Examen Final - Sommatif H24

# 420-5W5

## Consignes

* Vous avez 1h50 pour compléter l’examen.
* Toute documentation est permise pour l’examen.
* L’utilisation d’IA est **interdite.**

# Question 1 – Tests unitaires et Mock (8 points)

À partir des fichiers de départs, tester **SeatsController**.

Le contrôleur est utilisé dans un système de réservation de places pour différents concerts et évènements.

Pour tester le contrôleur, vous devrez utiliser un ou plusieurs mock.

Vous devrez donc utiliser le package Nuget Moq.

ReserveSeat

L’action sert à réserver une place dans une salle de concert. Il doit y avoir 4 tests pour couvrir les 4 cas suivants. Chaque cas doit inclure un assert pour vérifier le résultat de l’appel à **ReserveSeat**.

1. Lorsque l’action s’exécute correctement, la place (classe Seat) est retournée à l’utilisateur.
2. Dans le cas où l’utilisateur connecté essaie de réserver une place qui est déjà réservée par un autre utilisateur, l’action retourne **Unauthorized**.
3. Dans le cas où le numéro de la place sélectionnée est plus que le maximum (la salle contient 100 places), l’action retourne **NotFound**.
4. Dans le cas où l’utilisateur connecté essaie de réserver une place, mais qu’il a déjà une place de réservée, l’action retourne **BadRequest**.

# Question 2 – BackgroundService et SignalR (8 points)

|  |  |
| --- | --- |
| **Section** | **Points** |
| Question 1 | |
| Test de ReserveSeat qui fonctionne correctement | 2 |
| Test de ReserveSeat avec une place qui est déjà réservée | 2 |
| Test de ReserveSeat avec une place qui n’existe pas | 2 |
| Test de ReserveSeat avec un utilisateur qui a été une autre place de réservée | 2 |
| Question 2 | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Total** | **16** |