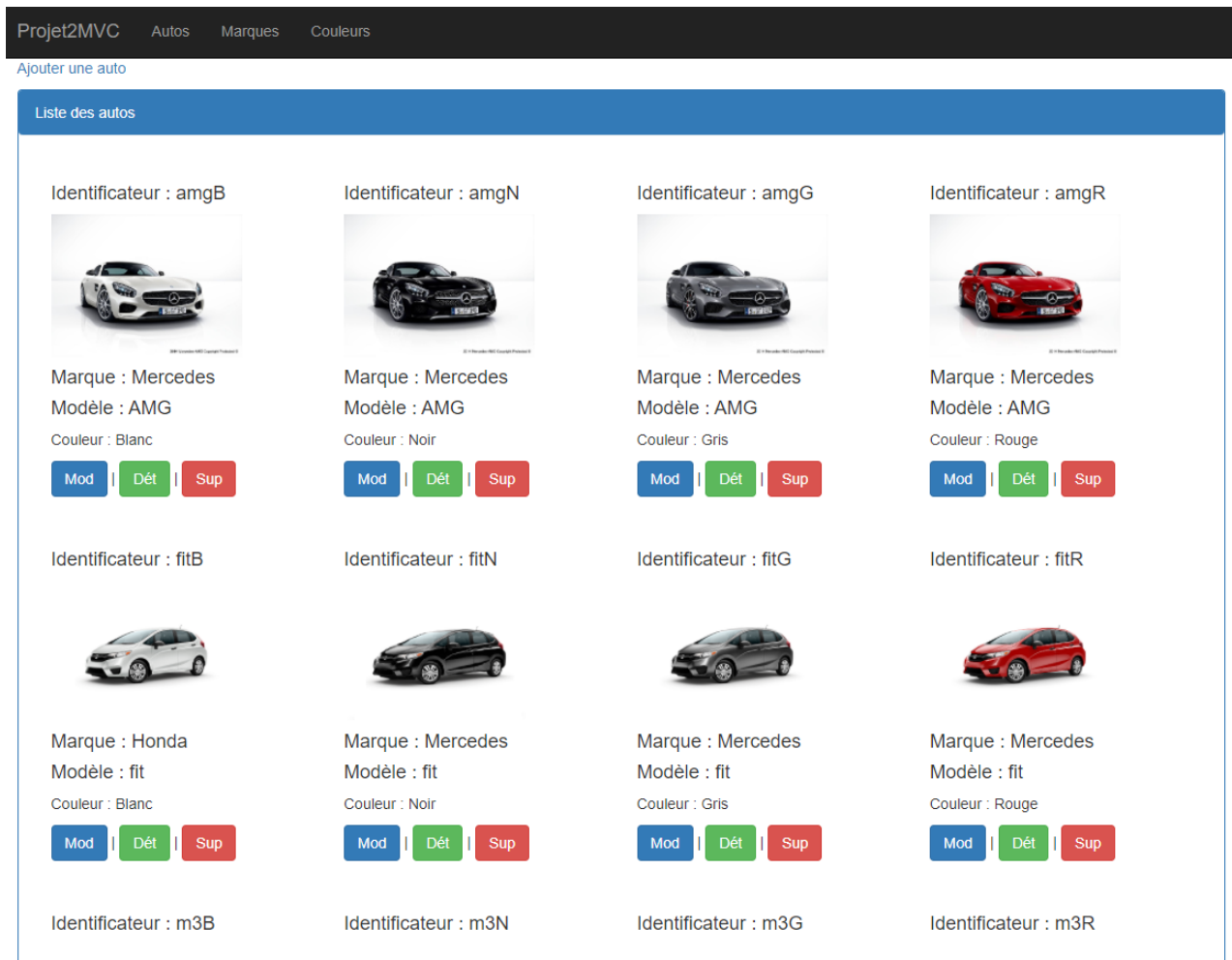
### Travail à effectuer :

L'objectif de cette partie de projet est de développer une application *ASP.NET* Core MVC permettant de manipuler des données des autos utilisant la base de données donnée dans le diagramme précédent.

## Présentation des fonctionnalités de l'application

### 1. Les données des autos

- Immédiatement au démarrage, l'application affiche une page qui contient : la liste des autos avec leurs images, un lien vers une page permettant d'ajouter les données d'une nouvelle auto.

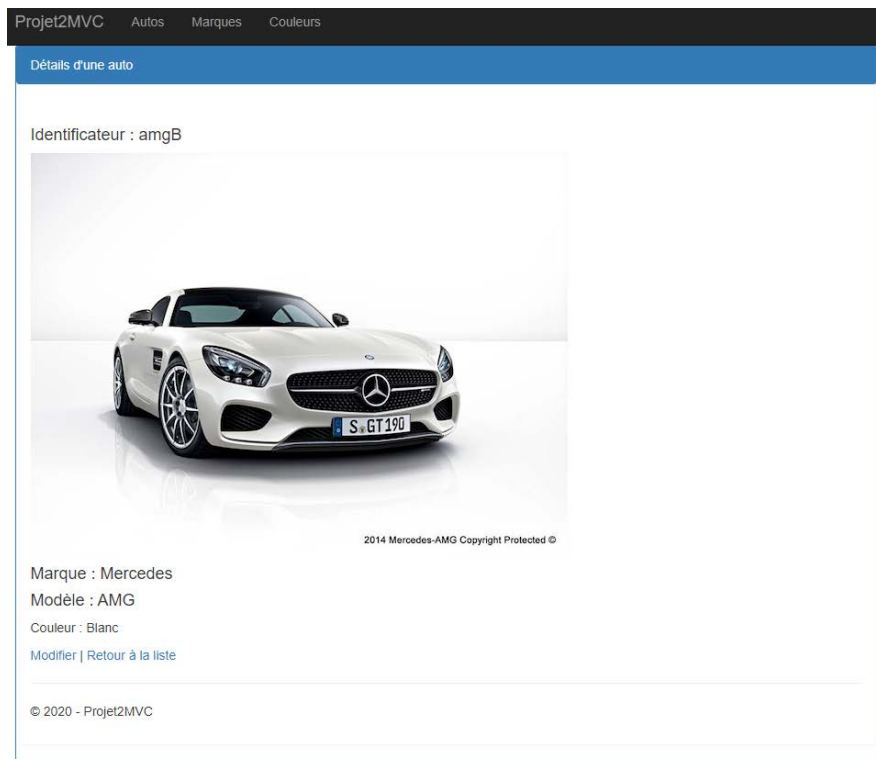


Exemple d'une page avec une liste d'autos et des liens vers d'autres pages

- Pour chaque élément dans la liste des liens vers des pages permettant :

➤ L'affichage des données détaillées d'une auto

- Pour chaque auto, voici les données qui devraient s'afficher :
  - Le nom de l'auto [Nom]
  - Le modèle de l'auto [Modele]
  - La marque de l'auto [Marque]
  - La couleur de l'auto [Couleur]
  - Une image de l'auto qui est enregistrée dans un fichier dans le dossier **wwwroot/images**. Le nom de ce fichier est associé à la propriété Nom de l'auto.



➤ La modification des données d'une auto existante

- Toutes les données saisies pour l'auto s'affichent comme si l'option *Affichage des données détaillées d'une auto* avait été sélectionnée.

➤ La suppression d'une auto

- Toutes les données saisies pour l'auto s'affichent comme si l'option *Affichage des données d'une entreprise* avait été sélectionnée.
- Une confirmation est demandée à l'utilisateur.

## 2. Les données des modèles

- L'application affiche une page qui contient : la liste des modèles, un lien vers une page permettant d'ajouter les données d'un nouveau modèle, et pour chaque élément dans la liste des liens vers des pages permettant :
  - L'affichage des données détaillées d'un modèle
    - Pour chaque modèle, voici les données qui devraient s'afficher :
      - Le nom de modèle [Modele]
  - La modification des données d'un modèle existant
    - Toutes les données saisies pour le modèle s'affichent comme si l'option *Affichage des données détaillées d'un modèle* avait été sélectionnée.
  - La suppression d'un modèle
    - Toutes les données saisies pour le modèle s'affichent comme si l'option *Affichage des données d'un modèle* avait été sélectionnée.
    - Une confirmation est demandée à l'utilisateur.

## 3. Les données des couleurs

- L'application affiche une page qui contient : la liste des couleurs, un lien vers une page permettant d'ajouter les données d'une nouvelle couleur, et pour chaque élément dans la liste des liens vers des pages permettant :
  - L'affichage des données détaillées d'une Couleur
    - Pour chaque couleur, voici les données qui devraient s'afficher :
      - Le nom de la couleur [Couleur]
  - La modification des données d'une couleur existant
    - Toutes les données saisies pour la couleur s'affichent comme si l'option *Affichage des données détaillées d'une couleur* avait été sélectionnée.
  - La suppression d'une couleur
    - Toutes les données saisies pour la couleur s'affichent comme si l'option *Affichage des données d'une couleur* avait été sélectionnée.
    - Une confirmation est demandée à l'utilisateur.

## 4. Barre ou menu de navigation

- Ajouter une barre ou un menu de navigation pour simplifier à l'utilisateur l'accès aux différentes pages de votre application.

## Partie 2 du projet

### Travail à effectuer :

L'objectif de cette partie de projet est de développer une application *ASP.NET* Core Razor Pages permettant de manipuler des données des autos avec les mêmes fonctionnalités développées pour l'application *ASP.NET* Core MVC dans la partie 1 de projet.

## Partie 3 du projet

### Travail à effectuer :

L'objectif de cette partie de projet est d'ajouter à l'application *ASP.NET* Core MVC développée dans la partie 1 de projet des fonctionnalités suivantes :

#### 1. Connexion à l'application

- L'utilisateur entre son **identifiant** (Email) et son **mot de passe** pour se connecter à l'application afin d'accéder aux informations des autos.
- Par la suite, l'application valide le combo **identifiant/mot de passe**. Si les données fournies sont correctes, l'utilisateur est autorisé à poursuivre, autrement il est redirigé à la page de connexion.
- **Il doit y avoir redirection automatique à la page de connexion si l'utilisateur tente de contourner le processus de connexion.**

#### 2. Création d'un compte utilisateur pour la connexion à l'application

- Afin que l'utilisateur puisse se connecter à l'application, il doit d'abord créer un compte utilisateur dans l'application en entrant son **identifiant** (Email), son **mot de passe** et en confirmant son **mot de passe** pour se connecter à l'application.
- Après l'inscription l'utilisateur est redirigé à la page de connexion pour se connecter à l'application afin d'accéder aux informations des autos.

Exemple d'un compte qui sera utilisé pour tester votre application

Nom d'utilisateur : test@gmail.com

Mot de passe : 123456

### Remise :

Remettre sur Léa ou via OneDrive les deux projets compressés en zip au plus tard le lundi 16 novembre 2020 en fin de journée.

- Projet en *ASP.NET* Core MVC avec authentification (partie 1 + partie 3).
- Projet en *ASP.NET* Core Razor Page (partie 2).