**LProjetFinal\_ XXXXXXX où XXXXXXX est votre matricule.**

**Vous devez faire le projet seul.**

**Sommaire :**

* Simple projet qui regroupera **les notions vues dans le cours**.
* Il n’y aura pas de modélisation puisque vous avez fait cette partie dans votre TP1.
* Il n’y aura pas d’identification/authentification d’usagers à faire.
* Aucun point ne sera alloué pour la beauté de votre projet.
* Vous aurez à faire un projet BD First
* Avec les migrations sql faites à partir de Evolve pour créer la BD, les tables, insérer les données.
* Il y aura une migration qui sera une vue complexe à faire et à montrer le résultat dans votre projet
* Il y aura une migration pour le chiffrement/déchiffrement d’une donnée sensible
* Il y aura une migration qui sera une procédure stockée avec paramètres à exécuter
* Il y aura une migration qui sera le traitement des images, tel que vu dans le cours.
* Il y aura une migration qui créera des index pour améliorer la performance de votre projet, index qui seront utiles compte tenu des recherches que vous allez faire dans les tables.
* Vous devez avoir une page dans laquelle l’utilisateur doit faire des choix, au moins 2, puis appliquer ces filtres avant d'afficher le résultat final. Il faut faire en sorte **d’éviter d'aller chercher des enregistrements inutilement**. N’allez chercher que les enregistrements que vous comptez présenter dans une vue Razor (comme l'étape 6 du labo sur la performance).
* **Vous utiliserez GitHub pour sauvegarder les progrès de votre projet. Lorsque le projet atteint une migration sql, donnez le nom FCT\_Migration\_VX.X\_\_.... à votre commit. Comme cela je pourrai facilement voir l’accomplissement des migrations. N’oubliez pas de me donner accès à votre GitHub à** [**benoit.caldairou@cegepmontpetit.ca**](mailto:benoit.caldairou@cegepmontpetit.ca)

**Explications détaillées :**

1. Vous devez créer votre projet **BD First**. Et vous devrez utiliser Evolve. Vous prenez votre script de votre TP1 et vous le brisez en 3 étapes : InitialCreate.sql, V1\_0\_\_Tables.sql, V1\_1\_\_InsertionDonnees.sql
2. Créez un fichier .txt qui contiendra votre commande evolve pour les migrations et votre commande pour forcer la génération des modèles et du data.
3. Les migrations seront comme suit :

**InitialCreate.sql qui crée la BD**

**V1\_0\_\_CreationTables**

Création des tables avec les contraintes

**V1\_1\_\_Insertion des données**

Insertion des données.

**V1\_2\_\_CreationVue**

Création d’une vue SQL complexe. Modifications dans le projet pour afficher le résultat de la vue.

**V1\_3\_\_ChiffrementDe…**

Création d’une procédure stockée et autre, pour le chiffrement/déchiffrement d’une donnée sensible. Modifications dans le projet pour afficher cette donnée sensible.

**V1\_4\_\_ProcedureStockeeMasterDetail**

Création d’une procédure stockée retournant des valeurs, avec paramètres. Modifications dans le projet pour avoir une vue EF dans laquelle on pourra entrer des données (Master) qui seront ensuite utilisées pour appeler la procédure stockée et afficher le résultat (Détail).

**V1\_5\_\_TraitementDesImages**

Il y aura une migration qui fera le traitement des images, tel que vu dans le cours. Modifications dans le projet pour avoir l’upload de l’image et l’affichage de l’image quelque part.

**V1\_6\_\_Performance**

Il y aura une migration qui créera des index pour améliorer la performance de votre projet, index qui seront utiles compte tenu des recherches que vous allez faire dans les tables.

**Élément de performance dans le code :**

* Vous devez avoir une page dans laquelle l’utilisateur doit faire des choix, au moins 2, puis appliquer ces filtres avant d'afficher le résultat final. Il faut faire en sorte **d’éviter d'aller chercher des enregistrements inutilement**. N’allez chercher que les enregistrements que vous comptez présenter dans une vue Razor (comme l'étape 6 du labo sur la performance).

**Grille de correction :**

|  |  |  |
| --- | --- | --- |
| **ÉLÉMENTS** | **Détails** | **PTS** |
| **Création du projet** | Il faut que le projet s’exécute sans erreurs et qu’il soit bien présenté (pas nécessairement beau mais avec une certaine logique). On doit être capable de retrouver les différents éléments facilement. | 5 |
| **Fichier .txt pour les commandes de migrations et la régénération du modèle** | J’en aurai besoin pour la correction et je ne veux pas avoir à les taper. | 5 |
| **InitialCreate.sql** | Création avec le Filestream. **Ajoutez votre matricule après FG\_Images** comme cela j’aurai des fichiers différents sur C:\EspaceLabo | 5 |
| **V1\_0\_\_CreationTables**  **V1\_1\_\_Insertion des données** | Séparer en au moins deux scripts de migration la création des tables et l’insertion de données. N’oubliez pas que les tables et autres objets SQL doivent être répartis dans au moins 2 schémas. | 5 |
| **V1\_2\_\_CreationVue** | La vue doit être intéressante et complexe. Le projet doit être ajusté pour voir le résultat de la vue. | 10 |
| **V1\_3\_\_ChiffrementDe…** | Bon choix de donnée sensible. Chiffrement et déchiffrement de cette donnée. Affichage de cette donnée déchiffrée dans le projet | 10 |
| **V1\_4\_\_ProcedureStockeeMasterDetail** | Procédure intéressante pour le projet. Le projet utilise la procédure pour présenter le détail des données du master passées en paramètre. | 15 |
| **V1\_5\_\_TraitementDesImages** | Les images devront être traitées telles que vues en classe. Quand vous remettez le projet, certaines images auront été uploadées. D’autres devront l’être dans les tests. | 10 |
| **V1\_6\_\_Performance** | Les index créés devront être basés sur les recherches de données faites dans votre projet. Un minimum de 2 index devront être créés. Avec des explications pourquoi ce sont des bons index dans le contexte de votre projet. | 10 |
| **Performance** | Une page Razor dans laquelle l’utilisateur doit faire des choix, au moins 2, puis appliquer ces filtres avant d'afficher le résultat final. Vous devez faire en sorte **d’éviter d'aller chercher des enregistrements inutilement**. N’allez chercher que les enregistrements que vous comptez présenter dans une vue Razor (comme l'étape 6 du labo sur la performance). | 10 |
| **Document expliquant les tests à faire pour vérifier que tout fonctionne. Avec les données de tests (images, etc)  Genre de démonstration.** | Le document doit être tel que si je suis les instructions du document, je peux tout vérifier. Si certains éléments ne sont pas complétés correctement, le document doit le mentionner. | 5 |
| **Qualité du français** | Votre document expliquant les tests à faire doit être rédigé dans un français correct, ainsi que les commentaires que vous mettrez dans votre code. | 10 |
| **À la fin de chaque cours, je dois pouvoir voir dans votre GitHub une avancée de votre projet final.** | | |

\*Si vous effectuez le travail en équipe, les notes seront ajustées pour refléter la part faite par chacun, en se basant sur les commits faits dans GitHub.