IndiGo

Fly With Indigo

All in one portal for fulfilling customer needs

Description

This project is a comprehensive flight maintenance and notification system designed for an airport, featuring a React-based frontend, a Python Flask API backend, and Kafka for real-time updates. This system aims to streamline communication between airport staff and customers, ensuring timely updates and efficient management of flight-related information. Administrators and staff can send notifications and broadcast messages, while customers can view real-time updates regarding their flight status, boarding passes, seat numbers, and more. The integration of Kafka ensures that these updates are promptly pushed to the frontend, providing a seamless and responsive user experience. Additionally, the system incorporates secure authentication protocols for both staff and customers, with the backend leveraging MongoDB for data storage and Node.js for socket management. This robust setup not only enhances operational efficiency but also significantly improves the customer experience by keeping them well-informed and engaged.

The pillars of strength- Tech stack

Python Backend

The app uses a Flask API based backend which connects to **MongoDB** and at the same time, provides routes for broadcasting over **Apache Kafka.**

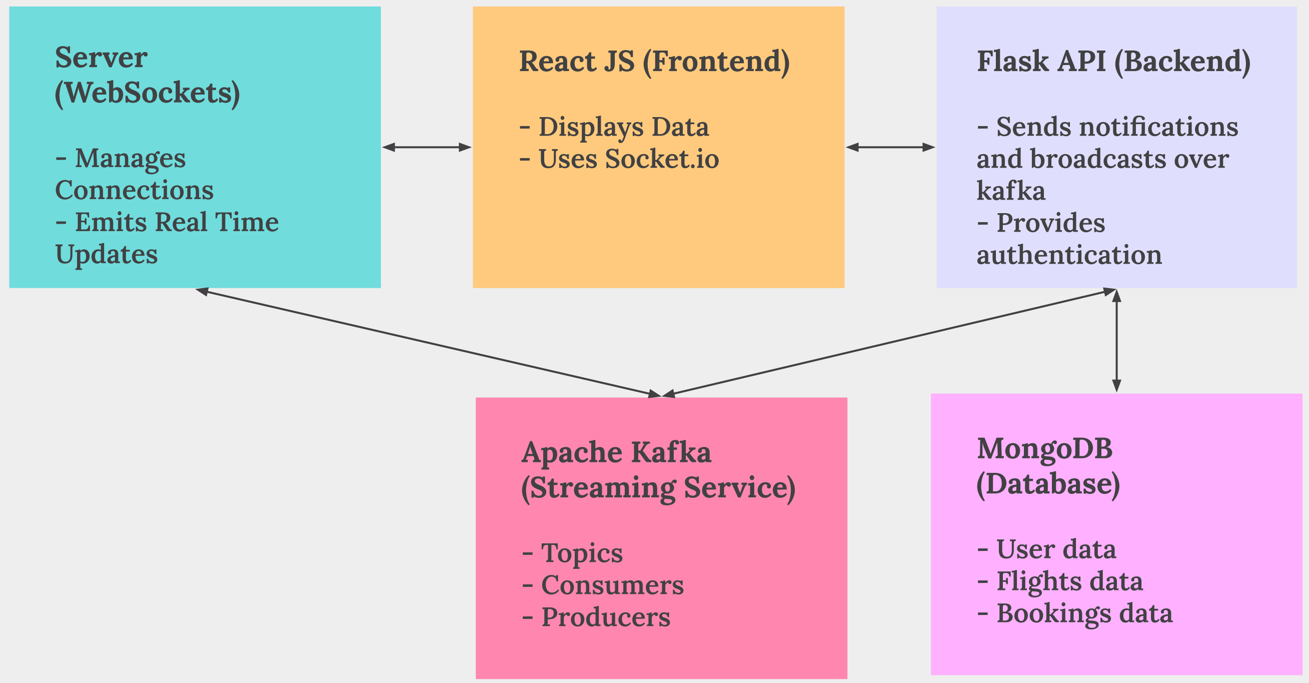
React JS Frontend

The app uses React JS and Tailwind CSS for providing a dynamic front-end for users to easily access and understand the portal.

Node.js Websockets

The **socket.io** library helps bridge the **Apache Kafka** to React based front end. Once the user connects to the websocket, they are looped in for real time updates.

Implementation Details - Diagram



List of features provided by the portal

User Sign Up

Any customer can easily sign up by entering their email address and setting up a password

User Log In

Any customer, staff person or admin can easily log in via the same form and still end up on respective dashboards.

Dashboard

For each role, the dashboard has been customised to cater to their needs.

Book a flight

Any customer can easily search and book a flight. The flight can be searched by arrival or departure too.

Real Time Updates

The admin can easily send either a notification to particular set of people, or can broadcast to all the customers in the database

Boarding

The staff can board or cancel the passenger as per their own request. An email notification for the same will be sent too.

The “Role” based division

Admin

Assigns other employees the role of admin or staff. Sends notifications to particular flights and broadcasts announcement to whole user database.

Staff

Registers new flights in the database and facilitates passenger boarding. Allows passengers to cancel their boarding.

Customers

Register themselves on the portal to search flights and book tickets. They receive updates both by email and on the portal, keeping them informed about their flight.

Apache Kafka & Websockets

Apache Kafka, which is massively scalable, as it allows the data to be distributed across multiple servers, is the best choice for sending out notifications and broadcasts.

Thus using Flask API, the routes are configured to send out data over Kafka topics named ‘notifications’ and ‘broadcasts’.

These events are then listened for by websockets, which are capable of reading the data from Kafka topic and further sending it over the websocket to React based frontend.

Emailing notifications and updates

The Email.js is a powerful library, which allows us to customize our own templates and setting up our own email service. The portal uses email.js to send mails to all passengers whenever announcements are made related to their particular flights. Other than that, whenever the staff marks the passenger as “boarded” or “cancelled”, the passenger received a mail, conveying the updated status of their travel booking.

Project Demo

Let’s take a view at the portal.

Future Scope

- Weather analytics to predict possible delay in flights.
- Using HDFS or PySpark with Kafka to store notifications or broadcasts.
- Adding baggage identification and tracking system.

Thank You!

Email - ayushb6732@gmail.com

LinkedIn - ayush-bansal-560597195