Examen 1 – 10%

**Vous venez d’intégrer une équipe qui travaille sur une plateforme de réservation d'hôtels.**

**Le projet est déjà commencé, voici les tâches qui vous sont assignées :**

1. Récupérer le projet sur Git
2. Corriger l’action Index pour afficher la liste d’hôtels
3. Modifier l’action Rechercher pour trouver le premier hôtel disponible : le mot clé cherché doit pouvoir correspondre au nom de l’hôtel ou à la ville. Ajouter la gestion d’erreur.
   1. Tester que tous les liens prévus fonctionnent maintenant.
   2. Modifier aussi la vue pour indiquer le nombre de chambres disponibles dans l’hôtel trouvé.
4. Créer une liste de 3 utilisateurs
5. Ajoutez une route, une action et une vue pour afficher la liste de tous les clients.
   1. Cette page doit avoir la même apparence générale que les autres.
   2. Ajouter un lien vers cette page.
6. Créer un modèle de vue pour afficher une réservation. Un utilisateur pourra seulement réserver une chambre à la fois.
7. Programmer et documenter l’action Réserver qui doit permettre de créer et d’afficher la réservation d’un utilisateur pour une chambre d’hôtel.
   1. La route a déjà été programmée : vous devez tester votre méthode en entrant le nom d’utilisateur et le numéro de l’hôtel directement dans l’URL.
   2. Valider les données : afficher des messages précis en cas d’erreur.
   3. Créer une vue et utiliser le modèle de vue pour afficher les détails de la réservation : numéro, nom de l’hôtel, ville, nom et prénom du client.
8. Accéder au projet de tests unitaires :
   1. Exécuter le test et assurez-vous de sa réussite
   2. Compléter le test pour vérifier que l’hôtel trouvé est bien à Gatineau
   3. Ajouter un test qui vérifie qu’une recherche avec le terme « abcd » ne fonctionnera pas.
9. Ajoutez un footer avec un message drôle.
10. Remettre le projet sur Git en ajoutant l’enseignante (vivgxojo) comme collaboratrice.

**Grille d’évaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critères d’évaluation** | **Excellant** | **Bien** | **Passable** | **Insuffisant** |
| Programmer la logique applicative du côté serveur. | Gestion adéquate des évènements côté serveur, incluant la création/utilisation de modèles/modèles de vues et la validation des données.  10 | Gestion adéquate des évènements côté serveur, confusion dans l’utilisation des modèles/modèles de vues et/ou validation incomplète.  8 | Gestion généralement correcte des évènements côté serveur. Plusieurs erreurs nuisent au fonctionnement et/ou certains éléments sont manquants.  6 | Le projet est majoritairement non fonctionnel.  5-0 |
| Programmer l’interface Web.  Programmer la logique applicative du côté client. | Création/modification adéquate des vues et de la page de disposition. Affichage de toute l’information demandée. Affichage adéquat des erreurs.  Tous les liens sont fonctionnels et tous les textes sont en français correct.  10 | Création/modification correcte des vues et de la page de disposition. Affichage incomplet de l’information demandée.  Affichage des erreurs.  Les liens sont fonctionnels.  8 | Création/modification correcte des vues et de la page de disposition.  Certaines spécifications ne sont pas respectées ou non fonctionnelles.  6 | Des problèmes majeurs dans les vues ou la page de disposition rendent le projet non fonctionnel.  5-0 |
| Contrôler la qualité de l’application. | Les tests unitaires démontrent le fonctionnement de l’application dans des cas normaux et dans des cas d’erreurs.  5 | Les tests unitaires sont présents et fonctionnels et ils démontrent partiellement le fonctionnement.  4 | Les tests unitaires démontrent une certaine logique mais présentent des erreurs.  3 | Aucun ajout aux tests unitaires ou les tests sont trop incomplets ou illogiques.  2-0 |
| Rédiger la documentation. | La documentation est complète et respecte les normes.  5 | La documentation est complète sans respecter les normes.  4 | La documentation est partielle.  3 | Aucune documentation n’a été ajoutée.  0 |

**Total /30**