**Flight Booking System**

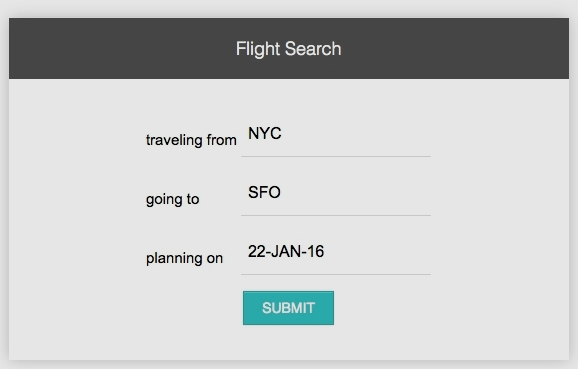
We will have to accomplish the following items in our microservices implementation:

1. Search, Booking, and Check-in interact with each other through asynchronous messaging.

As shown in the preceding diagram, we are implementing four microservices as an example: Search, Fare, Booking, and Check-in. In order to test the application, there is a website application developed using Asp.net MVC. The asynchronous messaging is implemented. In this sample implementation, the default H2 database is used as the in-memory store for demonstration purposes

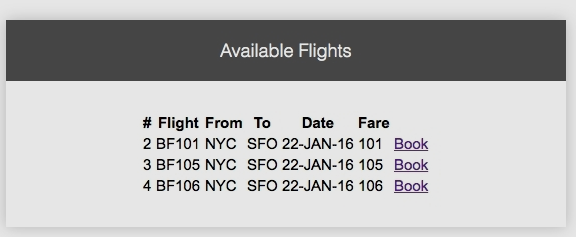
The browser asks for basic security credentials. Use guest or guest123 as the credentials. This example only shows the website security with a basic authentication mechanism. service-level security can be achieved using OAuth2.

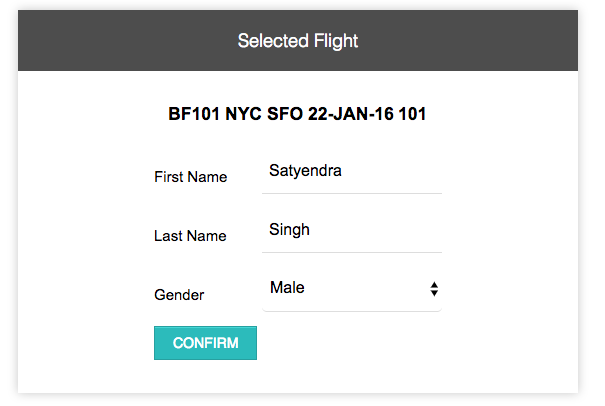
Entering the correct security credentials displays the following screen. This is the home screen of our Flight Booking application:



The SUBMIT button invokes the Search microservice to fetch the available flights that meet the conditions mentioned on the screen. A few flights are pre-populated at the startup of the Search microservice. Edit the Search microservice code to feed in additional flights, if required.

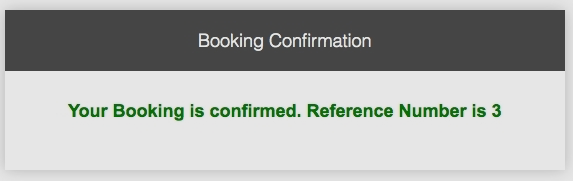
The output screen with a list of flights is shown in the next screenshot. The Book link will take us to the booking screen for the selected flight:





The following screenshot shows the booking screen. The user can enter the passenger details, and create a booking by clicking on the CONFIRM button. This invokes the Booking microservice, and internally, the Fare service as well. It also sends a message back to the Search microservice:

If booking is successful, the next confirmation screen is displayed with a booking reference number:



Give a checkin link after confirmation.

Click on the CheckIn link to perform the check-in.

If check-in is successful, it displays the confirmation message, as shown in the next screenshot, with a confirmation number. The Check-in service sends a message to Booking to update the check-in status:

