A25 – POO 4 - Examen final (30 %)

I - Objectif de l'examen

L'examen vise à évaluer votre compréhension des notions vues en cours à savoir :

- Protection des données;
- Authentification;
- Gestion des autorisations;
- Journalisation;
- Tests unitaires.

II - Contexte

Ce travail doit être réalisé et remis individuellement en m'envoyant votre code sous forme d'un fichier ZIP via LEA.

Vous pouvez consulter vos notes de cours, toute documentation pertinente, internet, etc. en dehors de communiquer avec votre camarade et des outils ChatGPT et Copilot.

III - Date de remise

Votre travail doit être remis au plus tard le lundi 15 décembre 2025 à 16h30. Vous avez donc 4h30 de temps pour réaliser et le remettre le travail.

Tout travail plagié se verra attribuer la note de 0.

-10% par minute, en cas de remise en retard.

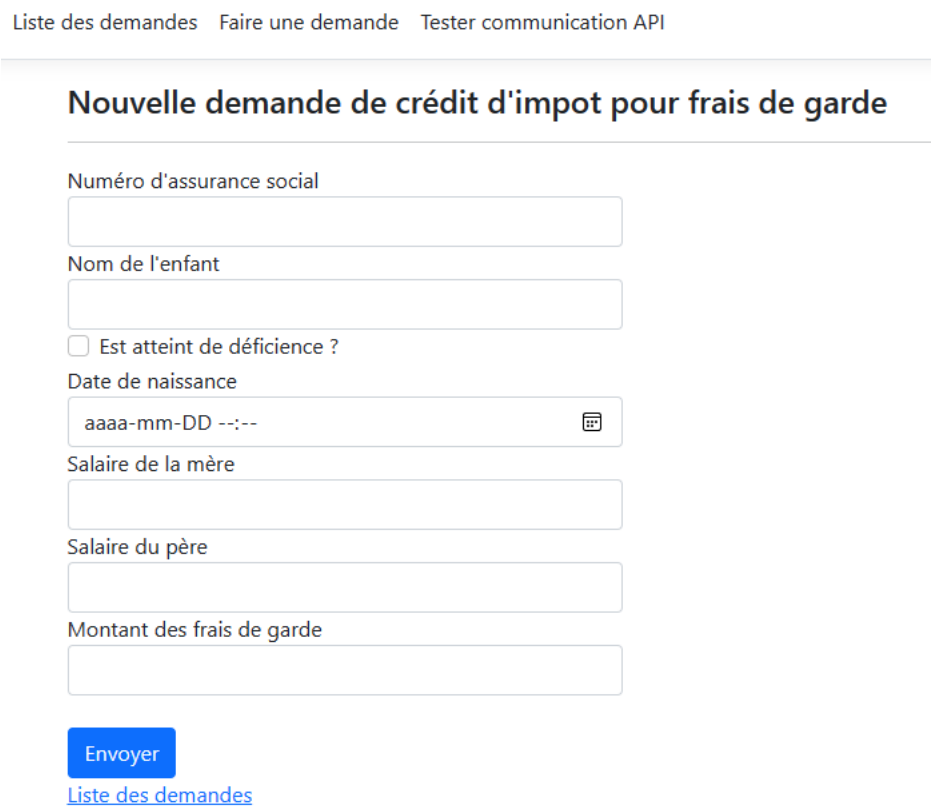
IV - Grille d'évaluation

	Excellent	Fonctionnel	Minimal	Insuffisant
Capacité 1 : Adopter des pratiques de programmation reconnues	**Journalisation et gestion des erreurs :** • La journalisation est toujours correctement utilisée; • Les erreurs sont toujours correctement gérées. **Implémentation des tests unitaires :** • Excellente couverture de code par les tests unitaires; • Utilisation adéquate des outils de tests unitaires ; • Utilisation adéquate des simulacres • Les assertions sont toujours utilisées correctement.	**Journalisation et gestion des erreurs :** • La journalisation est presque toujours correctement utilisée; • Les erreurs sont presque toujours correctement gérées. **Implémentation des tests unitaires :** • Bonne couverture de code par les tests unitaires; • Utilisation presque adéquate des outils de tests unitaires ; • Utilisation presque adéquate des simulacres; • Les assertions sont presque toujours utilisées	**Journalisation et gestion des erreurs :** • La journalisation est la plupart du temps correctement utilisée; • Les erreurs sont la plupart du temps correctement gérées. **Implémentation des tests unitaires :** • Couverture de code par les tests unitaires passable; • Utilisation la plupart du temps adéquate des outils de tests unitaires; • Utilisation la plupart du temps adéquate des simulacres ; • Les assertions sont la plupart du temps utilisées correctement;	**Journalisation et gestion des erreurs :** • La journalisation est rarement correctement utilisée; • Les erreurs sont rarement gérées. **Implémentation des tests unitaires :** • Médiocre couverture de code par les tests unitaires; • Utilisation des outils de tests unitaires rarement adéquate; • Utilisation des simulacres rarement adéquate; • Les assertions ne sont pas correctement utilisées
Capacité 2 : Programmer en utilisant des fonctions avancées du langage	**Sécurité et bonnes pratiques:** • Les bonnes pratiques de programmation sont toujours respectées; • Les données sensibles sont toujours correctement protégées; • L'authentification est toujours correctement utilisée; • La gestion des autorisations est toujours correctement effectuée.	**Sécurité et bonnes pratiques:** • Les bonnes pratiques de programmation sont presque toujours respectées; • Les données sensibles sont presque toujours correctement protégées; • L'authentification est presque toujours correctement utilisée; • La gestion des autorisations est presque toujours correctement effectuée.	**Sécurité et bonnes pratiques:** • Les bonnes pratiques de programmation sont la plupart du temps respectées; • Les données sensibles sont la plupart du temps correctement protégées; • L'authentification est la plupart du temps correctement utilisée; • La gestion des autorisations est la plupart du temps correctement effectuée.	**Sécurité et bonnes pratiques:** • Les bonnes pratiques de programmation ne sont pas respectées; • Les données sensibles sont rarement protégées; • L'authentification est la rarement utilisée; • La gestion des autorisations est rarement effectuée.

Mise en contexte

Revenu Québec vous a mandaté d'implémenter une application permettant de faire des demandes de crédit d'impôt pour frais de garde.

Pour faire une demande, le formulaire suivant doit être rempli.



Une fois qu'une demande est transmise, Revenu Québec procède au calcul du crédit d'impôt pour frais de garde selon les éléments suivants :

- Les parents ont droit au crédit si le montant des frais de garde payé dépasse 2300$;
- Le crédit d'impôt pour frais de garde est calculé sur la différence entre 2300$ et le montant payé par le parent;
- Le taux appliqué tient compte du revenu familial (somme du salaire du père et de la mère);
- Le barème de taux suivant est appliqué :

Supérieur ou égal à	Inférieur à	Taux
0	22 000	80%
22 000	40 000	75%
40 000	60 000	72%
60 000		60%

Travail à faire

Avant de réaliser ce qui est demandé, prenez le temps d'exécuter les applications et voir ce qui est déjà fait, comment ça fonctionne et vous assurer que tout est correct. Des erreurs ont été volontairement laissées dans le code.

Exercice 1 : Journalisation (5 points)

Apportez les ajustements au ServiceProxy (classe FraisGardeServiceProxy) pour répondre aux besoins suivants :

- Les codes HTTP retournés par l'API doivent être journalisés. Les codes 2xx avec le niveau information, les codes 4xx avec le niveau erreur, et les codes 5xx avec le niveau critique.
- Pour les codes 5xx vous devez générer une exception de type HttpRequestException avec pour message « **Erreur grave dans l'API.** »

Exercice 2 : Protection des données (7 points)

Le NAS étant une information sensible, vous devez assurer sa protection. Pour cela, vous devez :

1. Protéger le NAS saisi par l'utilisateur dans le champ de filtre avant de l'envoyer à l'API;
2. Modifier la méthode Get de la classe FraisGardeController de l'API pour protéger le NAS des informations qui seront retournées à l'application MVC et effectuer adéquatement le filtre sur le NAS;
3. Ajuster la méthode Index de la classe DemandeCreditController de l'application MVC pour déchiffrer le NAS avant de le retourner à la vue.

Note : Un même mot chiffré à plusieurs reprises avec la même clé, le même algorithme ne produit pas le même résultat.

Vous n'avez pas besoin de protéger le NAS lors de la création d'une demande.

Exercice 3 : Authentification et autorisation (5 points)

1. On souhaiterait mettre en place la gestion des autorisations pour protéger le contrôleur DemandeCreditController de l'application MVC. Apportez les ajustements nécessaires à l'application pour supporter les autorisations en tenant compte des éléments suivants :
    - Il faut être authentifié pour accéder à n'importe quelle page de ce contrôleur;
    - Les personnes avec le rôle Utilisateur ou Gestionnaire ou Administrateur peuvent accéder à la page Index permettant d'afficher la liste des demandes (méthode d'action Index);
    - Les personnes avec le rôle Utilisateur et Gestionnaire ou Administrateur peuvent accéder à la page permettant d'enregistrer une demande (méthodes d'action Create).
2. Configurez le mot de passe afin qu'il respecte les conditions suivantes :
    - La taille du mot de passe doit être de 8 caractères au minimum;
    - Les majuscules et minuscules ne sont pas obligatoires;
    - Le mot de passe doit compter des caractères alphanumériques;
    - Le mot de passe doit avoir au maximum 2 caractères identiques.

Les utilisateurs suivants ont été créés pour vous permettre de tester votre implémentation :

Nom	Mot de passe	Rôle
test1@test.com	Test1123*	Utilisateur
test2@test.com	Test2123*	Utilisateur, Gestionnaire
test3@test.com	Test3123*	Administrateur

Exercice 4 : Tests unitaires (8 points)

1. En tenant compte de l'énoncé, écrivez les tests unitaires permettant de tester la méthode **CaculerCredit** de la classe CalculCredit. Vous devez utiliser FluentAssertions pour vos assertions.
2. Testez unitairement la méthode **Create (HttpPost)** de la classe DemandeCreditController de l'application MVC en tenant compte de ce qui suit :
    a. L'API n'est pas appelée si le montant des frais est inférieur ou égal à 2300$ et le message suivant est affiché : « Le Montant des frais de garde doit être supérieur à 2 300$ pour avoir droit au crédit. »;
    b. Si l'API retourne le code statut « Created », rediriger l'utilisation vers l'Index;
    c. Sinon retourner dans le ModelState une erreur avec pour clé « Erreur du service Web »;
    d. En cas d'erreur les informations saisies par l'utilisateur doivent être retournées dans la vue.

Bon Examen!