

Description du Projet

Nom du Projet : ICanFly

Type de Projet : Plateforme de réservation de vols spatiaux

Technologie : Symfony

Objectif :

Créer une plateforme interactive permettant aux utilisateurs de réserver des vols spatiaux, consulter les détails des missions et suivre leurs réservations en temps réel. La gestion des rôles et permissions garantit une séparation claire des responsabilités entre clients, opérateurs, et administrateurs.

Objectifs du Projet

1. Permettre aux clients de réserver des sièges pour des missions spatiales.
 2. Fournir aux opérateurs un moyen d'ajouter et de gérer les missions.
 3. Offrir un espace administrateur pour superviser l'ensemble des opérations.
 4. Intégrer des APIs externes (SpaceX, Stripe, YouTube).
 5. Assurer une sécurité robuste avec des rôles et une authentification sécurisée.
 6. Mettre en place un suivi des réservations en temps réel.
-

Fonctionnalités Principales

1. Gestion des Utilisateurs

- **Rôles :**
 - **Client :** Peut réserver des vols et gérer ses réservations.
 - **Opérateur :** Peut ajouter, modifier ou supprimer des missions et lives.
 - **Administrateur :** Supervise les utilisateurs, les réservations, les missions et lives.
- **Authentification Sécurisée :**
 - Login/mot de passe.
 - 2FA (Authentification à Deux Facteurs) / ~ SSO
- **Permissions :**

Le projet utilise les Voters Personnalisé pour gérer les permissions sur les missions :

- Un opérateur peut uniquement modifier ou supprimer ses propres missions.
- L'admin a un accès complet pour modifier ou supprimer n'importe quelle mission.
- Les Voters sont appliqués sur les actions d'édition et de suppression des missions.

Les Voters permettent d'assurer une gestion granulaire des permissions et d'éviter qu'un opérateur modifie les missions d'un autre.

2. Réservation de Vols Spatiaux

Exploration des Missions :

- Liste des vols disponibles (ex: Starship, Falcon 9, Dragon).
- Filtres par destination, date et prix.

Détails des Missions :

- Destination (orbite terrestre, Lune, Mars, etc.).
- Mission Scientifique/touristique (ex: objectif de la mission, activité, guide présent)
- Prix par siège.
- Date et durée du vol.
- Description détaillée avec images et vidéos.

Système de Réservation :

- Sélection du vol et choix du nombre de places.
- Simulation de paiement via une API (Stripe).

3. Suivi des Missions en Temps Réel

Sur le site son présent plusieurs compte à rebours sur le public et sur le dashboard pour indiquer le temps restant jusqu'au lancement des missions.

4. Gestion des Entités et Relations

Entités :

1. **Booking**
2. **Mission** (classe parent) :
 - **ScientificMission** (héritage)
 - **TouristMission** (héritage)
3. **Notification**
4. **Payment**
5. **ResetPasswordRequest**
6. **Seat**
7. **Spaceship** (classe parent) :
 - **Dragon** (héritage)
 - **Falcon9** (héritage)
 - **Starship** (héritage)
8. **User** (classe parent) :
 - **Client** (héritage)
 - **Operator** (héritage)
 - **Admin** (héritage)
9. **VideoStreaming**

5. API

- **API Interne :**
 - Récupération des missions et ses détails : **api/missions** ou **api/missions/{id}**
- **API Externe :**
 - **API SpaceX :** Récupération des informations des missions.
 - **API de paiement :** Simulation de paiements avec Stripe.
 - **Api YouTube :** Récupération des vidéos en live de la NASA.

6. Espace Administrateur

- Gestion des utilisateurs.
- Gestion des missions et réservations.
- Gestion des lives.

7. Documentation

- **Cahier des charges .**
- **Schéma de la base de données.**
- **Fixtures** pour initialiser les données.
- **Guide d'installation** ([README.md](#)) avec :
 - Instructions pour l'installation et le démarrage du projet.
 - Comptes de tests.
 - Processus de validation.

8. Tests et CI/CD

- **Tests :**
 - **Test unitaire :** Test pour créer un utilisateur, une réservation et une mission
 - **Test fonctionnel :** Simulation d'une création d'utilisateur avec la bdd
- **CI/CD :**
 - Pipeline d'intégration continue (CI) :
 - Exécution des tests unitaires.
 - Analyse statique (PHPStan).
 - PHP-CS-FIXER pour le style de code

9. Pages Différentes

1. Page d'accueil
2. Les 3 prochaines missions
3. À propos de ICanFly
4. Liste des missions
5. Détails d'une mission
6. Liste des lives/des directs
7. Détails du live/du direct
8. Page partenaires avec liste des lancements
9. Détails du lancement

10. Page de connexion/inscription
 11. Tableau de bord utilisateur
 12. Profil utilisateur
 13. Liste des missions côté BO
 14. Détails d'une mission côté BO
 15. Page ajouter une mission
 16. Page modifier une mission
 17. Page listing des réservations de l'utilisateur
 18. Page gestion des réservation
 19. Page détail d'une réservation
 20. Page ajouter une réservation
 21. Page modifier une réservation
 22. Page gestion des lives
 23. Page détail d'un live
 24. Page ajouter un live
 25. Page modifier un live
 26. Page gestion des utilisateurs
 27. Page détail d'un utilisateur
 28. Page ajouter un utilisateur
 29. Page modifier un utilisateur
-

Points Bonus

Temps Réel et Asynchrone

- **Asynchrone :**
Pagination des réservations côté BO
- **Notifications :**
Réception de notification via l'adresse mail lors de l'inscription et le changement de mot de passe de l'utilisateur.
- **Temps Réel :**
Suivi des lancements, missions en direct.

Commandes personnalisées

Une commande personnalisée pour envoyer des rappels de mission :

Cette commande est exécutée automatiquement chaque jour à 08h via un cron job. Elle vérifie si des missions commencent dans les 24 heures suivantes et, si c'est le cas, elle envoie une notification aux utilisateurs ayant des réservations pour leur rappeler que leur mission approche et qu'ils doivent se préparer pour le décollage.