M. Larmoire est artisan menuisier ébéniste en Normandie, dans le Nord de la France. Il crée, rénove et fabrique des meubles et des éléments de décoration : tables, portes, armoires, chaises, frises et sculptures en bois massif, qu’il réalise sur mesure. Il souhaite faire connaitre ses réalisations et envisage la création d’un site web qui les présenterait. Il n’a pas du tout de site web pour le moment et ne possède aucune compétence en la matière.

Le nouveau site web utilisant ASP.NET Core pour améliorer ses fonctionnalités et sa maintenabilité. Voici une proposition d'architecture pour une telle application web, comprenant des diagrammes de classes, des modèles de base de données pour SQL Server, des diagrammes de cas d'utilisation et des exemples de fichiers CSHTML.​

**1. Diagramme de classes**

L'architecture proposée comprend les classes principales suivantes :​

* **Meuble** : Représente un meuble avec des propriétés telles que l'ID, le nom, la description, le style, le type de bois, la date de création et l'URL de l'image.​
* **Style** : Définit le style d'un meuble (par exemple, Japonais, Moderne).​
* **Client** : Représente un client avec des informations telles que l'ID, le nom, l'email et le numéro de téléphone.​
* **Order**  : Associe un client à un ou plusieurs meubles commandés, avec des détails tels que l'ID de la commande, la date et le statut.​
* **OrderItems**  : Associe à Orders les détails d’une commande tels que l'ID de la commande, etc.​

Voici une représentation simplifiée en pseudo-code :​

class Meuble {

int MeubleID;

string Nom;

string Description;

Style Style;

string TypeDeBois;

DateTime DateCreation;

string ImageURL;

}

class Style {

int StyleID;

string Nom;

}

class Client {

int ClientID;

string Nom;

string Email;

string Telephone;

}

class Order {

int CommandeID;

Client Client;

List<Meuble> Meubles;

DateTime DateCommande;

string Statut;

}

**2. Modèle de base de données pour SQL Server**

Le modèle relationnel correspondant comprend les tables suivantes :​

* **Meubles** : Stocke les informations sur les meubles.​
* **Styles** : Contient les différents styles disponibles.​
* **Clients** : Enregistre les informations des clients.​
* **Orders** : Enregistre les commandes passées par les clients.​
* **OrderItems** : Table intermédiaire pour gérer la relation plusieurs-à-plusieurs entre les commandes et les meubles.​

Les relations entre les tables sont définies par des clés étrangères appropriées.​

**3. Diagramme de cas d'utilisation**

Les principaux cas d'utilisation pour le site web sont :​

* **Consulter les meubles** : Les visiteurs peuvent parcourir la galerie de meubles disponibles.​
* **Rechercher par style** : Les utilisateurs peuvent filtrer les meubles selon le style souhaité à venir.
* **Contacter l'artisan** : Les clients potentiels peuvent envoyer des messages via un formulaire de contact.​
* **Passer une commande** : Les clients peuvent sélectionner des meubles et passer une commande en ligne.​

**4. Exemples de fichiers CSHTML**

* **Pageprincipal, Index.cshtml** : Page d'accueil affichant une introduction et une galerie de meubles.​
* **Details.cshtml** : Affiche les détails d'un meuble spécifique (à faire plus tard).
* **Contact, index.cshtml** : Formulaire permettant aux clients de contacter l'artisan.​
* **Shop, Index.cshtml** : Page permettant de passer une commande.​

Ces fichiers CSHTML utilisent le moteur de vue Razor pour intégrer du code C# dans le HTML, facilitant ainsi la génération dynamique de contenu.​

En adoptant cette architecture, le site web de M. Larmoire bénéficierait d'une structure robuste, facilitant la gestion des contenus et l'interaction avec les clients, tout en offrant une expérience utilisateur améliorée.

Voici un exemple détaillé des tables pour la base de données SQL Server avec leurs attributs et contraintes.

**1. Table Meuble**

Stocke les informations sur les meubles créés par l'artisan.

CREATE TABLE Meuble (

MeubleID INT PRIMARY KEY IDENTITY(1,1),

Nom NVARCHAR(255) NOT NULL,

Description NVARCHAR(MAX),

StyleID INT NOT NULL,

TypeDeBois NVARCHAR(100),

DateCreation DATETIME DEFAULT GETDATE(),

ImageURL NVARCHAR(500),

FOREIGN KEY (StyleID) REFERENCES Styles(StyleID)

);

**2. Table Style**

Liste les styles de meubles disponibles (ex: Moderne, Rustique, Japonais).

CREATE TABLE Styles (

StyleID INT PRIMARY KEY IDENTITY(1,1),

Nom NVARCHAR(100) UNIQUE NOT NULL

);

**3. Table Client**

Stocke les informations des clients.

CREATE TABLE Clients (

ClientID INT PRIMARY KEY IDENTITY(1,1),

Nom NVARCHAR(255) NOT NULL,

Email NVARCHAR(255) UNIQUE NOT NULL,

Telephone NVARCHAR(20),

Adresse NVARCHAR(500)

);

**4. Table Order**

Stocke les commandes passées par les clients.

CREATE TABLE Commandes (

CommandeID INT PRIMARY KEY IDENTITY(1,1),

ClientID INT NOT NULL,

DateCommande DATETIME DEFAULT GETDATE(),

Statut NVARCHAR(50) CHECK (Statut IN ('En attente', 'Confirmée', 'Expédiée', 'Annulée')),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)

);

**5. Table OrdersItem**

Gère la relation plusieurs-à-plusieurs entre Commandes et Meubles.

CREATE TABLE **OrdersItem** (

CommandeID INT,

MeubleID INT,

Quantite INT DEFAULT 1 CHECK (Quantite > 0),

PRIMARY KEY (CommandeID, MeubleID),

FOREIGN KEY (CommandeID) REFERENCES Commandes(CommandeID) ON DELETE CASCADE,

FOREIGN KEY (MeubleID) REFERENCES Meubles(MeubleID) ON DELETE CASCADE

);

Ces tables assurent :

* Une gestion efficace des meubles et de leurs styles.
* La possibilité pour les clients de commander plusieurs meubles.
* Un suivi des commandes avec des statuts précis.

**Bibliographie détaillée**

Microsoft documentation;

Internet google recherche pour template par exemple;

Chapgpt pour valider mon code source et trouver mes erreurs de codage;

Utilisation du code source du professeur;