Flight Planner

STA 141B Final Project

## Introduction:

Traveling has been an important part of modern society. Many people travel for business purposes, to study abroad, or just to enjoy some nice family vacation time together. In a way, traveling had become an essential part of communication between different countries, as many political officers also travel to better represent their government in World meetings and conventions. With this in mind, our group created a shiny application that is design to help travelers’ better plan their flights ahead of time. The application will not only display flight information regarding total flight time, flight frequency, ticket price, etc. while also be able to accommodate with users changing their flight origin or destination as they plan their travels. Our main goal with this application is to make the air travel portion of traveling process easier to handle and more transparent.

## Source Data:

All displayed materials comes from one of the industry recognized flight database Amadeus, where we were able to retrieves flights information that corresponds to users’ input. This is a dynamic database that works with real time information relay with constant updates from different channels. It collects data not only from various airport and airline all over the world, but also other transportations including railroad, cruise lines, etc. It also works with travel agencies and local vacation related business, hotels, and even tour operators.

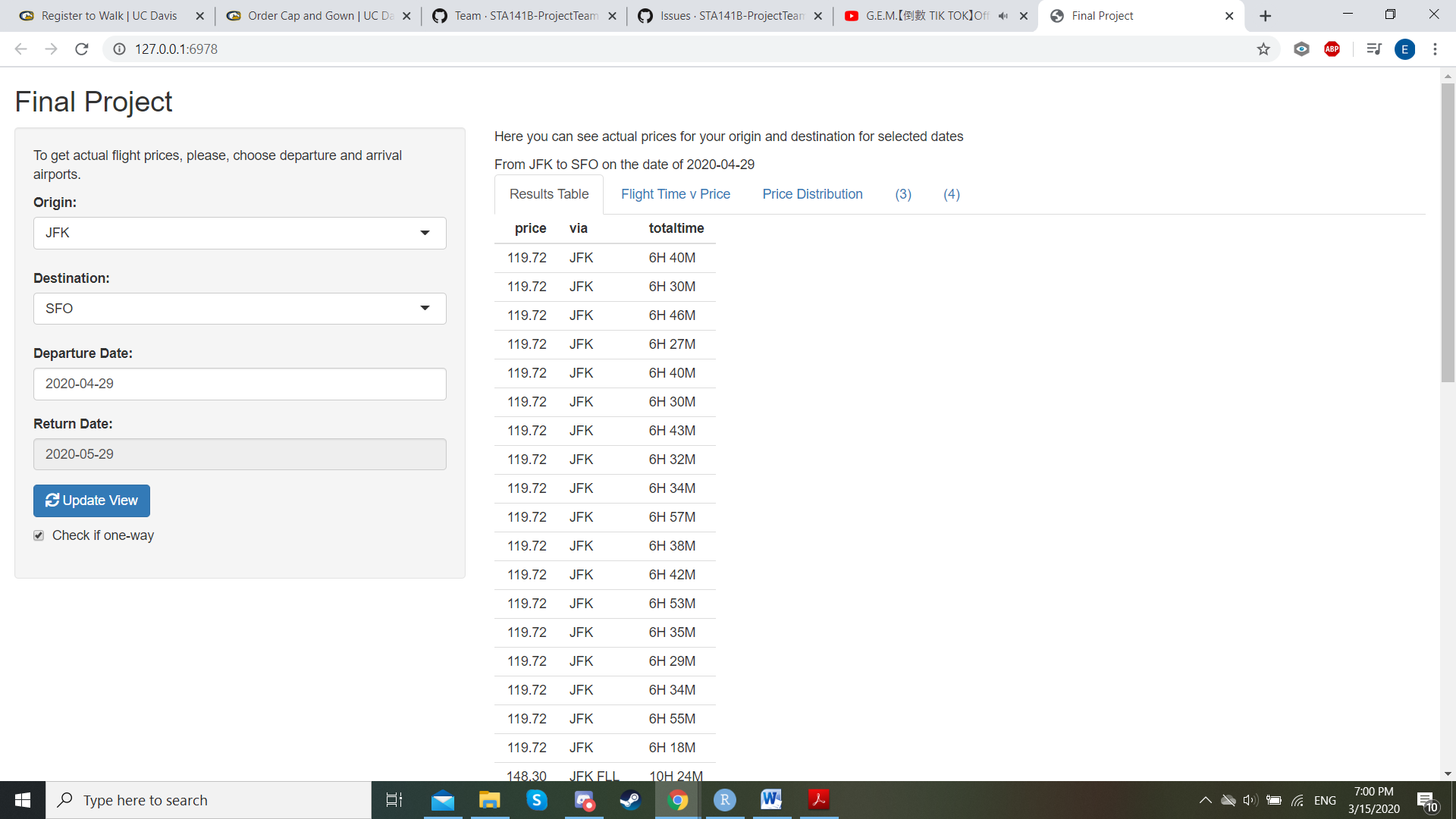
## Data Retrieval:

Amadeus itself provides a well-developed API for querying data. It utilized the “https” protocol with the methods GET and POST. In the application, when working with orders, the DELETE method is also used to help comb through the retrieved data. There are numerous Amadeus APIs available that provides a variety of possibilities for searching all kinds of flight offers, creating ticket orders, and more. The API uses a two-step authorization. At the first step, access token is obtained sending POST request with user input. The body of POST request is similar to GET request parameters string; i.e. it is a string containing name/values pairs (“name=value”) separated by “&”. At the second step, actual data request is sent using GET method.

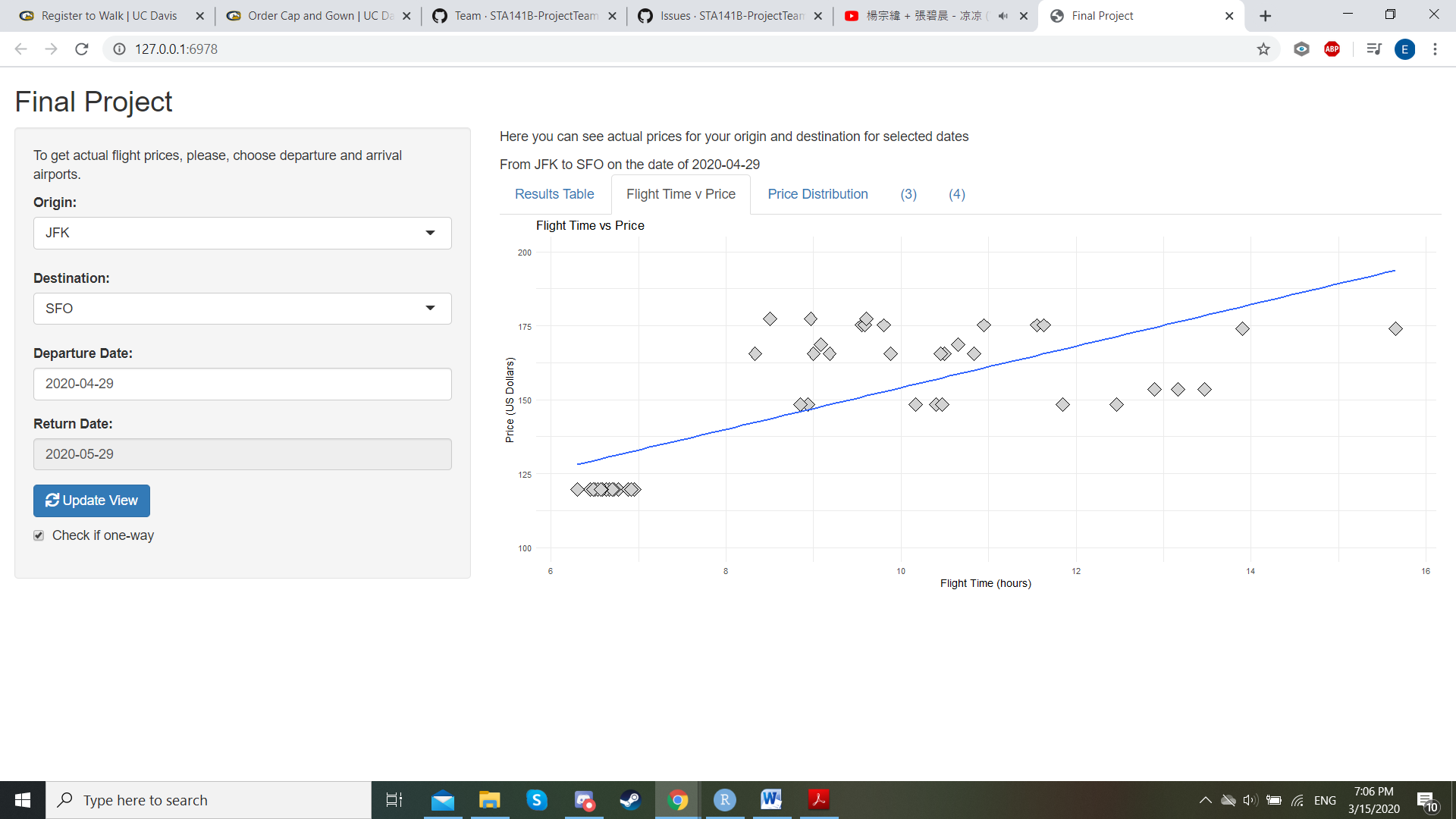
This kind of two-step authorization provides additional flexibility and security as each access token can be used only once. Any third party personnel will not be able to change the clients’ travel without their permission. Consequently, data can also be received from Amadeus directly by a client, and all JSON processing job could be done on the client side. Client JSON parsing is within the scope of further steps of the project.

The company itself aims for a comfortable and stress free travel planning tool, which also aligns with our goal in this project. However, in this shiny application, we are focusing mainly on the flight aspect of the travel and shoot for an easy to use interface which allows the users to better maneuver through the massive amount of data available through Amadeus.

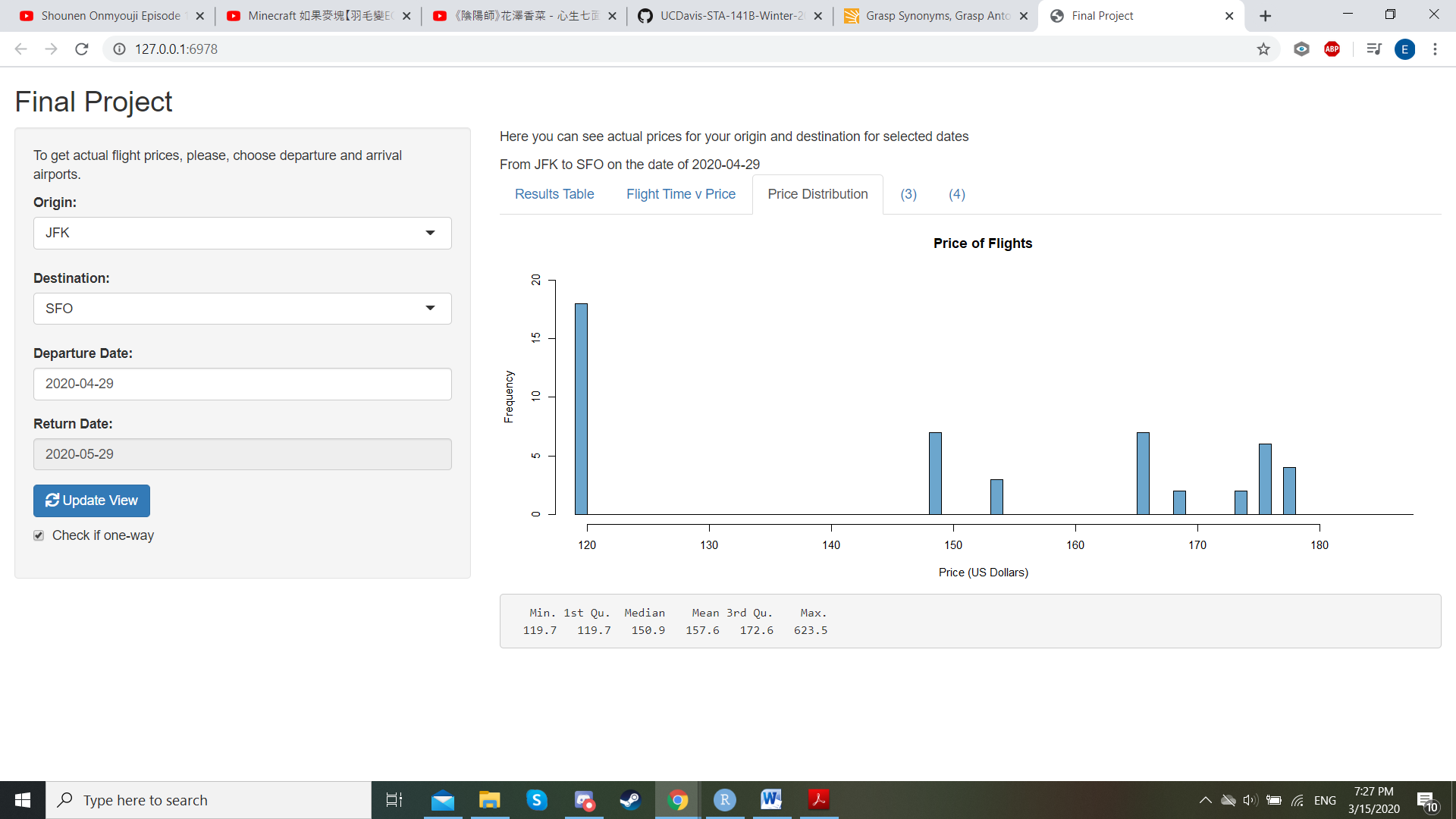
## User Guide:



The application has two panels, the left panel and the main panel. On the left panel, there are several controls that allow user input. User can choose departure airport, arrival airport, and departure date. After data input, user can press “Update View” button for the data to pass on to the server. The retrieved data is then shown in the main panel. The main panel has several tabs. The first tab displayed all the offers found on Amadeus for the given departure and destination airports from the selected date. It showcased a table with three columns including offered price, transit airports, and total time of travel. The total time of travel is the total flight time between the scheduled departure time of the initial leg and the scheduled arrival time of the final leg of the trip.



The second tab is a scatterplot that displays the price and total flight times of the offers found on Amadeus for the given departure and destination airports from the selected date. From this tab, we can attempt to find a trend between price and the amount of travel time expected for the trip. This tab can help users get a better grasp at the currently offering ticket price and can also help the users plan for transportation after they had reach their destinations. For example, most flights from JFK to SFO (the flight information/user input is displayed in the left panel at the page above) generally takes around six to seven hours with the price ranging from $120 up to as high as $200 US dollars.



The third tab is a price distribution histogram of the offers found on Amadeus for the given departure and destination airports from the selected date. From this tab, details the price variations between different flights. Continuing the flight data from JFK to SFO, we know that the range of ticket price ranges from $120 up to as high as $200 US dollars. The bar graph also showcases the amount of flights available for each ticket price ranges. For example, we can tell from the graph above, that majority of the plane ticket is around $120 US dollars and have around 18 flights available on the selected date. However, we can also see that there are still some higher price tickets that are also available that may be indicative of different travelling classes, e.g. first or business classes vs economy.

A close up of a map

Description automatically generated

The fourth tab is a map that contains the locations of 100 airports. From this tab a user can see what airports are near the area they want to travel to. A user can also see if there are multiple airports near their desired destination. The user can then use this information to further analyze flight prices. For example, if one wanted to travel to New York they could simply hover over the New York area and see that there are three different airports in that area. Once the user has the information of which airports they can travel with. They can then use the Amadeus data to see which airport will give them the best price.