

Documentation du Projet AIR PARADISE - Chatbot de Réservation de Vols

1. Présentation Générale

AIR PARADISE, compagnie aérienne spécialisée dans les vols aux États-Unis, a développé un chatbot intelligent pour faciliter la réservation de vols. Ce projet combine:

- **Laravel** pour la structure backend et la gestion des données
- **JavaScript** pour l'interface chatbot interactive
- **FAISS** (Facebook AI Similarity Search) et **XGBoost** pour la recherche sémantique de villes et la prédiction des prix

2. Architecture Technique

2.1 Technologies Utilisées

Composant	Technologie	Rôle
Frontend	HTML/CSS/JavaScript	Interface utilisateur et chatbot
Backend	Laravel (PHP)	Gestion des données, routage, logique métier
API ML	Flask (Python)	Recherche de villes et prédiction de prix
Base de données	MySQL (via Laravel)	Stockage des réservations et historiques
Modèles ML	FAISS, XGBoost, Sentence Transformers	Recherche sémantique et prédiction

2.2 Flux de Données

Utilisateur → Chatbot JS → API Laravel → (optionnel) API Flask → Base de données

3. Composants Principaux

3.1 Système de Chatbot (JavaScript)

Fonctionnalités principales :

- **Gestion de conversation** avec suivi du contexte (bookingInfo)
- **Recherche intelligente** (Correction automatique) des villes via FAISS
- **Prédiction des prix** en temps réel
- **Processus de réservation** en 4 étapes

Fonctions principales :

javascript

Gestion des messages

- addBotMessage() // Ajoute un message du bot
- addUserMessage() // Ajoute un message utilisateur

Logique métier

- processUserInput() // Traite les entrées utilisateur
- simulateFlightSearch() // Simule la recherche de vols
- fetchCitySuggestion() // API FAISS pour les villes
- fetchPriceSuggestion() // API XGBoost pour les prix

3.2 API Machine Learning (Flask/Python)

Deux endpoints principaux :

1. **POST /search** (FAISS)

- Recherche sémantique de villes
- Utilise le modèle all-MiniLM-L6-v2 pour les embeddings
- Renvoie les villes les plus proches de la requête

2. **POST /searchPrice** (XGBoost)

- Prédit le prix d'un vol basé à l'aide du modèle XGBoost sur :

- Villes de départ/d'arrivée
- Distance
- Compagnie aérienne
- Date et heure

3.3 Backend Laravel

Routes principales :

- **POST /flight-history** : Stocke l'historique des recherches
- **POST /process-booking** : Traite les paiements et crée les réservations

Modèles Eloquent :

- FlightHistory : Historique des recherches
- Booking : Réservations confirmées

4. Flux Utilisateur

1. Initialisation :

- Le chatbot demande la ville de départ
- Correction automatique via FAISS si nécessaire

2. Recherche :

- Saisie de la destination
- Prédiction du prix via XGBoost
- Sélection de la date via formulaire

3. Résultats :

- Affichage des vols disponibles (simulés)
- Sélection d'un vol

4. Paiement :

- Saisie des informations personnelles
- Validation via Laravel
- Confirmation de réservation

5. Gestion d'erreurs :

- Si aucune ville trouvée → Nouvelle tentative
- Si aucun vol disponible → Réinitialisation

5. Modèles de Machine Learning

5.1 FAISS pour la Recherche de Villes

- **Modèle** : all-MiniLM-L6-v2 (Sentence Transformers)
- **Données** : Liste de 99 villes américaines
- **Fonctionnement** :

5.2 XGBoost pour la Prédiction de Prix

- **Features** :
 - Distance entre villes
 - Compagnie aérienne
 - Mois, jour de la semaine
 - Heure de départ
- **Pré-traitement** :
 - Encodage des catégories (OneHotEncoder)
 - Normalisation des données

6. Installation et Déploiement

6.1 Pré-requis

- PHP 8.0+, Composer
- Python 3.8+, pip
- Node.js
- MySQL

6.2 Configuration

1. Backend Laravel :

- `composer create-project laravel/laravel air-paradise-chabot`
- **php artisan serve**

2. API Flask :

- `pip install faiss-cpu flask numpy`
- `source venv/bin/activate`
- `python -m venv venv`
- `pip install sentence_transformers`
- `pip install xgboost`
- **`cd faiss_python/ python city_embedding.py`**

3. Frontend :

- `npm install`

npm run dev

7. Améliorations Possibles

1. **Intégration réelle** d'une IA pour une communication plus interactive
2. **Amélioration du modèle** de prix avec plus de données grace a la base de donnée contenant les historiques
3. **Système de recommandation** basé sur l'historique
4. **Interface admin** pour gérer les données
5. **Système d'authentification**

8. Conclusion

Ce projet démontre une intégration efficace entre :

- Un frontend interactif avec chatbot
- Un backend Laravel robuste
- Des modèles de machine learning performants

L'architecture permet une extension facile pour ajouter de nouvelles fonctionnalités ou intégrer des données réelles de compagnies aériennes.

