

① Programmation Orientée Objet en C# (POO)

POO est un paradigme de programmation basé sur la manipulation d'objets qui regroupent des attributs et des méthodes.

↳ concepts clés.

- **classe**: un modèle définit des attributs et des méthodes
- **objet**: une instance concrète d'une classe créée avec new
- **Encapsulation**: protéger les données via des attributs privés et des méthodes publiques.
- **Héritage**: une classe peut hériter d'une autre pour réutiliser son code
- **Polymorphisme**: un même objet peut avoir plusieurs comportements selon le contexte.
- **Abstraction**: cacher les détails d'implémentation et montrer l'essentiel
- **constructeur**: sert à initialiser un objet lors de sa création
- **Modificateurs d'accès**: Ils définissent qui peut accéder aux variables/méthodes
 - public: accessible partout
 - private: accessible seulement dans la classe.
 - protected: accessible dans la classe et classes filles
- **Interface**: contient uniquement des méthodes sans corps.
- **List<T>**: liste modifiable (ajout, suppression).
- **IEnumerable<T>**: collection en lecture seule.

② ASP.NET (MVC)

.NET est une plateforme de développement créée par Microsoft qui permet de développer, exécuter et déployer des applications.

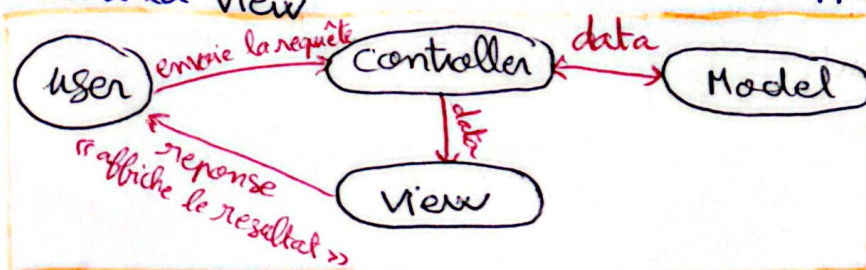
CLR (Common Language Runtime): c'est le **moteur d'exécution** de .NET

↳ **MVC** (Model - View - Controller): c'est une architecture qui permet de séparer le code pour mieux organiser l'application

Model: représente les données et la logique métier.

View: s'occupe uniquement de l'affichage et l'interface utilisateur (HTML)
⇒ fichier **.cshtml**

Controller: reçoit les requêtes de l'utilisateur, appelle le Model, envoie les données à la View



} comment MVC travaille (flux)

③ Creation d'une page web avec ASP.NET (fichiers)

- ↳ controllers/ ⇒ logique, métier, actions (C#)
- ↳ models/ ⇒ Données, objets, classes (C#)
- ↳ Views/ ⇒ Pages HTML dynamiques (.cshtml)
- ↳ shared/ ⇒ Layout general (- Layout.cshtml)
- ↳ home/ ⇒ Pages spécifiques au contrôleur Home.
- ↳ App-Start/ ⇒ Fichiers de config (RouteConfig.cs)
- ↳ Global.asax ⇒ Point de départ de l'app, appelle RouteConfig
- ↳ packages.config ⇒ Packages NuGet installés.
- ↳ web.config ⇒ Config de l'application (base données ...)

Le routage : est un mécanisme qui détermine comment une URL entrante est mappée à un contrôleur et une action {controller}/{action}/{id?}

④ Entity Framework Core (EF core)

- **ORM** qui permet de manipuler la base de données avec des classes C#
- **EDM** (Entity Data Model) est une représentation en mémoire de l'ensemble des métadonnées.

• Sans EF core

- on écrit du SQL
- on gère manuellement la connexion, les requêtes, les résultats

• Avec EF core

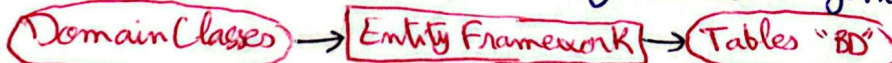
- on travaille avec des classes C#
- EF core s'occupe de créer le SQL, exécuter les requêtes, convertir les résultats en objets.

⑤ Les différentes approches pour créer une base de données.

1. Database First: on crée la base de données et Entity Framework génère les entités à partir de cette base de données



2. Code First: on crée les entités puis Entity Framework génère la BD



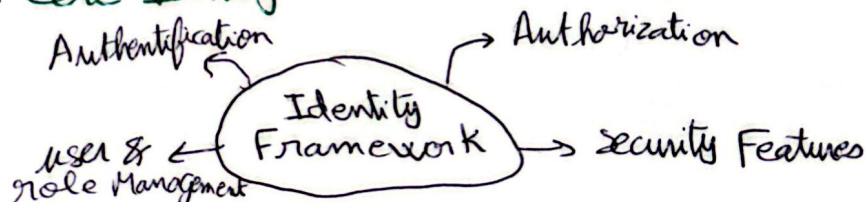
- **Migrations :** servent à synchroniser la base de données avec les classes C#

Add - Migration Initial Create ← créer une migration

Update Database ← Appliquer la migration.

- chaque table = un DbSet.
- **DbContext** = la classe centrale qui connecte l'app à la base, exécute les requêtes

⑥ ASP.NET Core Identity.



→ **Authentification :** Processus de vérification de l'id d'un user.

→ **Autorisation :** Processus de détermination des droits d'accès.

Packages NuGet:

- Microsoft.AspNetCore.Identity.EntityFrameworkCore

- Microsoft. Asp Net Core. Identity. UI
- Microsoft. EntityFrameworkCore. Sql Server.
- Microsoft. EntityFrameworkCore. Tools.

* classe AuthDbContext.cs :

```
namespace Web-App-Migration.Data {
    public class AuthDbContext : IdentityDbContext {
        public AuthDbContext(DbContextOptions<AuthDbContext>
            options) : base(options) {}
    }
}
```

* Appsettings.json.

→ est un fichier de configuration d'une app ASP.NET Core, Il sert à stocker les paramètres de l'app, séparés du code.

* Program.cs.

→ est le fichier où on configure tous les services de l'app au démarrage.

* base Identity (Migration)

→ Generation de la migration : dans la console PM

Add-Migration IdentityInitial - Context AuthDbContext

→ Application de la migration.

Update-Database - Context AuthDbContext

⑦ LINQ to Entities.

LINQ to Entities est l'utilisation de LINQ sur EntityFramework Core. Il permet d'écrire des requêtes en C# pour interroger une base de données sans écrire de SQL.

Types d'opérateurs LINQ.

- Projection : Select, SelectMany
- Filtrage : where, ofType.
- Tri : OrderBy, ThenBy, OrderByDescending.
- Jointures : Join, GroupJoin, Include.
- Agrégation : Count, Sum, Min, Max, Average.
- Groupement : GroupBy.
- Quantification : Any, All, Contains.