GitHub Link: https://github.com/edwardam5/507_FinalProject_Team5.git

Video Link: https://youtu.be/LFSEu1k96hY

Design Breakdown of the ELT based United States Airline Flight Status Dashboard Pipeline

Summer Purschke, Minsu Kim, and Anusia Edward

Shiley-Marcos School of Engineering, University of San Diego

Introduction

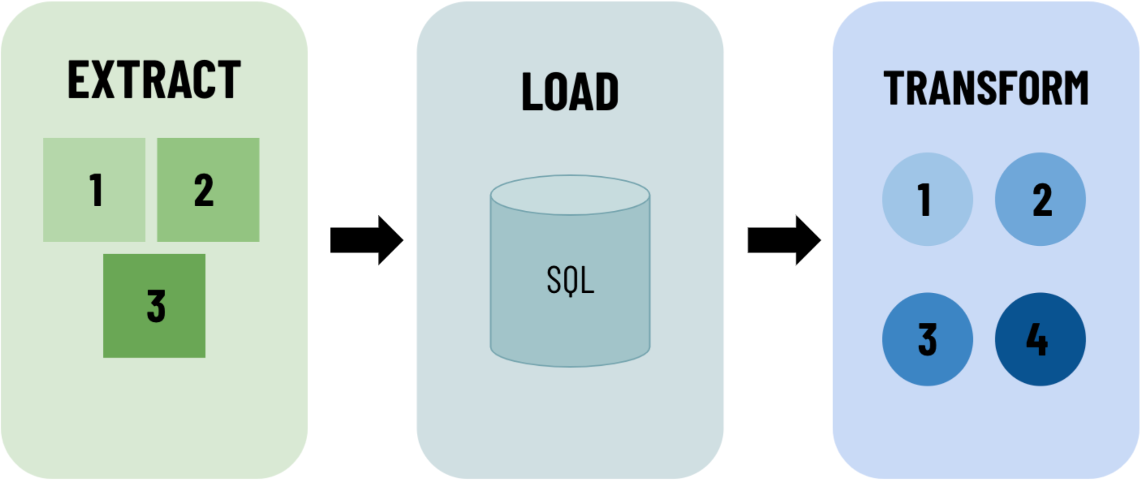
Within the United States, the most popular mode of public transportation for long distance travel is airplanes (*Long Distance Transportation Patterns: Mode Choice*, n.d.). Air travel in the U.S. is overseen by the Federal Aviation Administration (FAA), which comes under the United States Department of Transportation (*Air Traffic by the Numbers*, n.d.). Based on the statistics provided by the FAA, there are around sixteen million flights yearly and around ten million yearly passengers. Over the course of the past ten years (2012-2022), twenty percent of U.S. flights were delayed (*Airline On-Time Statistics and Delay Causes*, 2022). Delays in flights can be caused by a number of reasons including late aircraft arrival, security measures, extreme weather conditions, and mechanical issues (*Air Traffic by the Numbers*, n.d.). These delays may prevent passengers from getting to their destinations in a timely manner. This can then affect the professional and personal affairs (i.e., business meetings, funerals, etc.) of the passengers. Therefore, as a passenger, being able to view flight status is an important component of feeling prepared and secure during travel. To help passengers have access to this information, a dashboard with visualizations was created to help with identifying flight status and making informed decisions when choosing specifics like airline type, departure and arrival location, and day of travel for their flights.

Methods

Kaggle was used to source all three of the datasets utilized within this project. The first dataset, labeled “flights,” has information regarding the following: day of the month on which the flight was taken, day of the week on which the flight was taken, the flight carrier (Delta Airlines, American Airlines, ExpressJet Airlines, Jetblue Airways Corporation, Frontier Airlines, Airtran Airways, Hawaiian Airlines, Envoy Air, Endeavor Air, Alaska Airlines, United Airlines, Virgin America, Southwest Airlines, Mesa Airlines, and Skywest Airlines), airport ID for the departure site, airport ID for the arrival site, the number of minutes a departure was delayed (-63 to 1863), and the number of minutes an arrival was delayed (-94 to 1845). It should be noted that negative departure and arrival times indicate that a flight departed earlier or arrived earlier than scheduled. The second dataset, labeled "airports," has information on the following: airport ID, the city the airport is located in, the state the airport is located in, and the name of the airport. The third dataset, labeled "statelatlong," has information on the following: state abbreviation for the U.S., latitude, longitude, and city. These datasets were chosen, as they provided adequate information needed to present passengers with details about their flight status.

Figure 1.

Architecture Diagram



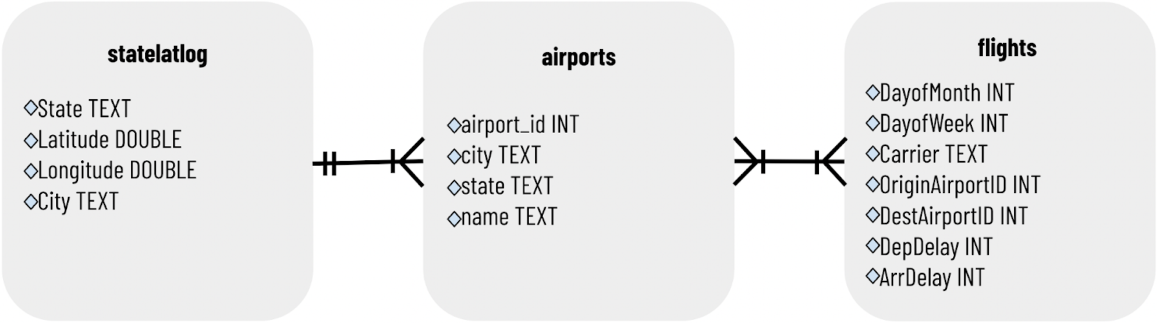
After researching and determining the appropriate data to pull for this dashboard, the following method, displayed through the Architecture Diagram, was carried out (Figure 1). The Architecture Diagram provides a visual of the steps taken to extract, load, and transform the data. In the first step, involving extraction, the three previously mentioned datasets ("flights", "airports", and "statelatlong") were extracted from Kaggle. These datasets were then loaded into

SQLWorkbench as a database titled *us_airlines*. This database was then accessed in Visual Studio's SQL through a Python notebook. Then, four transformations were carried out on the datasets as seen in Figure 1.

The schema for the three datasets is displayed in Figure 2. Transformations were carried out based on the relationships seen within the schema. The relationship between "statelatlong" and "airports" is such that each airport can only be associated with one and only one state, and each of the states can be associated with multiple airports. This relationship is indicated through the parallel lines and the single line with crow's feet respectively. The relationship between "airports" and "flights" is that an airport can be associated with multiple flights and flights can be associated with multiple airports. This is displayed through the crow's feet with a single line.

Figure 2.

Final Schema Diagram



Based on the relationships seen through the schema, the transformations performed include: (1) inner join on the "flights" table onto the "airports" table to connect departure city, state, and airport with departure id, (2) inner join on the "air_data_t1" table onto the "airports" table to connect arrival city, state, and airport with arrival id, (3) inner join on the "air_data_t2" table onto the "statelatlong" table to obtain longitude and latitude of the departures based on U.S.

departure state in the “airports” table, (4) inner join “air_data_t3” table onto “statelatlong” table to obtain longitude and latitude of the arrivals based on U.S. arrival state in the “airports” table. These transformations were saved as follows: (1) “air_data_t1”, (2) “air_data_t2”, (3) “air_data_t3”, and (4) air_data_t4.

Following the transformations, six visualizations were produced. The first visualization created is an interactive map (Figure 3). This interactive map contains zoom features, an interactive search bar, destination pins and pop-ups indicating the searched location, and flight paths. The way in which this map was created was through using the following functions: *folium.Map, folium.CircleMarker,* and *folium.PolyLine* from the *folium* package. The *folium.Map* function was used to create the baseline map, while the *folium.CirleMarker* was utilized for the departure latitude and longitude point and for the arrival latitude and longitude point. The *folium.PolyLine* function was used to connect the departure and arrival points, thus creating the flight path. After the interactive map visualization, the interactive table was produced (Figure 4). This table displays U.S. flight status information that is controlled using an interactive dropdown button containing the flight numbers. The way in which this interactive table was produced was through the combination of the *hvplot.table* function from the *hvplot* package and the *pn.widget* function from the *panel* package.

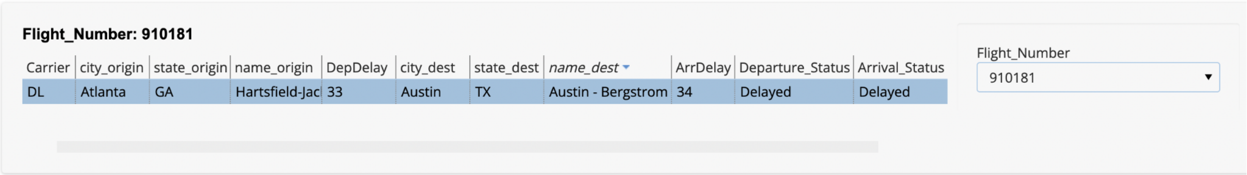
Next, an interactive bar chart was produced, which visualizes the average departure time in minutes by day of the week (Figure 5). When one hovers over this bar chart, a pop-up of the average delay time for that particular day is seen. This visualization was created through the *px.bar* function of the *Plotly* package. Then, the fourth visualization is an interactive bar chart that was produced to visualize the average arrival time in minutes by the specified day of the week (Figure 6). When one hovers over this bar chart, a pop-up of the average delay time for that particular day is seen. This figure was also produced through *Plotly'*s *px.bar* function. Finally, the last visualization is an interactive chart that indicates the various arrival and departure times in terms of how early or late the aircraft was in minutes by the aircraft carrier. This figure was created through pipelining *Panel* widgets using the *pn.widets.RadioButtonGroup* function and the *hvplot* function. The *pn.widets.RadioButtonGroup* function was used to create the interactive button, allowing for the chart to switch from departure stats to arrival stats per carrier. All six visualizations were then pipelined to be outputted as a browser server through the *Panel* packages as well as the *interactive()* function, which allowed the pipelined outputs to be interactive on the dashboard.

Results

The dashboard pipeline outputted a total of five different visualizations. The first visualization at the top of the dashboard is an interactive table (Figure 3). Based on a given flight number, the interactive table will pull information regarding that particular flight. A dropdown bar containing the flight numbers is utilized to carry out this function. This will then provide flight information, which will include the following: airline carrier, departure city, departure state, departure airport name, number of minutes the departure is delayed or early, the arrival city, arrival state, arrival airport name, number of minutes the arrival is delayed or early, departure status, and arrival status. Additionally, when a user hovers over the information, it becomes highlighted (in blue) for ease of visibility. Furthermore, there is a slider at the bottom of the information table that allows the user to scroll through the information.

Figure 3.

Interactive Table of Flight Status by Flight Number



Flight_Number: 910181

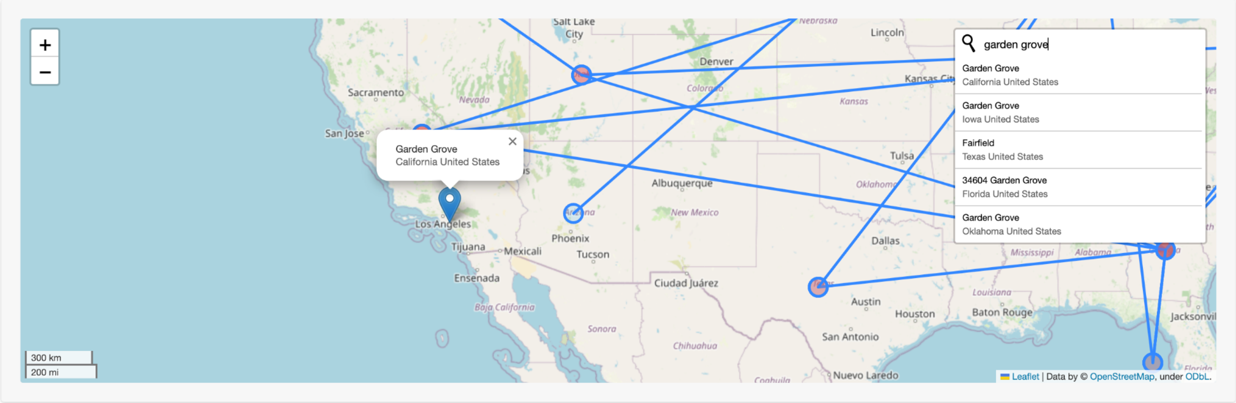
Carrier	city_origin	state_origin	name_origin	DepDelay	city_dest	state_dest	name_dest	ArrDelay	Departure_Status	Arrival_Status
DL	Atlanta	GA	Hartsfield-Jac	33	Austin	TX	Austin - Bergstrom	34	Delayed	Delayed

Flight_Number: 910181

The next visualization that is seen on the dashboard is an interactive map. This interactive map contains a few different features. First, it contains a search bar that allows a user to search for any destination. Once the destination is chosen, a blue indicator appears along with a pop-up indicating the location. This pop-up can also be exited. The next feature that this interactive map has is a zoom feature that allows one to zoom close enough to streets and far enough to see the entire world map. Another feature that the interactive map contains is the flight paths, as seen in Figure 4. The red circles indicate the destination of the flight while the blue circles indicate where the flight took off from.

Figure 4.

Interactive Airline Flight Path Map



garden grove

Garden Grove California United States

Garden Grove Iowa United States

Fairfield Texas United States

34604 Garden Grove Florida United States

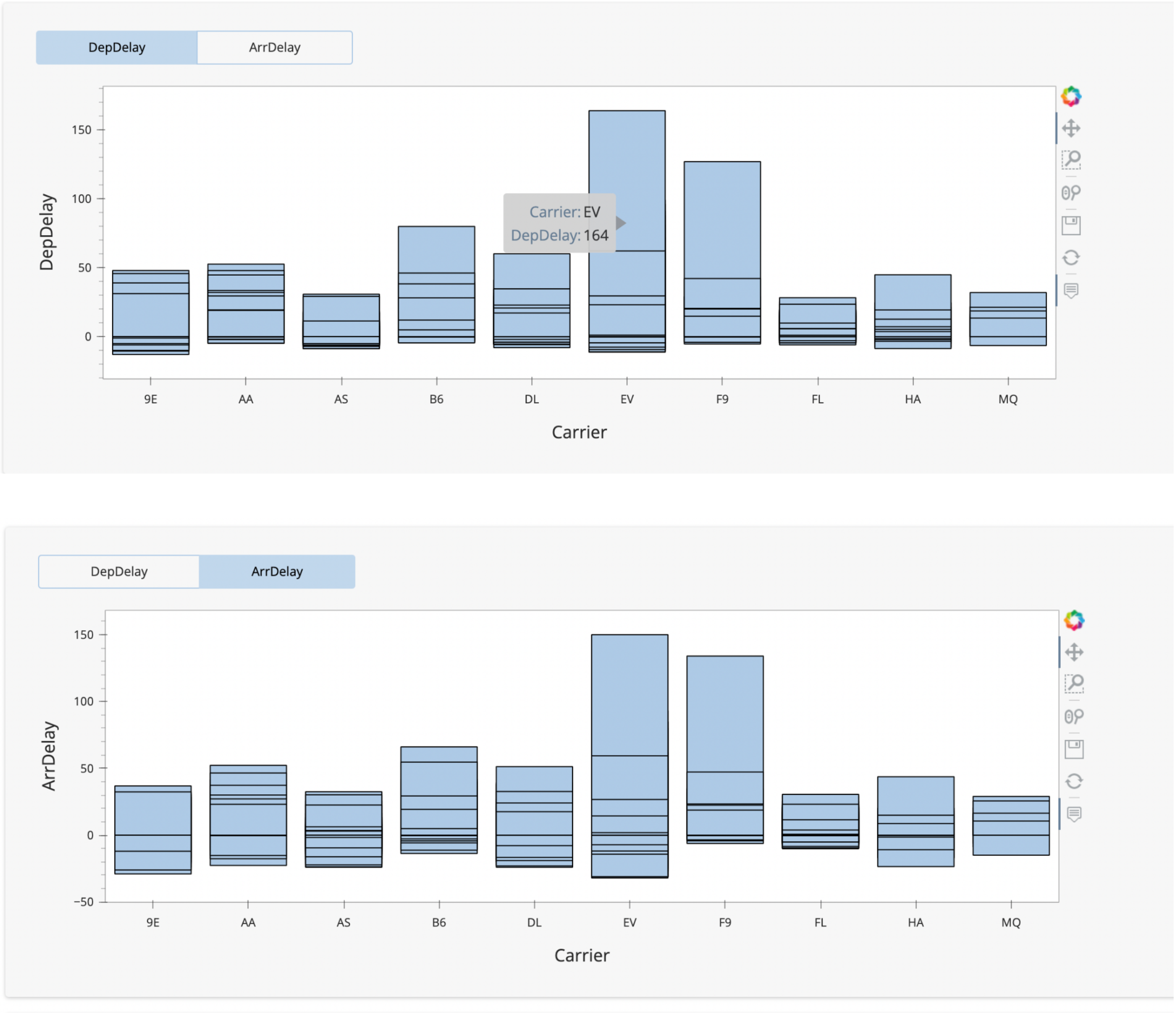
Garden Grove Oklahoma United States

Garden Grove California United States

The third visualization on the dashboard is the bar chart, which shows the range of departure times as well as arrival times in terms of how many minutes early or late the flights are per each airplane carrier. This visualization includes a button at the top left that allows the user to flip back and forth between arrival delays and departure delays. Additionally, this visualization has a pop-up feature that indicates varying stats pertaining to departure and arrival time in minutes per carrier as seen in Figure 5.

Figure 5.

Bar Chart of Departure and Arrival Delays by Carrier



The fourth visualization that was outputted onto the dashboard is the interactive bar chart of the average departure time in minutes by day of the week (Figure 6). This visualization's interactive feature is the pop-up information box that appears when a user hovers over the chart. The information provided is in regards to how long it takes for a plane to depart on a specified day. For example, on average, U.S. airplanes depart approximately fourteen minutes late on Fridays, as seen in Figure 6. The last visualization on the dashboard is an interactive bar graph depicting the average arrival time in minutes per each day of the week (Figure 7). A pop-up information box, in response to a user hovering over, is its key interactive feature.

Figure 6.

Average Departure Time (Min) by Day of the Week

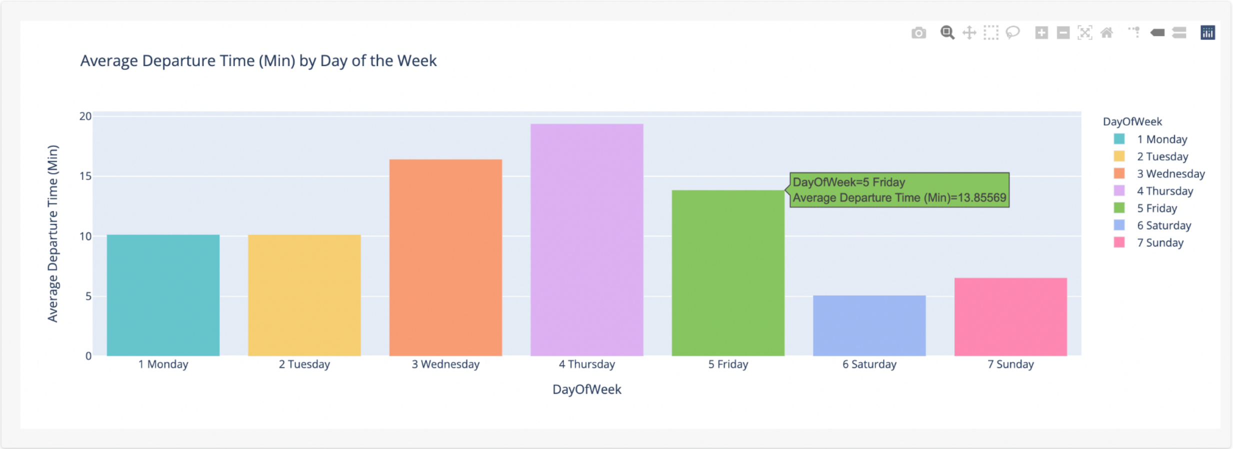
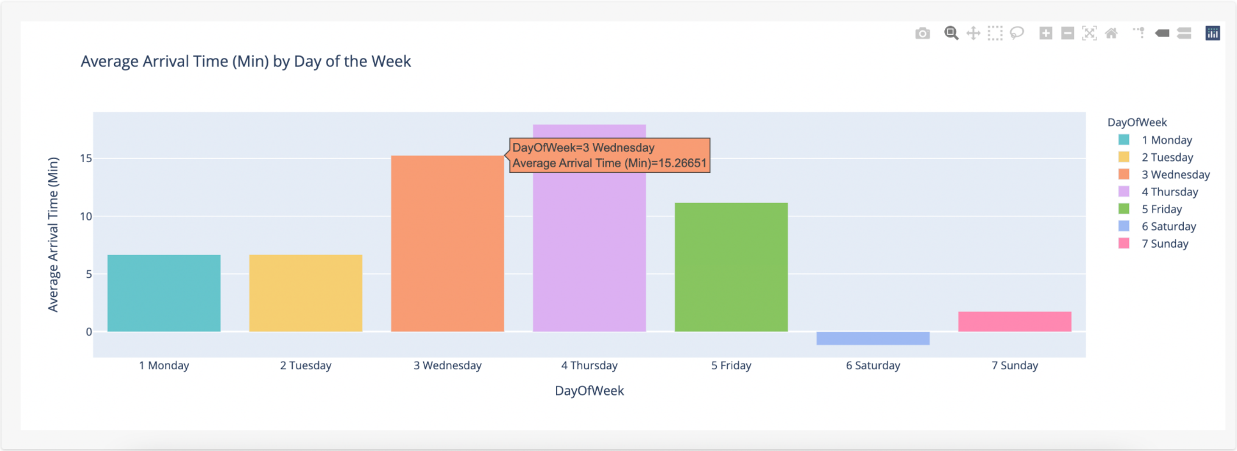


Figure 7.

Average Arrival Time (Min) by Day of the Week



Discussion

For this project, an interactive dashboard was created through a pipeline, complete with five different interactive visualizations. The output of this dashboard is useful because it provides individuals with a secure platform to gather useful information regarding their US based flights. For example, the first visualization of the interactive table can be used by a U.S. flight passenger to determine when their flight will depart or arrive, along with the airport that they are departing from as well as the airport location of their arrival. This may be particularly useful when determining how to schedule and book their post-flight events and lodging locations. Additionally, it can be useful for the people receiving the person flying through US airlines, as they will be able to determine approximately how long they are going to have to wait for whomever they are picking up. The second visualization of the interactive map can be useful when wanting to see the flight paths. Also, through the interactive map, an individual will be able to determine the nearest facilities such as retail stores, hotels, and other commodities through the search bar and zoom feature. The bar chart below the interactive map, with radio buttons to see arrival and departure delays by carrier, is also useful. Information regarding generally how long each airline carrier is delayed, in terms of their departure and arrivals, can be useful to an individual when considering which carrier to book their flight with. Depending on how urgently they may need to fly, they can consider each of the carrier's departure and arrival delay times on average. Correspondingly, the bar graphs indicating the average delay time for arrival and departure by day of the week for the US airlines can be taken into consideration when determining when to book one's flight.

From a technical perspective, the dashboard is also useful in terms of management and upkeep. When considering the scalability of the pipeline for this database, for example, the system is using a static dataset; however, it is possible for this system to scale if live data were chosen to be implemented. Additionally, this dashboard is useful from a technical perspective, as the system setup is fairly straightforward and simple enough to be reproducible. Although this dashboard effectively communicates the necessary information for an individual concerned with the status of their US based flight, some improvements are possible. In the future, one could further improve this dashboard by including a way in which an individual could track their flight paths on the map in real time. This would be beneficial for not only the aircraft passengers, but also those who are waiting to receive passengers. The system is also not entirely secure, as the outputted data is run through a local host system. Local servers do not have the same capacity as an HTTP site in terms of security (Puripunpinyo, 2022). Updating the system to be more secure through the development of a Root Secure Sockets Layer (SSL) certificate may be a useful option to be considered (Ganguly, 2022). Additionally, adding in widgets that indicate weather patterns, along with collecting data and information about why flights were delayed may prove to be beneficial.

References

Airline On-Time Statistics and Delay Causes. (2022). Bureau of Transportation Statistics. https://www.transtats.bts.gov/OT_Delay/OT_DelayCause1.asp?20=E

Air Traffic By The Numbers. (n.d.). Federal Aviation Administration. https://www.faa.gov/air_traffic/by_the_numbers

Flights and Airports Data. (2019, May 30). Kaggle. https://www.kaggle.com/datasets/tylerx/flights-and-airports-data?select=airports.csv

Ganguly, S. (2022, August 3). *How to test HTTPS Websites from Local Host | BrowserStack*. BrowserStack. https://www.browserstack.com/guide/how-to-test-https-websites-from-localhost#:~:text=You%20need%20to%20develop%20a,key.

Long Distance Transportation Patterns: Mode Choice. (n.d.). Bureau of Transportation Statistics. https://www.bts.gov/archive/publications/america_on_the_go/long_distance_transportation_patterns/entire#:~:text=Across%20all%20age%20groups%2C%20the,those%20age%2065%20and%20above.

Puripunpinyo, H. (2022, April 19). *The Danger of Exposing Localhost to Internet - Hussachai Puripunpinyo - Medium*. Medium. https://hussachai.medium.com/inviting-hackers-to-your-machine-using-local-tunneling-19ce2d275e8e

USA lat,long for state abbreviations. (2017b, February 3). Kaggle. https://www.kaggle.com/datasets/washimahmed/usa-latlong-for-state-abbreviations