

# **RAPPORT DE PROJET : AIR PARADISE**

**(4 SPE IA)**

## **Noms des membres du groupe :**

**- TEMGOUA KENNE CARINE**

**- KENFACK ARIOL**

**- FONKUI WILLIAM**

## **INTRODUCTION**

Air Paradise est un projet de chatbot intelligent conçu pour faciliter la recherche et la réservation de vols internes aux États-Unis. L'objectif était de proposer une interface intuitive combinant intelligence artificielle, traitement du langage naturel, modélisation de prix et expérience utilisateur fluide via une application Streamlit.

Le projet s'appuie sur deux piliers :

- La prédiction de prix de vols
- L'automatisation du dialogue avec les utilisateurs via un chatbot intelligent

## **2. OBJECTIFS DU PROJET**

- Développer un modèle IA capable de prédire le prix d'un vol en fonction de plusieurs critères (aéroports, horaire, jour, etc.)
- Mettre en place un chatbot permettant de guider l'utilisateur dans sa recherche de vol, tout en interagissant de manière naturelle.
- Intégrer une interface utilisateur ergonomique permettant de consulter, estimer et réserver des vols en quelques clics.

### **3. TECHNOLOGIES ET BIBLIOTHEQUES UTILISEES**

#### ❖ **Backend / API**

- **FastAPI** : pour créer une API REST performante et rapide
- **Pydantic** : pour valider les schémas de données à la volée
- **joblib** : pour charger efficacement le modèle de prédiction
- **requests** : pour appeler les endpoints depuis l'application frontend

#### ❖ **Machine Learning**

- **scikit-learn** : pour entraîner le modèle de régression de prix
- **category\_encoders (HashingEncoder)** : pour encoder les codes IATA
- **pandas** : pour la manipulation des données tabulaires

#### ❖ **Traitement du langage**

- **Google Generative AI (Gemini)** : pour extraire les informations de vol dans un langage naturel
- **langdetect** : pour détecter automatiquement la langue (FR / EN)
- **dateparser** : pour interpréter des expressions comme "demain matin" ou "8h"

#### ❖ **Frontend / Interface utilisateur**

- **Streamlit** : pour créer une application web interactive rapidement
- **streamlit\_extras** : pour des composants comme switch\_page\_button
- **FPDF / qrcode** : pour générer un billet d'avion PDF avec QR code embarquable

## **4. MODELE DE PREDICTION DE PRIX**

### **4.1 Données et prétraitement**

Le modèle s'appuie sur :

- Le code IATA de départ et d'arrivée
- La date, heure, durée et jour de la semaine du vol

L'heure est transformée en "période" (matin / journée / soir) et encodée par `get_dummies`, tandis que les codes IATA sont encodés avec un `HashingEncoder`.

### **4.2 Modèle ML**

Le processus complet de modélisation a été mené à travers une série de notebooks Jupyter structurés :

- **01\_exploration.ipynb** : analyse exploratoire des données (répartition des vols, plages horaires, distances, etc.)
- **Select\_variables.ipynb** : sélection des variables les plus influentes sur le prix
- **02\_preprocessing.ipynb** : normalisation, encodage des données, gestion des valeurs manquantes
- **03\_model\_training.ipynb** : entraînement du modèle avec régression linéaire ou autre modèle supervisé, validation croisée
- **04\_mapping.ipynb** : correspondance et enrichissement des aéroports avec labels lisibles

Le modèle est ensuite exporté au format `.pkl` pour être intégré à l'API de prédiction utilisée dans l'application Streamlit.

Ce pipeline garantit reproductibilité, propreté des données, et performance optimale. Un modèle de régression a été entraîné sur ces données. Il est appelé depuis une API à chaque fois qu'un utilisateur demande une estimation.

## **5. CHATBOT INTELLIGENT**

### **5.1 Approche**

Le chatbot repose sur **Gemini AI (Google)** qui permet d'extraire dynamiquement des entités d'une requête utilisateur (origine, destination, date, heure). L'IA comprend les formulations naturelles, détecte les intentions, et s'adapte à la langue.

### **5.2 Validation**

Des contrôles sont appliqués sur les réponses :

- Aéroports valides et différents
- Plage horaire entre 06:00 et 22:00
- Date cohérente

En cas d'informations manquantes, le chatbot relance automatiquement l'utilisateur.

## **6. INTERFACE UTILISATEUR STREAMLIT**

L'application est divisée en plusieurs pages :

### **❖ home.py**

Page d'accueil accueillante avec fond dégradé, logo et boutons pour accéder au chatbot ou au formulaire. Elle présente les fonctionnalités du projet.

### **❖ chatbot.py**

- Chat conversationnel
- Interaction IA + prédiction prix
- Affichage des messages dans st.chat\_message
- Confirmation avant réservation

#### ❖ **interface.py**

- Formulaire manuel pour sélectionner un vol
- Appel à l'API de prédiction
- Redirection vers la réservation

#### ❖ **reservation.py**

- Formulaire utilisateur (email, téléphone, passeport, etc.)
- Génération de billet PDF avec QR code
- Envoi par email avec SMTP

#### ❖ **Sidebar personnalisée**

- Choix de langue dynamique
- FAQ contextuelle selon le mot-clé sélectionné
- Bouton de réinitialisation de la conversation

## **7. ENREGISTREMENT DES CONVERSATIONS**

Toutes les interactions sont enregistrées dans le fichier tests/conversations.csv, permettant :

- L'audit des conversations
- La possibilité d'analyse statistique ou NLP plus tard

## **8. NETTOYAGE DU PROJET**

Plusieurs fichiers préparatoires ont été retirés car non utilisés dans la version finale :

- workflows.py
- api.py
- messages.py
- rag\_engine.py

## **CONCLUSION**

Air Paradise combine IA, UX et automatisation pour offrir un système de réservation aérienne innovant. Il intègre :

- Une estimation intelligente de prix
- Une interface conversationnelle naturelle
- Une réservation numérique avec billet PDF et QR code

Le projet est fonctionnel, complet, et respecte le cahier des charges.

### **PISTES D'AMELIORATION (PERSPECTIVES)**

- Intégration d'une base de données (PostgreSQL ou MongoDB)
- Chat vocal (speech-to-text + text-to-speech)
- Dashboard admin pour suivre les réservations et feedbacks
- Analyse automatique des logs avec modèles NLP pour détecter les intentions ou problèmes récurrents