

MINI PROJET

Sujet : conception et développement d'une
Application web de gestion de vol

Réalisée par :

- ELMENGOUG Fadoua
- RHAZOULI Safaa
- GUESSOUS ikram
- KAEDOUDI souad

Encadré par :

- Mr. ELBAGHAZAOUI Bahaa Eddine

Table of Contents

Remerciement	4
Introduction générale :	5
Chapitre1 : Contexte Général du mini projet	6
1 Centralisation des données critiques :	7
2 Protection et gestion des accès :	7
3 Rationalisation des opérations aériennes :	7
Chapitre 3 : Cadre générale du projet et conception	9
4 Présentation du Projet :	10
4.1 Contexte	10
4.2 Architecture de l'application :	10
4.3 Diagramme de classe :	11
4.4 Diagrammes de séquences :	12
5 . Environnements logiciels :	16
5.1 Outils logiciels front end :	16
5.2 Outils logiciels backend :	16
1.1 Technologies utilisées :	17
6 Choix technologiques :	23
6.1 . Choix du React :	23
6.2 Choix du plateforme Java avec Spring Boot:	23
Chapitre 3 : Réalisation	26
Conclusion :	32

Tableau de figures:

Figure 1 architecture globale	11
Figure 2 diagramme de class	12
Figure 3 diagramme de séquence 1	14
Figure 4 diagramme de séquence 2	15
Figure 5 logo 1	16
Figure 6 sts logo	17
Figure 7 Logo react.....	18
Figure 8 mermaid logo	18
Figure 9 :PostgreSQL LOGO	19
Figure 10 logo de taiga	21
Figure 11 Docker logo	22
Figure 12 : postman logo.....	22
Figure 13 HOME PAGE.....	27
Figure 14 ABOUT PAGE	27
Figure 15 MY BOOKING PAGE	28
Figure 16 BOOKING PAGE	28
Figure 17 CONTACT US PAGE	29
Figure 18 LOGIN PAGE.....	29
Figure 19 LOGOUT PAGE	30

Remerciement

Je tiens à exprimer mes sincères remerciements à Pr. BE. Elbaghazaoui, notre encadrant, pour son accompagnement, ses conseils avisés et son soutien tout au long de la réalisation de ce mini-projet. Sa disponibilité, son expertise et ses orientations précieuses ont été déterminantes pour mener à bien ce travail.

Votre encadrement a non seulement enrichi nos connaissances académiques, mais a également renforcé nos compétences pratiques dans le domaine du génie logiciel.

Merci pour votre investissement et votre engagement envers la réussite de vos étudiants.

Avec toute ma gratitude,

Introduction générale :

Dans un contexte où l'aviation joue un rôle clé dans le transport international, la gestion efficace et centralisée des informations liées aux vols aériens devient essentielle. Les systèmes de gestion d'inventaire pour les compagnies aériennes doivent répondre à des exigences de performance, de sécurité, et d'accessibilité, tout en couvrant divers aspects tels que la planification des vols, la gestion des passagers, des avions, des équipages et des aéroports.

Le présent mini-projet, réalisé dans le cadre du cursus à l'ENSA-Beni Mellal pour l'année universitaire 2024/2025, a pour objectif de concevoir et de développer une application complète et sécurisée pour la gestion d'inventaire des vols aériens. Cette solution, bâtie sur une architecture moderne utilisant Java et Spring Boot, mettra en œuvre une API REST sécurisée par des jetons JWT pour garantir l'authentification et l'autorisation des utilisateurs.

Outre la gestion des données critiques, le projet intègre les bonnes pratiques en matière de persistance des données avec Spring Data JPA et PostgreSQL, ainsi qu'une chaîne CI/CD pour assurer l'intégration et le déploiement continus. La conteneurisation avec Docker permettra une flexibilité accrue lors de la mise en production, tandis qu'un front-end interactif, développé avec des technologies modernes telles que React ou Vaadin, assurera une expérience utilisateur fluide et intuitive.

Ce rapport détaille les étapes clés du projet, depuis l'analyse et la conception jusqu'à l'implémentation et la validation, tout en mettant en lumière les principaux défis rencontrés et les solutions apportées.

Chapitre1 : Contexte Général du mini projet

1 Centralisation des données critiques :

- Le projet vise à regrouper et structurer l'ensemble des informations liées aux vols, avions, passagers, réservations, équipages, et aéroports. Cela permettra une organisation cohérente et un accès efficace pour tous les utilisateurs concernés.

2 Protection et gestion des accès :

- Une priorité est donnée à la sécurité des informations en intégrant des mécanismes pour vérifier l'identité des utilisateurs et gérer leurs droits d'accès, tout en préservant la confidentialité des données sensibles.

3 Rationalisation des opérations aériennes :

- Le système permettra d'améliorer l'efficacité des processus, tels que la gestion des vols et des ressources, afin de minimiser les erreurs, simplifier la coordination et répondre aux besoins opérationnels de manière plus fluide.

Chapitre 3 : Cadre générale du projet et conception

4 Présentation du Projet :

4.1 Contexte

Dans le cadre de la gestion des opérations aériennes, il est essentiel de disposer d'un système permettant de regrouper et de structurer les informations essentielles liées aux différents aspects de l'aviation. Cela inclut les données relatives aux vols, aux avions, aux passagers, aux réservations, aux équipages, ainsi qu'aux infrastructures aéroportuaires. Une organisation claire et centralisée de ces informations est fondamentale pour garantir la fluidité et l'efficacité des opérations.

Par ailleurs, la sécurité des données et des accès joue un rôle crucial. Les informations sensibles, telles que les données personnelles des passagers ou les détails opérationnels des vols, nécessitent une protection adéquate. Un mécanisme robuste pour gérer les autorisations et assurer la confidentialité est indispensable afin de préserver la confiance des utilisateurs et de respecter les réglementations en vigueur.

Enfin, la gestion optimale des processus liés aux vols et aux ressources associées est un levier clé pour améliorer les performances globales. Une planification efficace, associée à un suivi en temps réel, permet de réduire les erreurs, d'optimiser l'utilisation des ressources, et d'assurer une réponse rapide aux imprévus. Ce système vise donc à transformer les défis opérationnels en opportunités d'amélioration continue.

4.2 Architecture de l'application :

L'architecture globale de notre application se résume comme suit :

Front-end, qui couvre la couche présentation à base React JS. Elle regroupe les interfaces Homme-Machine et les scripts assurant la communication avec la partie backend, en se basant sur le protocole HTTP (HyperText Transfer Protocol). Le format d'échange est JSON (JavaScript Object Notation).

Back-end, qui couvre le serveur de l'application enveloppée par la technologie NodeJS permettant d'exécuter du Javascript côté serveur. Grâce à son fonctionnement non bloquant, il permet de concevoir des applications en réseau performantes, telles qu'un serveur web, une API

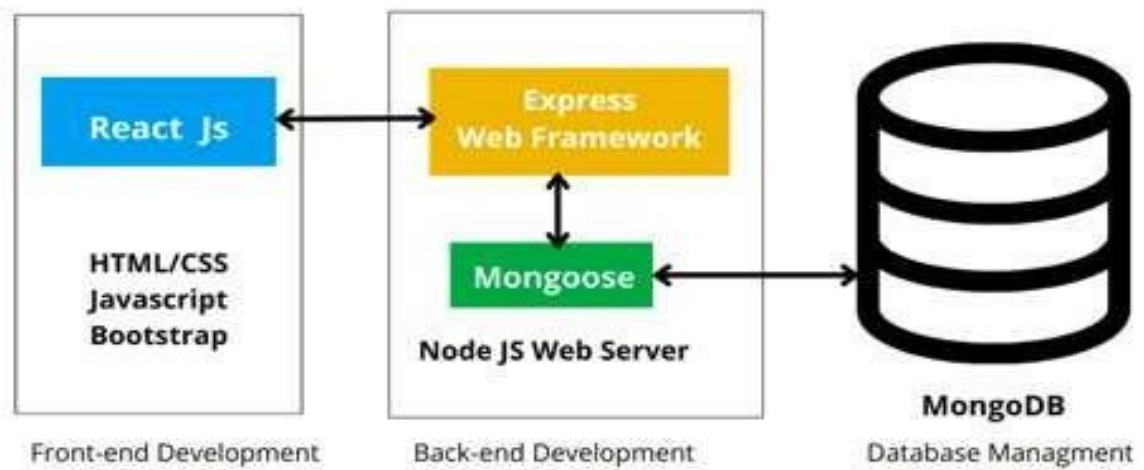


Figure 1 architecture globale

4.3 Diagramme de classe :

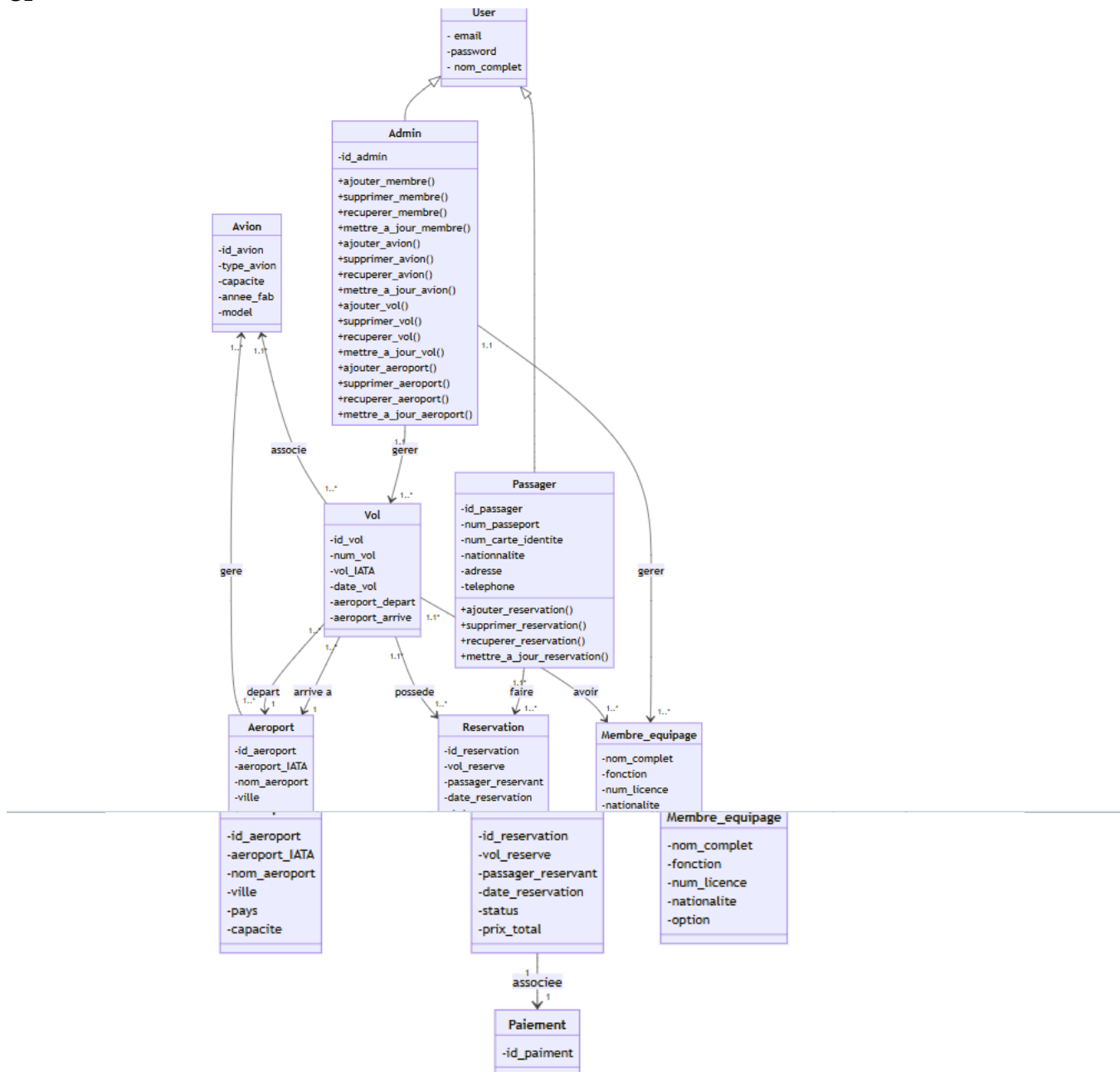


Figure 2 diagramme de class

4.4 Diagrammes de séquences :

Figure 4Diagramme de séquence

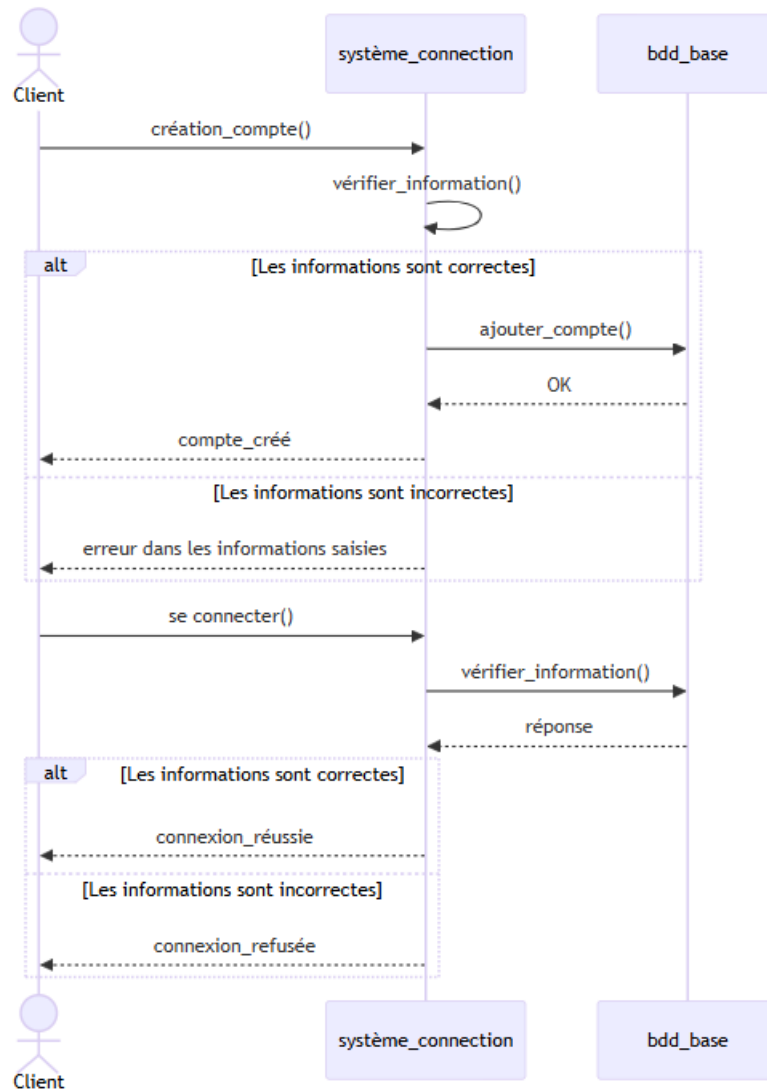


Figure 3 diagramme de séquence 1

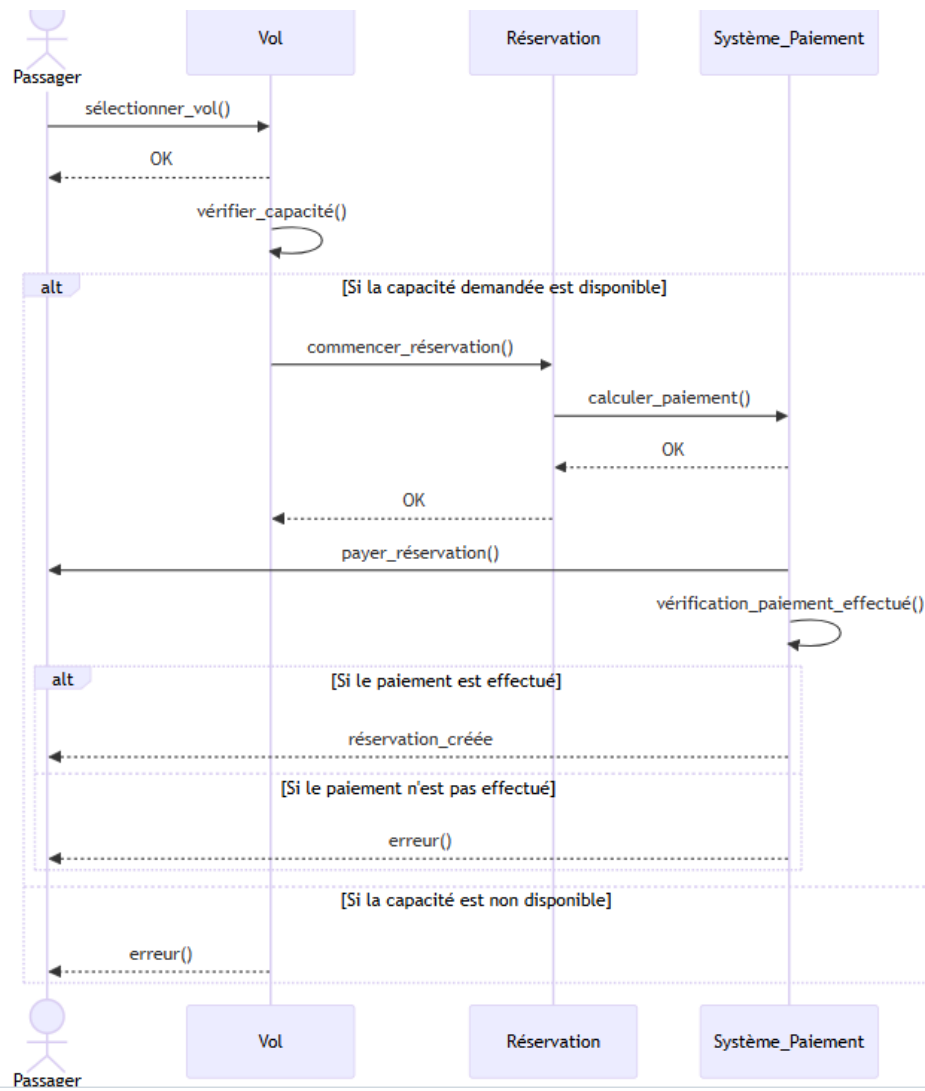


Figure 4 diagramme de séquence 2

5 . Environnements logiciels :

Nous détaillons dans cette partie les logiciels et les langages utilisés pour le développement de notre application web.

5.1 Outils logiciels front end :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré



Figure 5 logo 1

5.2 Outils logiciels backend :

STS (Spring Tool Suite) est un environnement de développement intégré (IDE) conçu pour faciliter le développement d'applications basées sur le framework Spring. Il est basé sur Eclipse et inclut un ensemble de fonctionnalités supplémentaires spécifiques à Spring, ce qui en fait un outil idéal pour les développeurs Java travaillant sur des projets Spring.



Figure 6 sts logo

1.1 Technologies utilisées :

- ReactJS:
 - React est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage

- React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS.
- La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité .

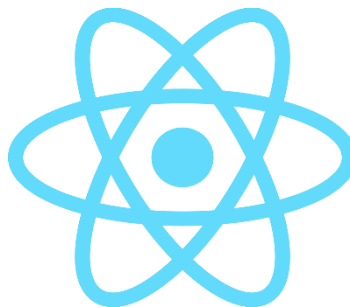


Figure 7 Logo react

- Mermaid :
 - Mermaid est un outil de création de diagrammes et de visualisations, basé sur une syntaxe simple en langage Markdown. Il permet de générer des diagrammes dynamiques et interactifs à partir de texte brut, rendant le processus de création rapide, intuitif et efficace, notamment pour les développeurs et les équipes techniques.



Figure 8 mermaid logo

- PostgreSQL :
 - PostgreSQL (souvent abrégé en **Postgres**) est un système de gestion de bases de données relationnelles (SGBDR) open source. Il est reconnu pour sa puissance, sa robustesse et ses fonctionnalités avancées, notamment :
 - Prise en charge des transactions ACID.
 - Requêtes SQL complexes et optimisation des performances.
 - Extensibilité avec des fonctions définies par l'utilisateur et des types de données personnalisés.
 - Gestion des données géospatiales via l'extension PostGIS.



Figure 9 :PostgreSQL LOGO

- Git /GitHub
 - Grâce à Git, les développeurs peuvent ****pousser**** (push) leurs modifications de code vers des référentiels distants et ****tirer**** (pull) les mises à jour effectuées par d'autres collaborateurs. Cette fonctionnalité de push et pull permet une synchronisation facile des contributions de chaque membre de l'équipe, assurant ainsi que tout le monde travaille avec la version la plus récente du code.
 - GitHub est une plateforme de développement collaboratif basée sur Git, permettant aux développeurs de stocker, de gérer et de suivre les modifications de leur code source en ligne. Créée en 2008, GitHub a révolutionné la manière dont les développeurs collaborent sur des projets logiciels en offrant des fonctionnalités supplémentaires pour faciliter le travail d'équipe.
 - Dans le cadre de mon projet de stage, GitHub a grandement facilité la collaboration entre moi et mon encadrant, Ouarrak Taha. En utilisant GitHub, nous avons pu partager notre code, suivre les modifications en temps réel, et travailler ensemble de manière efficace malgré la distance. Les fonctionnalités de gestion de projet offertes par GitHub, telles que les pull requests et les issues, ont été essentielles pour maintenir une communication claire et une organisation structurée tout au long du développement de l'application de communication en temps réel.



Figure 10 GitHub logo

- Taiga :

Taiga est une plateforme open source de gestion de projets agile. Elle est conçue pour faciliter la collaboration entre les équipes et gérer efficacement les projets, qu'ils soient basés sur la méthodologie Scrum, Kanban ou une combinaison des deux. Taiga est particulièrement apprécié pour sa simplicité, son interface conviviale et ses fonctionnalités puissantes.



Figure 10 logo de taiga

- Docker :

Docker est une plateforme open source conçue pour automatiser le déploiement, la gestion et l'exécution des applications dans des **conteneurs**. Ces conteneurs sont des environnements isolés, légers et portables qui contiennent tout ce dont une application a besoin pour fonctionner (code, dépendances, bibliothèques, etc.).



Figure 11 Docker logo

- Postman :

Postman est un outil collaboratif puissant utilisé pour le développement, le test, et la documentation des API (Application Programming Interfaces). Il permet aux développeurs d'interagir avec des API RESTful et d'autres types d'API, de les tester efficacement et de les documenter. Disponible sous forme d'application de bureau ou d'extension de navigateur, Postman est largement adopté dans le monde du développement logiciel.



Figure 12 : postman logo

6 Choix technologiques :

Nous justifions, dans ce qui suit, les plus importants nos choix technologiques

6.1 . Choix du React :

React est un effort que Facebook a lancé en 2015, et il a effectivement provoqué une vague sur le marché des Framework hybrides. Quelques années après son introduction sur le marché, il est déjà l'un des plus populaires.

Les avantages de React :

- Jusqu'à 80 % d'une base de code peut être partagée entre les plates-formes, selon la complexité de l'application.
- En plus de la réutilisation du code, il permet de prévisualiser immédiatement les résultats, en plus d'offrir des éléments prêts à être appliqués, ce qui réduit considérablement le temps de développement.
- La fonction de rechargement à chaud permet aux développeurs de voir les modifications apportées au code en quelques secondes, et non en quelques minutes lorsqu'ils utilisent des technologies natives.
- React Native se concentre sur l'interface utilisateur, rendant une interface très réactive.

6.2 Choix du plateforme Java avec Spring Boot:

- **Robustesse et maturité**
 - **Java** est un langage établi, connu pour sa fiabilité et sa stabilité dans les applications d'entreprise.
 - **Spring Boot**, basé sur le framework Spring, est conçu pour simplifier le développement de projets complexes tout en

conservant les meilleures pratiques.

- **Simplicité et productivité**

- **Démarrage rapide** : Spring Boot propose un écosystème prêt à l'emploi avec des configurations par défaut, permettant de se concentrer sur la logique métier.
- **Spring Initializr** : Génère automatiquement la structure du projet et configure les dépendances nécessaires.
- **Moins de configuration** : Grâce à la configuration basée sur des annotations et des conventions plutôt que sur des fichiers XML complexes.

- **Gestion des API REST puissante**

- Spring Boot permet de créer des API REST facilement grâce à des outils comme **Spring MVC**.
- Support natif pour les contrôleurs, les DTO (Data Transfer Objects), et la sérialisation JSON via **Jackson**.
- Documentation intégrée avec **Swagger/OpenAPI** pour simplifier les tests et la communication avec les équipes front-end.

- **Sécurité avancée**

- **Spring Security** est un composant intégré pour gérer l'authentification et l'autorisation.
- Prise en charge de **JWT** pour sécuriser les interactions avec l'API.
- Gestion fine des rôles et permissions, essentielle dans un projet impliquant différents types d'utilisateurs (administrateurs, passagers, équipages, etc.).

- **Intégration avec PostgreSQL**

- **Spring Data JPA** offre une abstraction puissante pour interagir avec des bases de données relationnelles comme PostgreSQL.
- Gestion simplifiée des entités, des relations complexes (one-to-many, many-to-many), et des migrations de bases de données.
- Requêtes personnalisées faciles à écrire en utilisant JPQL ou des méthodes dérivées.

- **Écosystème moderne et évolutif**
 - Compatible avec des technologies modernes comme **Docker** pour la conteneurisation.
 - Intégration native avec des outils de CI/CD comme **Jenkins** et **GitLab CI**.
 - Hautement scalable : idéal pour des systèmes nécessitant une montée en charge progressive.
- **Communauté et support**
 - Spring est l'un des frameworks les plus utilisés au monde, avec une vaste communauté.
 - Documentation complète et nombreuses ressources pour apprendre et résoudre les problèmes.
- **Compatibilité multiplateforme**
 - Fonctionne de manière optimale sur plusieurs systèmes d'exploitation et environnements (Windows, macOS, Linux).
 - Adapté aux déploiements sur site ou dans le cloud (AWS, Azure, GCP).
- **Flexibilité pour le front-end**
 - L'API Spring Boot peut être consommée par différents types de front-end, comme **React**, **Angular**, **Flutter**, ou **Vaadin**, sans besoin de modifications majeures.

Chapitre 3 : Réalisation

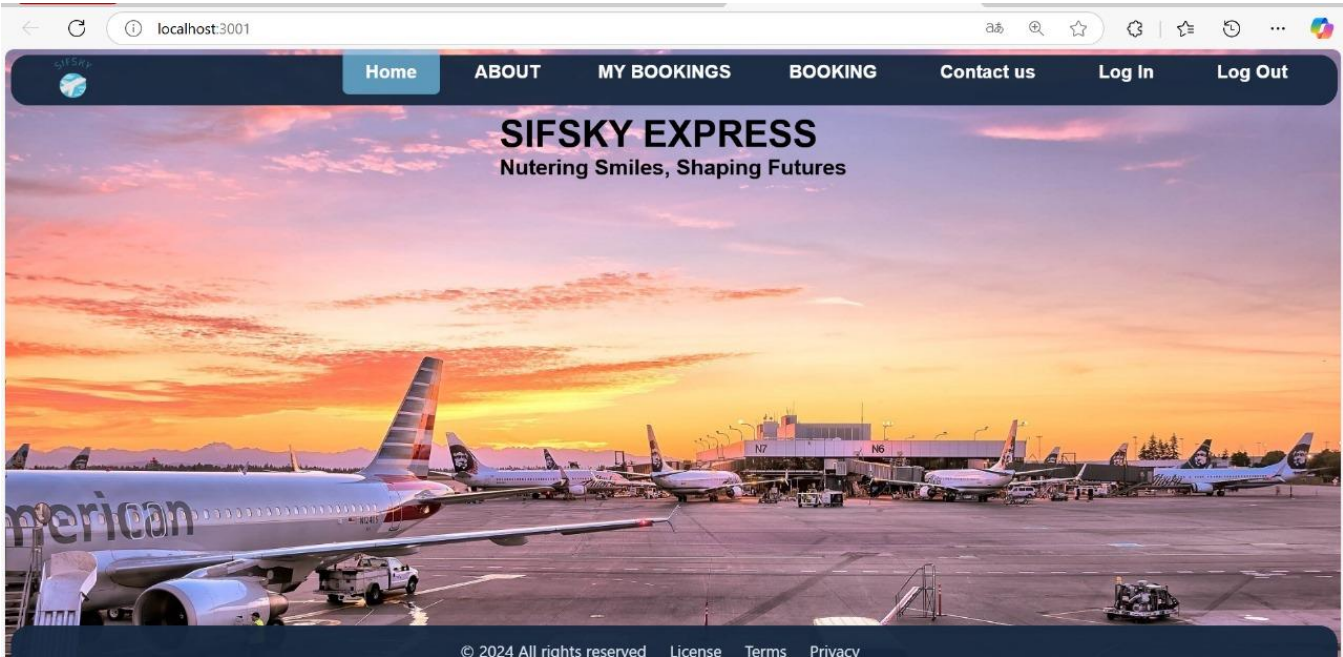


Figure 13 HOME PAGE

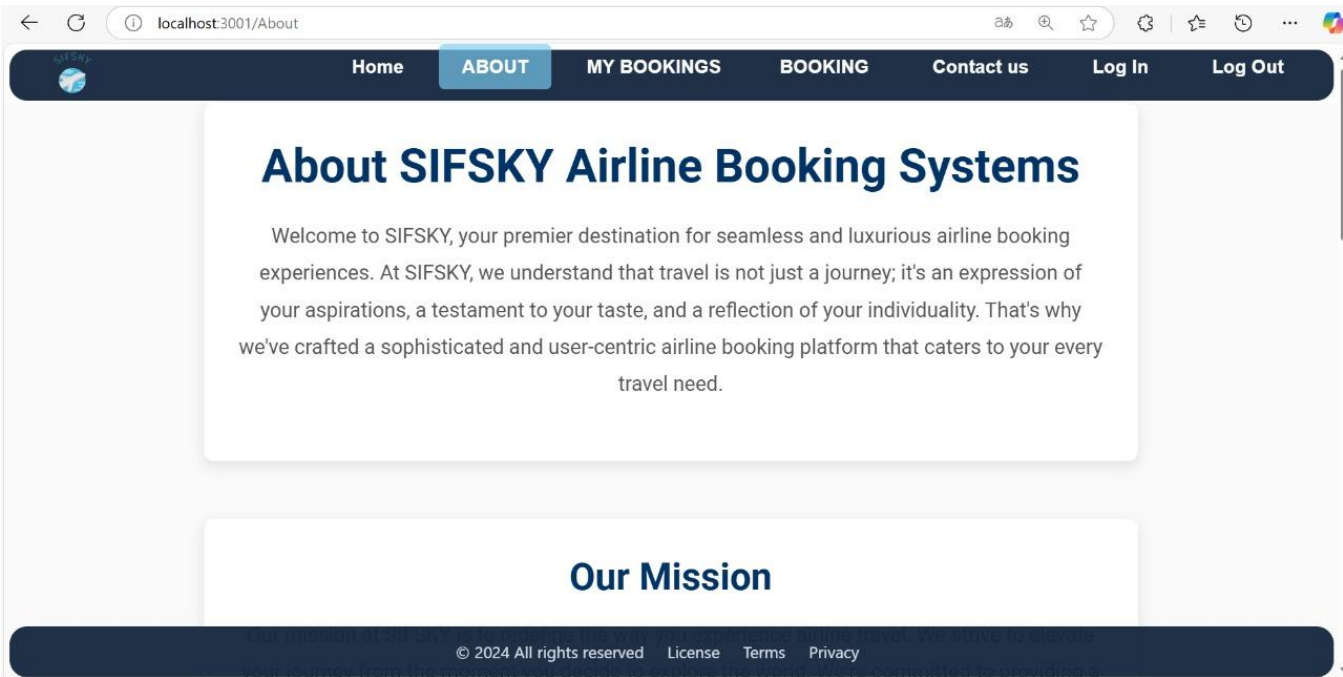


Figure 14 ABOUT PAGE

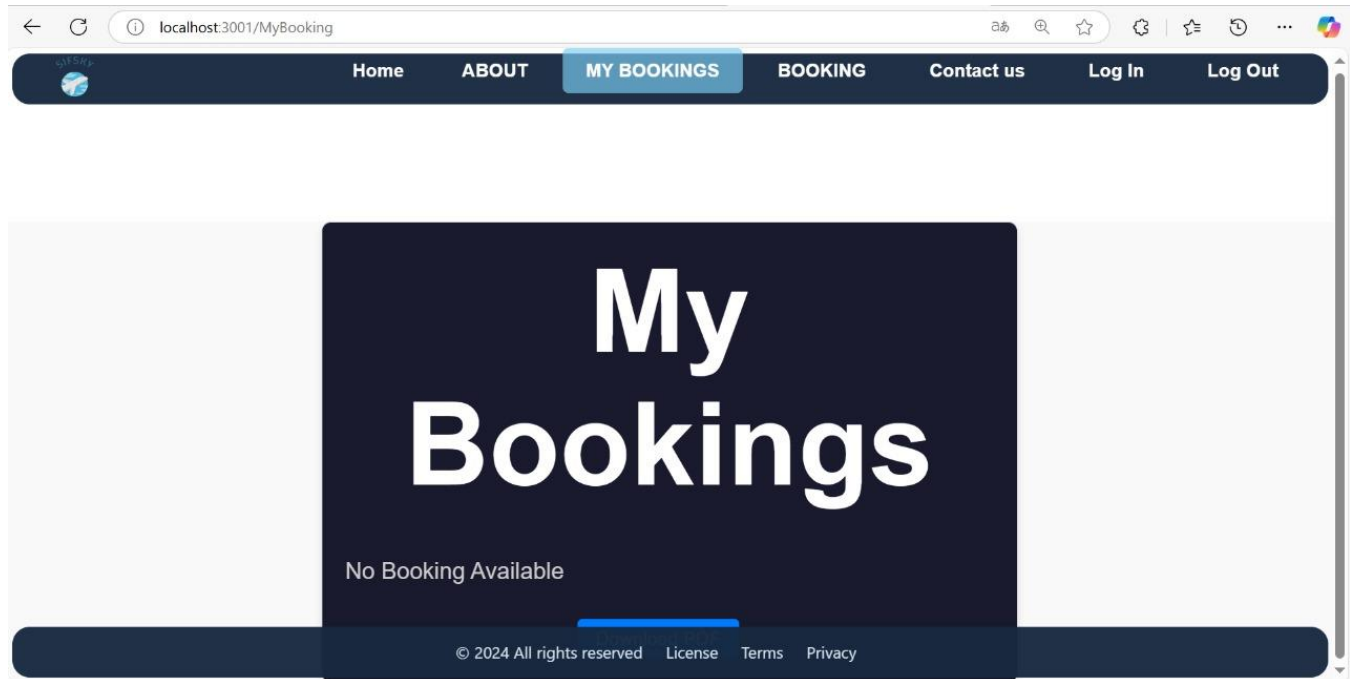


Figure 15 MY BOOKING PAGE

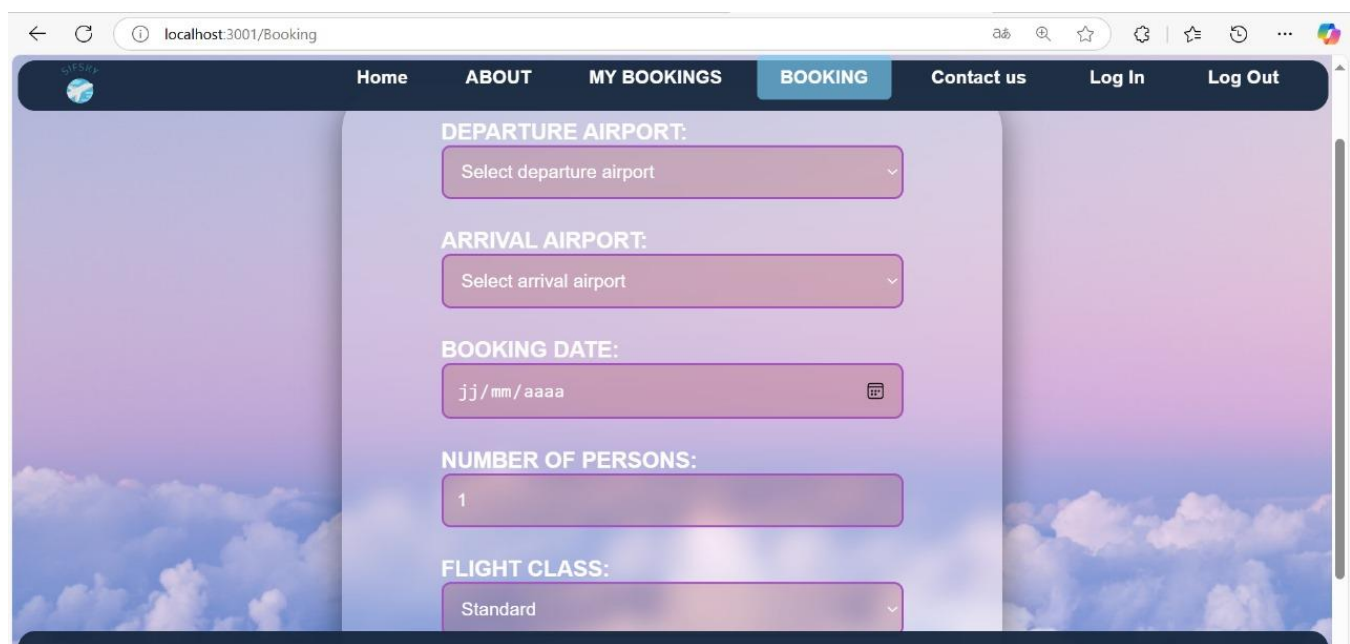


Figure 16 BOOKING PAGE

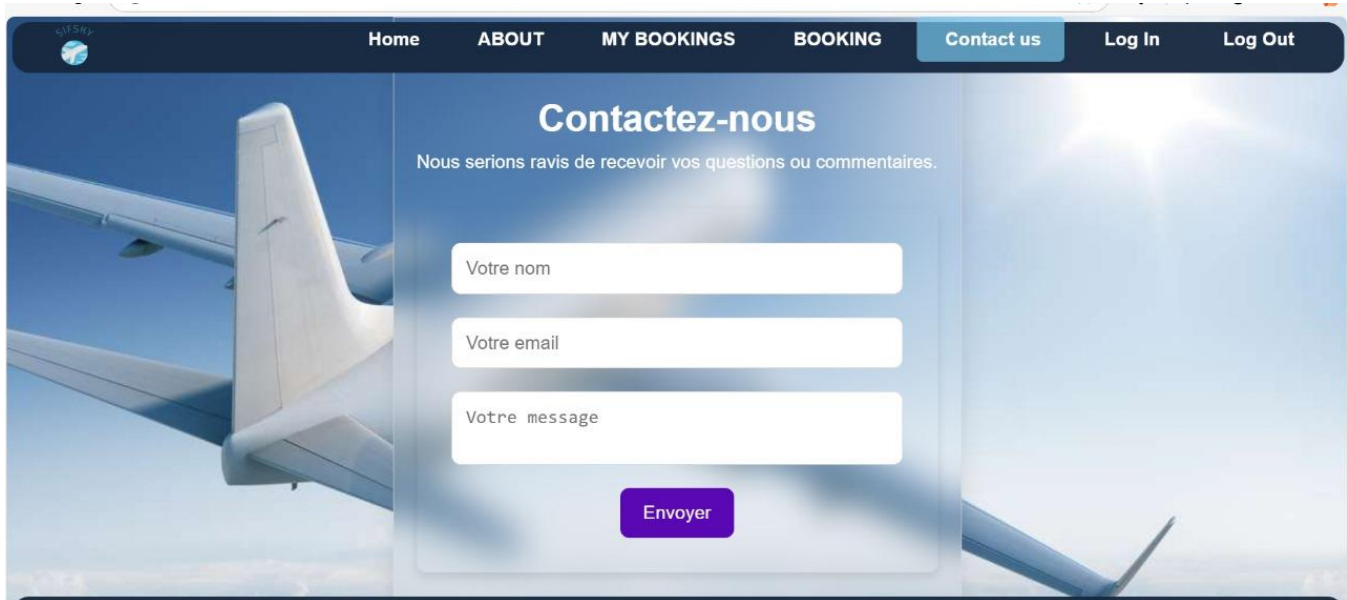


Figure 17 CONTACT US PAGE

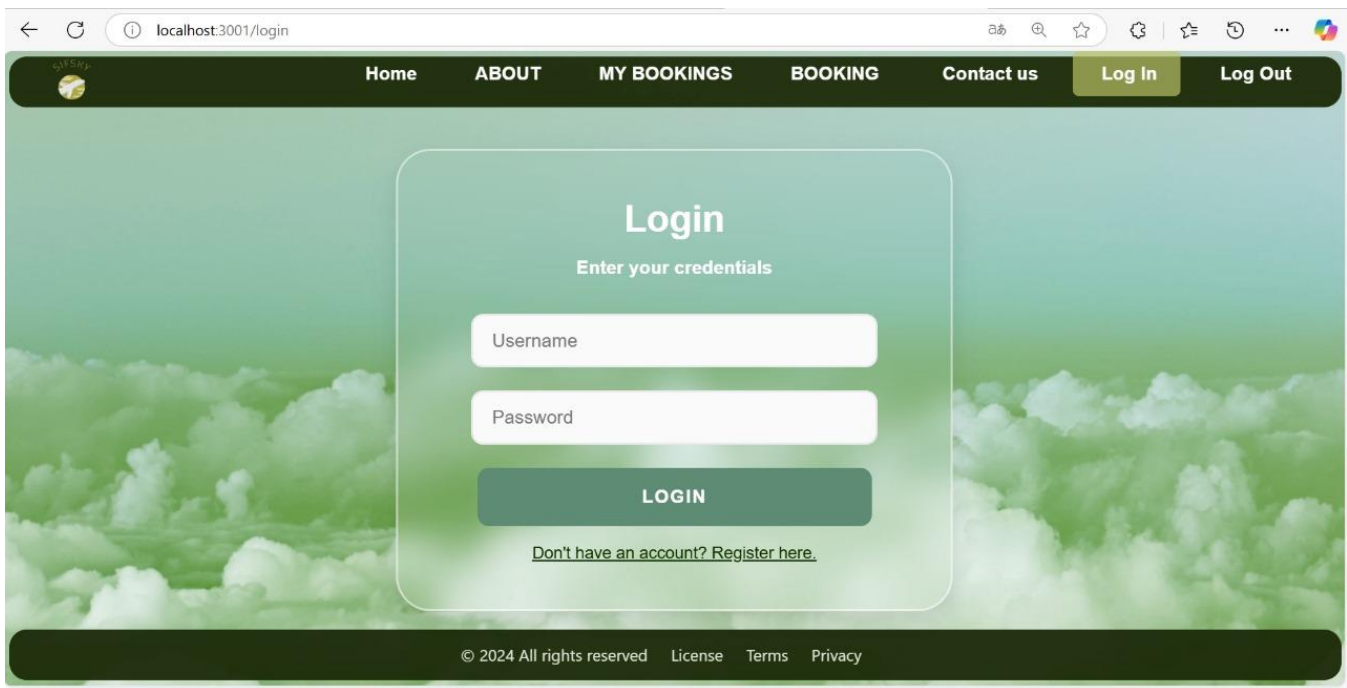


Figure 18 LOGIN PAGE

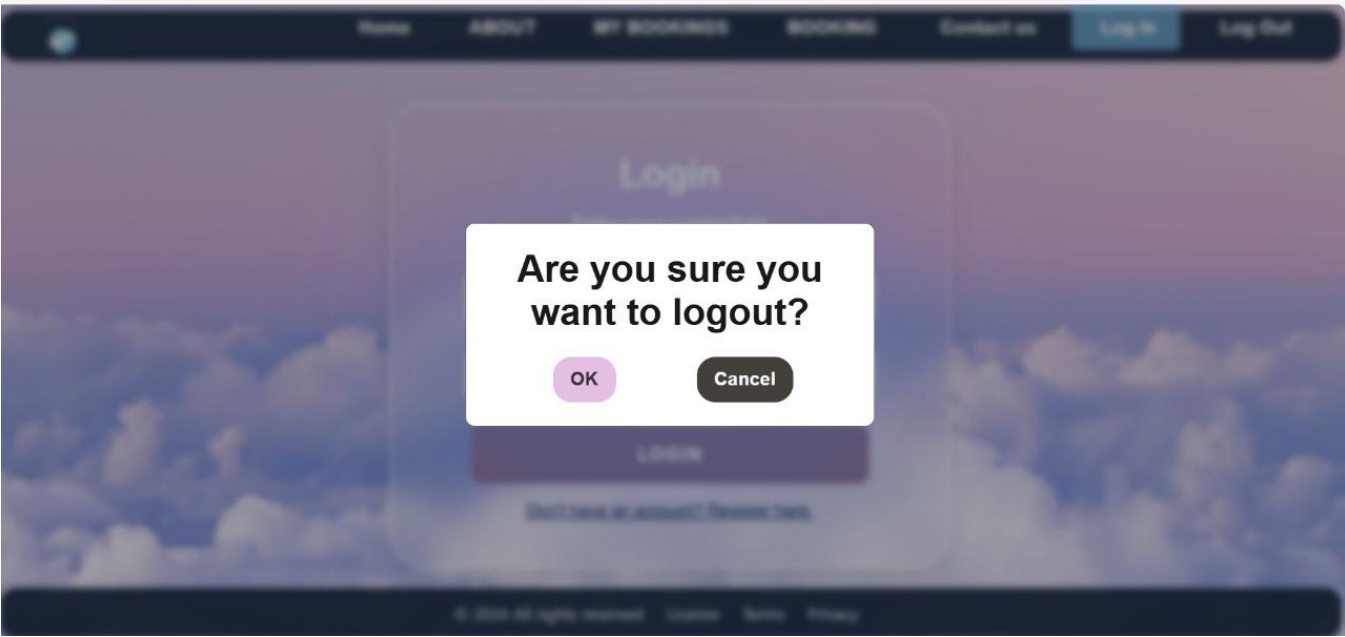
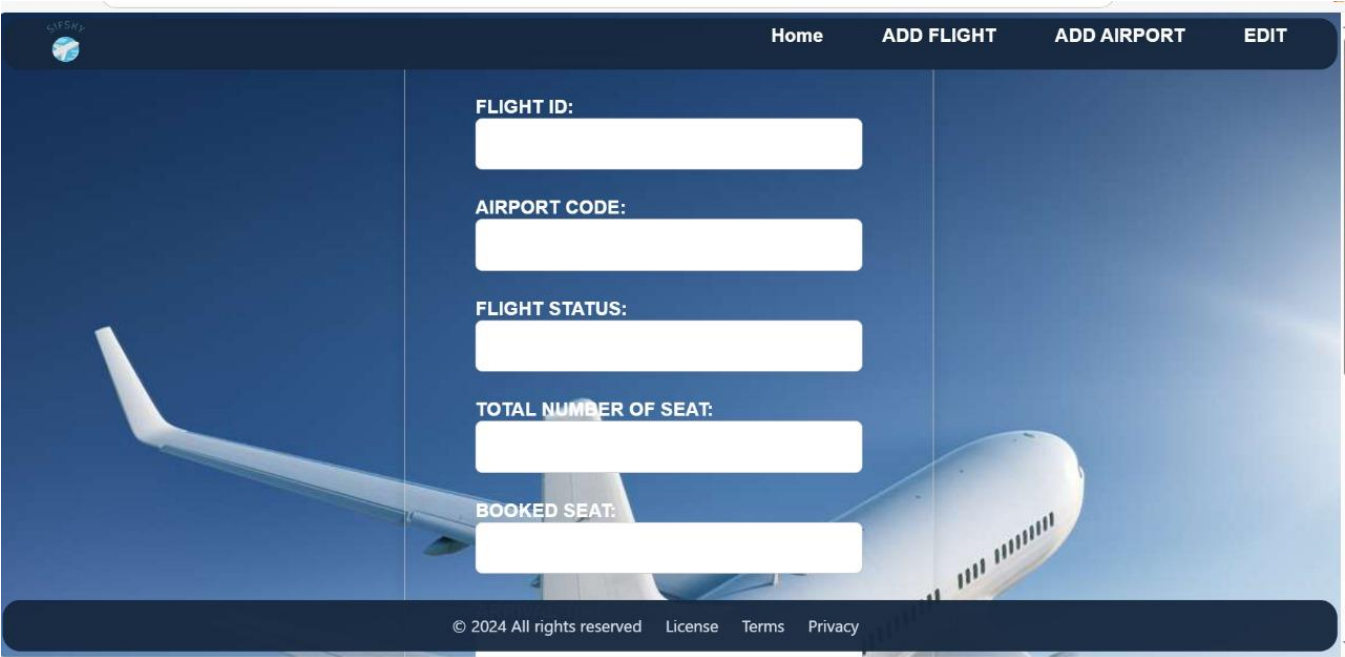

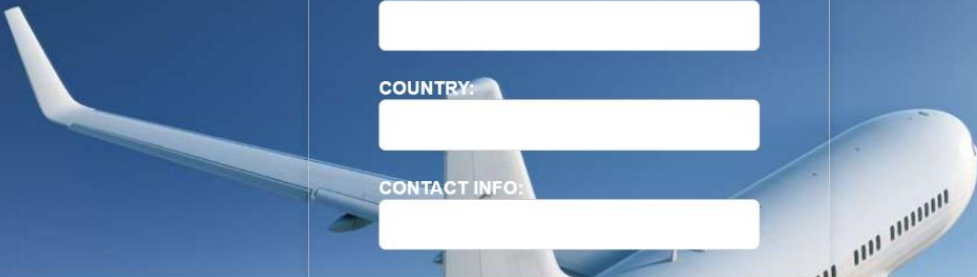


Figure 19 LOGOUT PAGE





[Home](#)[ADD FLIGHT](#)[ADD AIRPORT](#)[EDIT](#)



AIRPORT CODE:


NAME:

CITY:

COUNTRY:

CONTACT INFO:

© 2024 All rights reserved License Terms Privacy



[Home](#)[ADD FLIGHT](#)[ADD AIRPORT](#)[EDIT](#)

JFK	On time	100	100	10.00	14.00	<button>Cancel</button>
-----	---------	-----	-----	-------	-------	-------------------------

Airport Details

Airport Code	Name	City	Country	Contact Info	Actions
JFK	John F. Kennedy International Airport	New York	USA	123-456-7890	<button>Edit</button>

© 2024 All rights reserved License Terms Privacy

Conclusion :

En conclusion, ce projet a permis de concevoir et de développer une solution complète de gestion d'inventaire pour le domaine de l'aviation, répondant aux exigences spécifiques des différentes parties prenantes. En intégrant des fonctionnalités robustes telles que la gestion des vols, des avions, des passagers, des réservations, des équipages, et des aéroports, le système offre une plateforme centralisée et sécurisée pour une gestion efficace des données critiques.

L'utilisation des outils modernes et des méthodologies appropriées a permis de garantir la qualité du développement, de la documentation et du déploiement. Les choix technologiques, comme l'API Spring Boot sécurisée par JWT, le stockage via PostgreSQL, et l'intégration continue avec Docker et GitLab CI, témoignent de l'approche rigoureuse adoptée pour assurer la scalabilité, la performance et la maintenabilité du système.

Ce projet constitue une base solide pour des améliorations futures, telles que l'intégration d'analyses prédictives ou d'outils d'aide à la décision, et démontre la capacité à relever des défis complexes dans un environnement technique exigeant.