# TP3 : JuliePro

## **Consignes (20% de la note finale)**

* Lisez toutes les intructions et la grille avant de commencer
* Il est fortement recommander de partir du projet de départ fourni
* Créez une branche pour le TP3 à partir du Main (vide) et ajoutez le projet fourni.
* Faites les migrations et update-database requis.

## **Étude de cas JuliePro**

Julie Loiselle est propriétaire de l’entreprise d’entrainement personnel JuliePro. Elle offre à ses clients (customer) un suivi personnalisé des entrainements et des objectifs. L’entreprise emploie une douzaine d’entraineurs (trainer) chevronnés et spécialisés (Speciality : perte de poids, haltérophilie, course, réhabilitation, etc). Un Trainer a une seule Speciality.

Julie peut gérer ses entraineurs (*Manage Trainers*) voir la liste, en ajouter, modifier ou supprimer (rendre non disponible), gérer les photos des entraineurs.

Chaque client est assigné à un entraineur qui planifie des séances (ScheduledSession) à chaque semaine (environ 2 à 5 selon le client) qui sont effectuées parfois avec l’entraineur, parfois seule. Le client peut visualiser les séances à venir ainsi que les séances passées (et cocher une séance lorsqu’elle est complétée.

L’entraineur fait un suivi régulier et vérifie que ses clients suivent bien le programme et communique avec lui si les séances s’espacent. Chaque séance est d’un entrainement particulier (CrossFit, Yoga, Course, Step, etc). Chaque entrainement (training) est soit cardio, musculaire, étirement et nécessite 1 ou plusieurs équipements. Le même équipement peut être utilisé dans plusieurs entrainements.

L’entraineur définit également avec son client un objectif courant (objective) : soit de perte de poids ou de distance. Lorsqu’il est atteint, le client indique la date et peut en déterminer un autre avec son entraineur.

Bien qu’il ne se concentre que sur un objectif à la fois qui doit être visible lorsqu’il regarde la liste des scheduledSession, le client pourra donc visualiser à la demande (bouton/lien qui affiche la liste de tous ses objectifs atteints en ordre décroissant). S’il n’y a pas d’objectif de déterminer, le bouton CreateObjective s’affiche à la place de l’objectif.

## **Principales étapes :**

Ne pas travailler directement dans la branche **Production**. Cette branche va contenir la version testée et approuvée (par vous à la place du client). Vous ajoutez une fonctionnalité à la fois, via la fusion (merge) de branche : la branche **Production** reste donc fonctionnelle même si la branche de la fonctionnalité en développement est boguée ou incomplète.

## **NOTE : Le projet Core contient les Services**

## **Section 1**

### Mise en place éléments de départ et Seeds

1. Générez la nouvelle BDpour le TP3 appelée JuliePro\_TP3 (vous aurez ainsi la version complète de départ)
2. Vous pouvez ajouter/modifier les données si vous le souhaitez

### Ajout de Controllers et Views

1. Ajoutez une Partial View qui affiche la liste des *CalendarEvents* (Date et Title)
2. Ajoutez la PartialView sur la View Home, visible à tous (où vous voulez dans la page).
3. Modifiez le code (service/controller) afin de n'afficher que les 3 prochains *CalendarEvents* (dans le Futur), voir les *Seeds* ou le contenu de la BD pour vérifier.
4. Modifiez le code afin de pouvoir lire la description d’un *CalendarEvent* lorsque l’on clique dessus (voir la séance sur JavaScript/Jquery)

### Ajout des services et Tests Unitaires

1. Ajoutez les packages et les références nécessaires lorsque requis
2. Nommez les projets tests adéquatement (n’oubliez pas de changer le namespace par défaut)
3. Créez les tests unitaires suivants:
   1. : Tests sur les services:
      1. Testez l'ajout de *Trainer* se fasse correctement en BD
      2. Ajoutez le service et le test unitaire pour *GetAllActive* ne retourne que les *Trainer* **actifs**
      3. Testez que le que Delete de *Speciality* réussi, l'action retourne à la View *Index*
      4. Ajoutez/Modifiez le service et le test qui valide que la création d’objectif n’est possible que il n’y a pas d’objectif en cours
      5. Ajoutez/modifiez le service et le test qui valide que la création de SheduledSession ne permette pas d’en créer plus de 5 dans le futur pour un *Customer*

### Mise en place MS Identity

1. Décomentez la classe **Initializer** le dossier **Initializer** dans le projet **JuliePro\_DataAccess**
2. Activez l’injection de dépendance dans le **Startup** en décommentant le text en bleu :

public IConfiguration Configuration { get; }

       // This method gets called by the runtime. Use this method to add services to the container.

       public void ConfigureServices(IServiceCollection services)

       {

           services.AddLocalization(options => options.ResourcesPath = "Resources");

….

           /\* Activer après avoir configurer MSIdentity\*/

           services.AddScoped<IDbInitializer, DbInitializer>();

   }

1. Implémentez MS Identity
   1. Les packages requis
   2. Les tables
   3. Les pages autogénérées
   4. L'injection de dépendances
   5. Etc
2. Ajoutez les autorisations ci-dessous en utilisant les constantes (*Utility*)
   1. Seul le **SuperAdministrator** a accès aux Controller et Views de la section *Trainer* 
      1. Exception l'action Index qui peut également être accessible aux **Trainer**; seul le bouton **Details** est accessible (**Edit** et **Delete** sont soit masqués ou Inactifs, à votre choix)
   2. Dans la barre des menus (*NavBar)*:
      1. que le menu **Trainer** ne soit visible que pour les **Trainer** et les **SuperAdministrator**
      2. que le menu **Management** ne soit visible que pour les **SuperAdministrator**

NOTE: la majorité des menus ne mènent null part volontairement