



Évaluation

Ruby on Rails

Spécifications

L'objectif est la réalisation d'un site web de type **réservation de billets d'avion**.
Ce site devra fonctionner de la façon suivante:

POUR LES UTILISATEURS ANONYMES:

Sur la page d'accueil, une liste **paginée** de Vo/s devra remonter.

Sur cette page, seront affichées les informations suivantes :

- Numéro du vol;
- Aéroport de départ;
- Aéroport d'arrivée;
- Date et heure de départ;
- Date et heure d'arrivée;
- Un bouton de réservation.

Au clic sur le bouton de réservation, un formulaire permettant la réservation d'une place sur ce vol sera affiché. La page comprenant ce formulaire devra reprendre les informations du vol choisi.

Le formulaire demandera les informations suivantes :

- **Choix de la classe**: économique ou affaires;
- **Nombre de passagers** (maximum **99** par réservation).

Une réservation réussie devra générer **l'envoi d'un email de confirmation** à l'utilisateur, reprenant les détails de la réservation (y compris un **identifiant** à 6 caractères alphanumériques pour celle-ci) ainsi que le vol réservé.

Un lien vers le formulaire d'authentification devra être toujours disponible sur le site.

POUR LES UTILISATEURS IDENTIFIÉS:

Les utilisateurs connectés ont la possibilité d'accéder à une interface permettant de gérer leurs réservations :

- **Liste des réservations effectuées**, avec les informations associées;
- **Annulation** d'une réservation.

CONSIDÉRATIONS GÉNÉRALES

1. Un fichier `seeds.rb` permettant de charger des vols dans votre base de données à partir d'un fichier CSV vous sera fourni. Vous êtes libres de le modifier pour l'adapter à votre modèle.
2. **La fonction de réservation sera réservée uniquement aux utilisateurs identifiés.** Si un utilisateur non identifié clique sur le bouton de réservation, il devra être redirigé vers le formulaire d'authentification.
3. On fera particulièrement attention aux différentes règles métier (liste non exhaustive).
Par exemple :
 - (1) Les réservations d'un vol sont ouvertes **jusqu'à l'heure de départ de celui-ci**;
 - (2) **Une réservation ne devra pas aboutir** si le nombre de sièges disponibles dans la classe choisie est inférieur au nombre de passagers indiqué par l'utilisateur. Vous êtes libre de choisir le comportement de votre application pour gérer cette situation.

BONUS

Si vous le souhaitez, **et uniquement après avoir implémenté toutes les fonctionnalités ci-dessus**, vous pourrez réfléchir à implémenter les fonctions supplémentaires de votre choix, dans l'application.

Quelques exemples :

- Possibilité de chercher les vols par aéroport de départ et/ou d'arrivée;
- Réservations payantes avec intégration d'une API de paiement;
- Génération d'un e-billet au format PDF, incluant un code-barres PDF417.
- ...

Tout effort supplémentaire allant au-delà des spécifications sera valorisé (jusqu'à 4 points supplémentaires), mais il vaut mieux se concentrer sur une implémentation correcte des fonctionnalités de base, plutôt qu'une mauvaise implémentation avec des fonctions supplémentaires.

Critères d'évaluation

Critère	Points
Respect des spécifications	4
Schéma de base de données	3
Modèles : cohérence avec le schéma et qualité des validations	5
Contrôleurs : pertinence et routes créées	3
Vues : design général, organisation	3
Qualité et lisibilité du code	2
Fonctionnalités supplémentaires (points bonus maximum)	+4

Rendu

Le rendu du projet se fera par invitation à votre dépôt GitLab contenant le code-source, sur le GitLab de l'IUT (ou GitHub), impérativement avant :

- **le lundi 13 juin 2022 23:59:59 pour le groupe A;**
- **le lundi 11 juillet 2022 23:59:59 pour le groupe B.**

Votre dépôt devra inclure obligatoirement un fichier **README.md** contenant:

- les instructions pour exécuter votre projet;
- Un retour d'expérience sur le projet (par exemple: difficultés, recherches effectuées);