**fLight Information System Using SQLite**

**(Group Name: SoonYeong)**

**Overview**

This project aims to develop a web application that allows passengers to easily check flight schedules using SQLite, a lightweight relational database that requires no additional installation. Leveraging the relational SQL concepts learned in the ADT course, the system will provide an intuitive and user-friendly interface for travelers to access real-time flight information.

By selecting a topic closely related to travel and city-to-city transportation, this project is designed to be both engaging and relatable. The subject matter ensures that other student groups can easily understand and find relevance in the application, making it an accessible and practical demonstration of database-driven web applications.

**Implementation Plan**

**1.1 Data**

To implement this service, the following datasets are required:

* **Basic Data** (Airport Code, Airline Code, Gate Number, Claim Number, Status, Delay or Cancellation Reasons, etc.)
* **Flight Data** (Season Schedule, Active Schedule Data)
* **User Data** (Administrator, Airline, and Agency User Information)
* **Management Data** (Display Information)

**1.2 Technology & Solution**

* **Backend** (Node.js + Express + SQLite + WebSocket)
  + Develop an API to load initial flight schedule data and provide real-time updates to the front end via WebSocket.
  + The dashboard and schedule management pages will also utilize WebSocket to load initial data and update real-time changes.
  + If WebSocket implementation is challenging due to time constraints, polling will be used as an alternative.
  + When implementing a web application, SQLite may be replaced with MySQL depending on the situation.
* **Frontend** (React + TanStack Table or AG Grid + WebSocket)
  + The UI will be built using React, with additional libraries to create a data grid that allows for easy database management (Insert, Update, Delete, Select). Suitable components and libraries will be identified and applied during the project.
  + The flight schedule display will be developed using **React + AG Grid + WebSocket**, ensuring real-time schedule updates.

This project is designed primarily to verify service functionality rather than to support a large-scale concurrent user environment. If high traffic volume is required, switching from **SQLite to MySQL or PostgreSQL** would be more suitable. However, for the scope of this course project, SQLite is deemed appropriate. The focus will be on defining relationships between data tables, specifying data types, constraints, **Primary Keys (PK), and Foreign Keys (FK)**, rather than on UI/UX design.

**1.3** **Implementation example**

Display UI

A screenshot of a flight schedule

AI-generated content may be incorrect.

Management UI

A screenshot of a document

AI-generated content may be incorrect.

**Expectation**

1. By implementing a web application for displaying flight operation information, this project provides hands-on experience in relational database design.
2. In addition to database design, the project allows for the design and implementation of the overall web application architecture, enhancing development skills.
3. The service has the potential to be expanded beyond airports and applied to various other domains.