My project is an **Airline Booking System** that allows users to search for flights, book tickets, and manage their reservations. The project was built using JavaScript in the backend and Express server. I used the MVC design pattern to organize the code, which made it easier to maintain and scale. One of the key features I implemented in this project is the use of middlewares. These allowed me to add additional functionality to the application, such as authentication and validation, without having to add it directly to the routes.

To handle the database, I used Sequelize ORM, which made it easy to interact with the database and perform CRUD operations. I also used Cron Jobs to schedule tasks such as sending confirmation emails to users and cleaning up expired reservations. Another important aspect of my project is the use of a message queue. I implemented RabbitMQ to handle the communication between different parts of the system and ensure that the system remains responsive even under high load. To handle inter-service

communication, I used axios, which is a popular library for making HTTP requests.

The architecture of my project is based on Microservices, which allows different parts of the system to be developed and deployed independently. I also used an API Gateway to route requests and handle authentication using JWT.

Finally, I deployed the project on AWS, which provides a reliable and scalable infrastructure for hosting the application.

Overall, I believe my Airline Booking System is a robust and scalable solution that can handle high traffic and provide a seamless user experience. The use of modern technologies such as MVC, Middleware, ORM, Cron Jobs, RabbitMQ, axios, Microservices, JWT and API Gateway, and deployment on AWS made the project more efficient and easy to maintain.

Git:

Link: