Davin Alexander(2802530653)

Davinalexander123@yahoo.com

algorithm & programming

Final Project Report

Introduction

I have created a Real-time flight Information on python using tkinter. The goals for this project is to make easier for Users to look on Flight Information anytime, anywhere. It can help Users to reduce stress especially those who may not unfamiliar with the airport. If the user was too far away from the board flight information, Users just look through the flight information on the page using Laptop, Phones, iPads and etc. Without walking just to look on the board flight information..

Background image for login page

So, here’s my project:



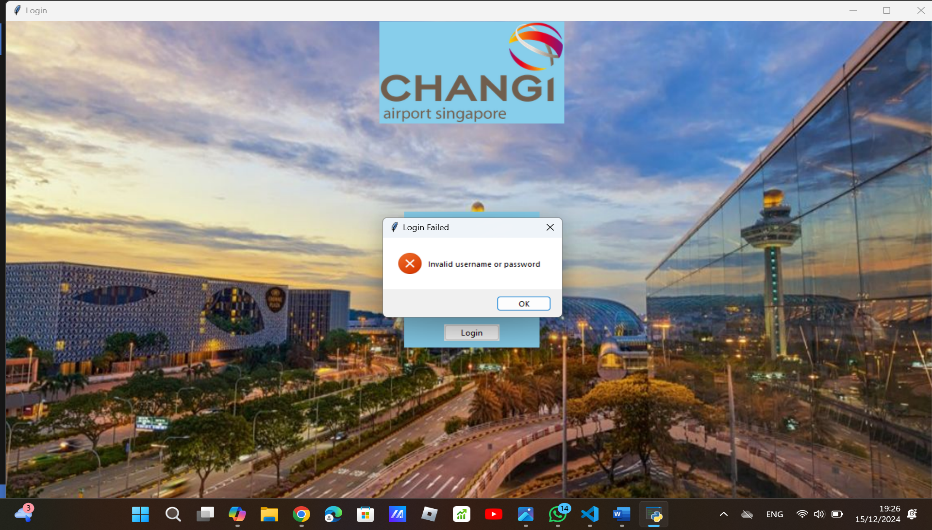
Here’s the code that I used to make a background of a login page. To make the background image fit to the screen. I resize to 1280

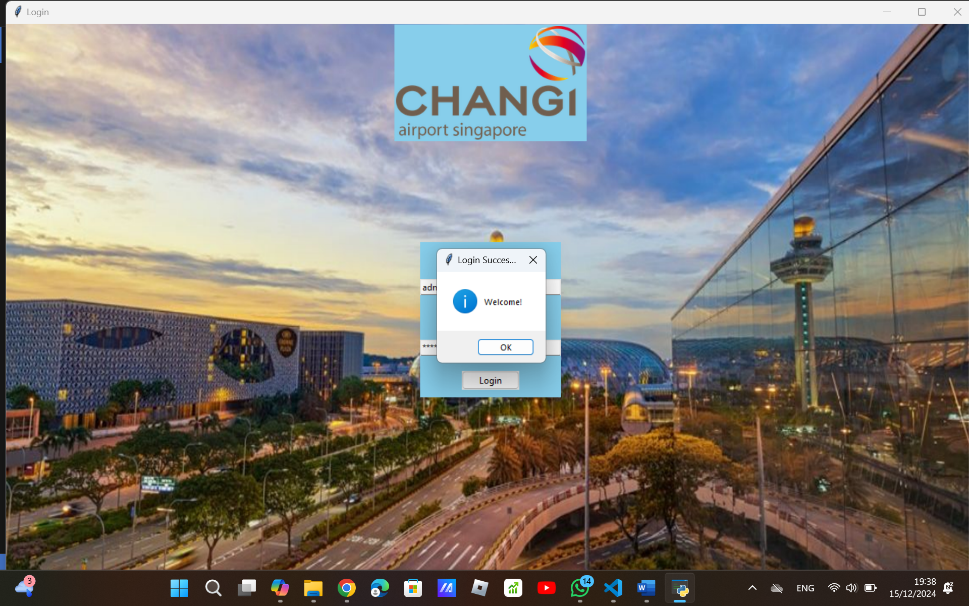


Login Page

So First of all, when the user run the code, It will redirect user to the login page first. Since it is only a demo, I create a simple Username as “admin”. And the password is “1234”.

If the user enter the wrong username or password, It will give user a warning and tell user “Invalid username or password.” It will show like this :



If the user enter the Username and Code correctly. The window will pop up and say “Welcome” and redirect user to the main page.

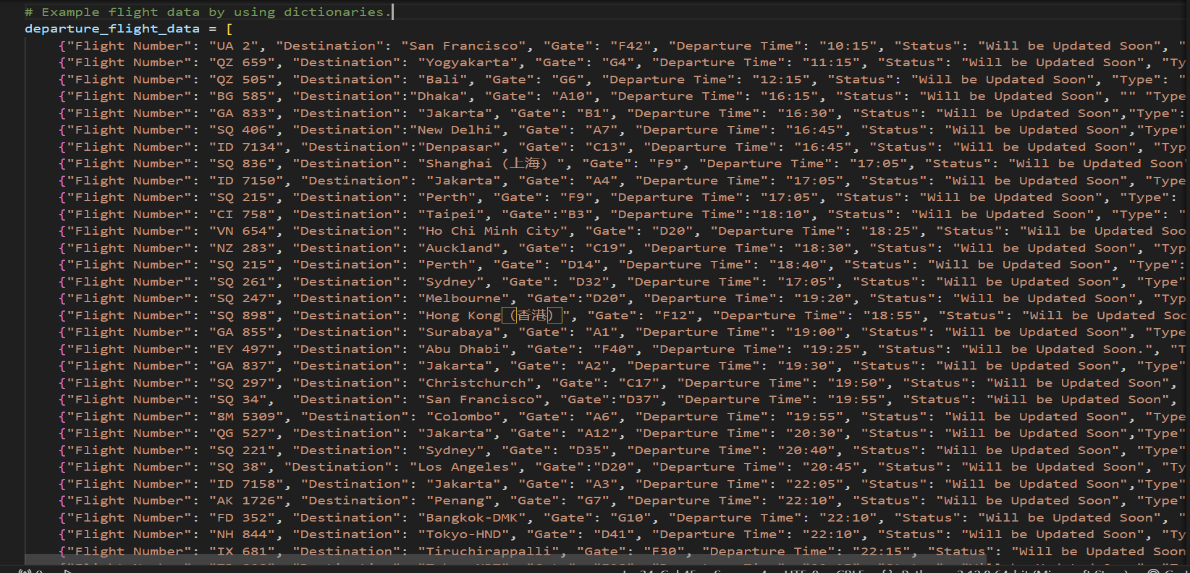


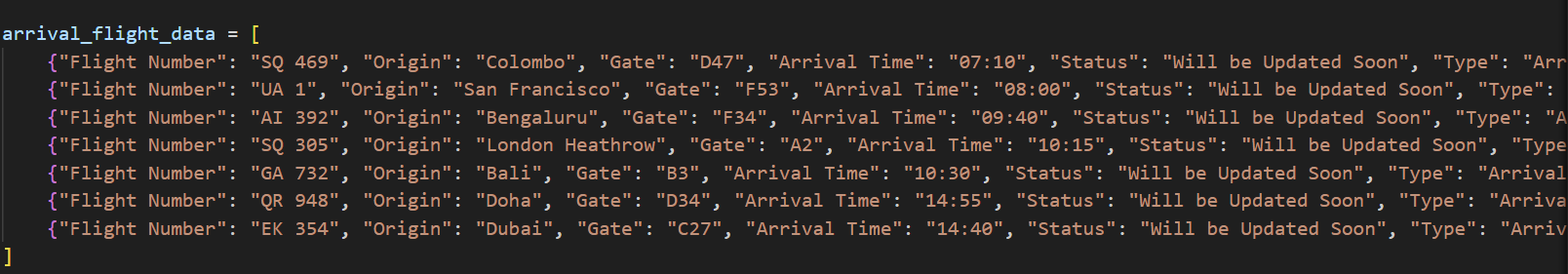
To make the login work, I use the def function. I print the Username and Password. I’m also create Admin and Password. To make sure the username and password are correct, I use the if function. If the Username and Password are matched. It will show a messagebox by printing “Login Successful, Welcome!.” And destroy the login window and proceed to the main page. If the Username or Password are not matched, It will give an error by showing a messagebox by printing “Login Failed, Invalid Username or Password.” And try again.

Main Page

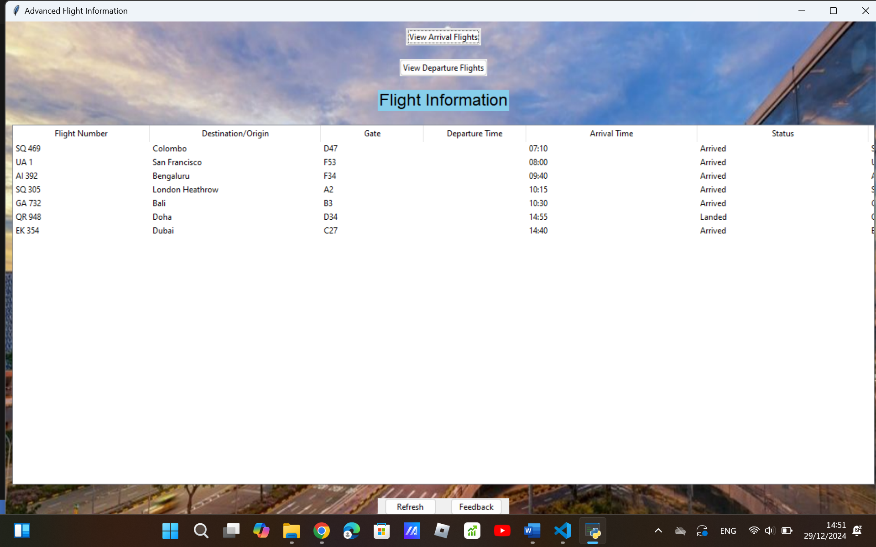


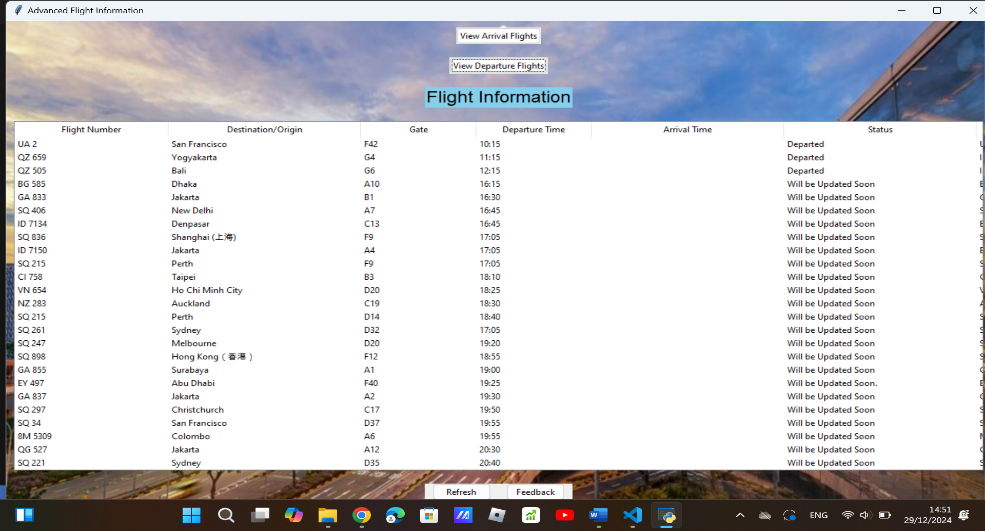
On the image above that is the Main Page where user will be redirected next. After the user have logged in on the Login Page It will show a Flight Information of all Departure and Arrival flight data. There are a few button which I have created. There are: ‘View Arrival Flights’, It will show ONLY the Arrival Flights data to make it easier for user if the user wants to search all arrival flight data. Next is the ‘View Departure Flights’ to show the user to view the Departure Flight Data ONLY. Next is the ‘Refresh’ button. the refresh button would be used to refresh the list of challenges or any dynamic content that might be updated frequently. And also to make User to go back, The user can just click the refresh button if the user is done viewing the Arrival Flights ONLY or Departure flights ONLY. And lastly, there is a feedback button where user can enter their feedback to know what we need to improve and to enhance our service.





To make a flight data, I use a dict. Function to make a dictionary and to store each flight data that I have input. So, I have to input the Flight Number, Origin, Gate, Arrival Time, Status & Type That I want to input on the Arrival Flight Data & I also have to input the Flight Number, Origin, Gate, Departure Time, Status & Type on the Departure Flight Data. So, I have to input it over and over again which make me spent a lot of time.

So, when the user clicked the ‘View Arrival Flights’. It will show the Arrival Flight Data ONLY. So It will make easier for user to find the arrival flights.

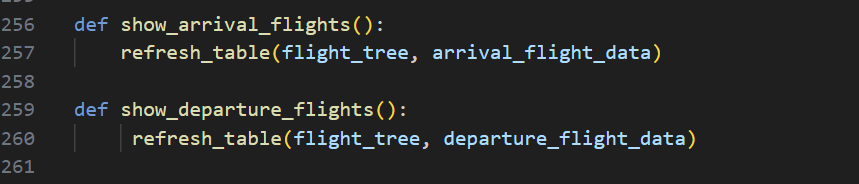


If the User clicked the ‘View Departure Flights’. It will show ONLY the Departure Flight Data.

Even though it’s a lot but It will make user easier to find it.

I make a code for the ‘View arrival Flights’ and ‘departure arrival flights’ To make this work I make a command by to go to the ‘show\_arrival\_flights’ and if the user clicks the button, it will refresh the table and it will go to the ‘arrival\_flight\_data’ where the arrival flight data stored and it contains a dictionary. As well as the ‘View Departure Flights’, I make the same code. If the user clicks the button, it will refresh the table and it will go to the ‘departure\_flight\_data’, it is a dictionary where the departure flight data is stored. And it will show it to the user the departure flight data only. And also to make the ‘show\_arrival\_flights’ and ‘show\_departure\_flights’ work I use the define(def) function. So I can call the code whenever I needed without repeating the code and it can make the code easier to read. as well as easier to update or debugging the program without affecting other code.

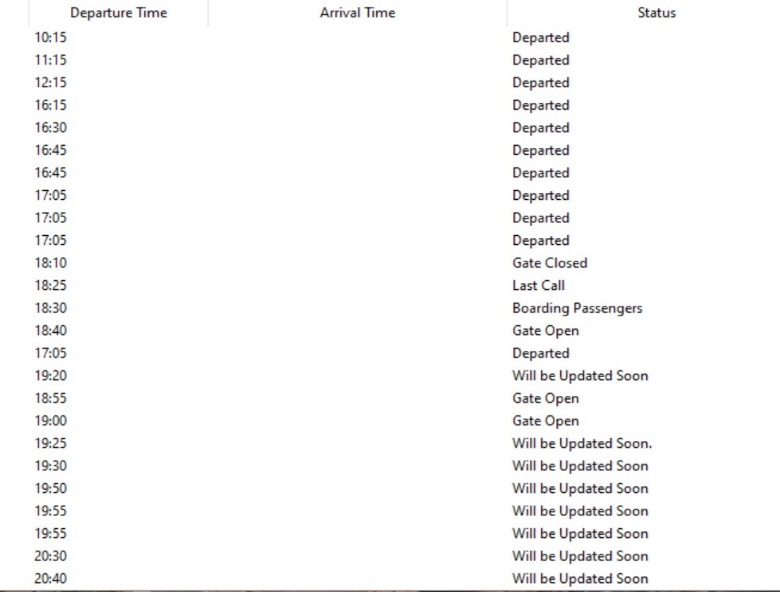




Real-Time for Arrival & Departure Time.

I’m also created a Real-Time Arrival & Departure Time to let user know what time does their flight will boarding or departs this will make user to plan their journey more effectively and reduce stress.

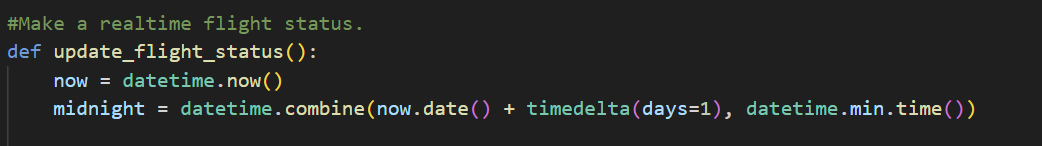
This is the real – time of Departures in Departure flight data:

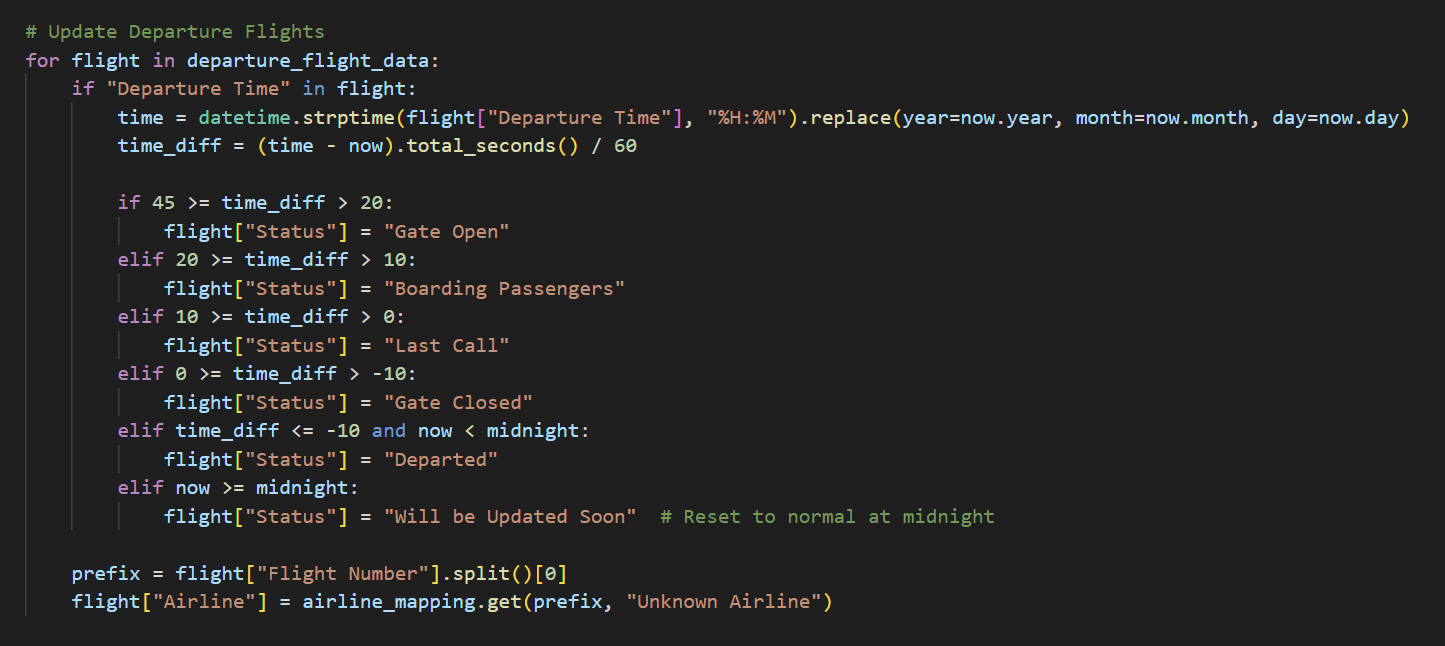


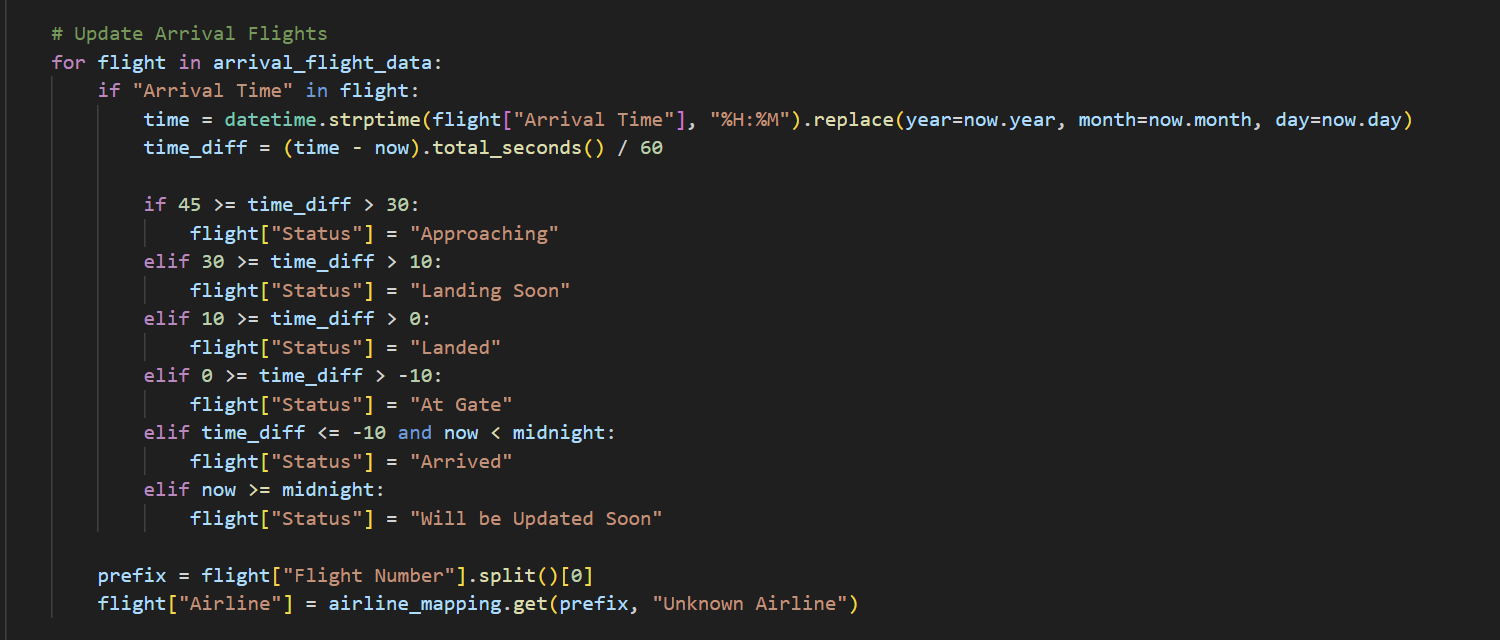
This is the real – time of Arrivals in the Arrival flight data:



To make the real-time on Departure and Arrival time I make the code functions:







Before I make the code using the for loop for flight functions, I make the now and midnight code. For the now code I put the ‘datetime.now()’ this is to retrieves the current date and time of the system at the moment this line of code is executed.

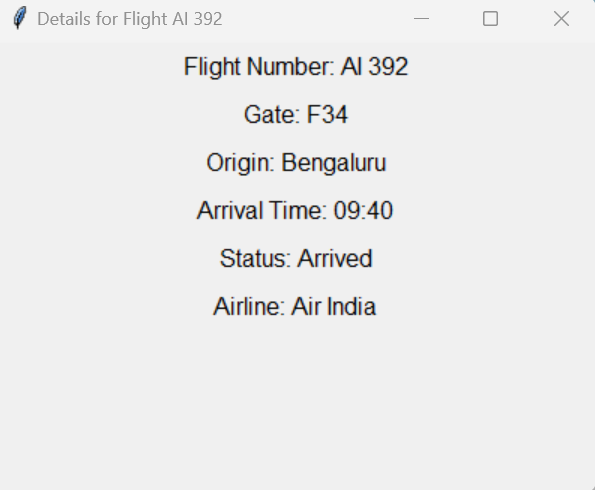
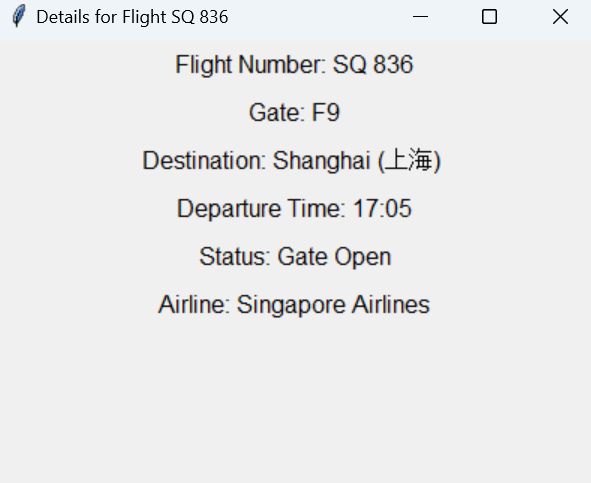
Whereas the midnight code I put ‘datetime.combine(now.date() + timedelta(days=1), datetime.min.time()). The now.date() + timedelta(days=1) use to combine the date of the next day. Whereas the datetime.min.time() combine The time at midnight (00:00:00). This variable is used to represent the exact moment when the next day starts at 12 AM. It’s very useful in scenarios where actions or updates need to occur based on the start of a new day.

So each departure time and arrival time I simply make the for loop and make the if function inside the for loop. I use the for loop because the code operates on a collection of departure\_flight\_data as well as on the arrival\_flight\_data. Which is a list that contains a data. Each item represents a single flight, and the goal is to process and update the status of each flight individually. I also put the if function in side the for loop on each arrival and departure Flights because each departure and arrival flights needs to be checked individually on what its current status should be based on the remaining time until departure and what airline corresponds to its flight number prefix.

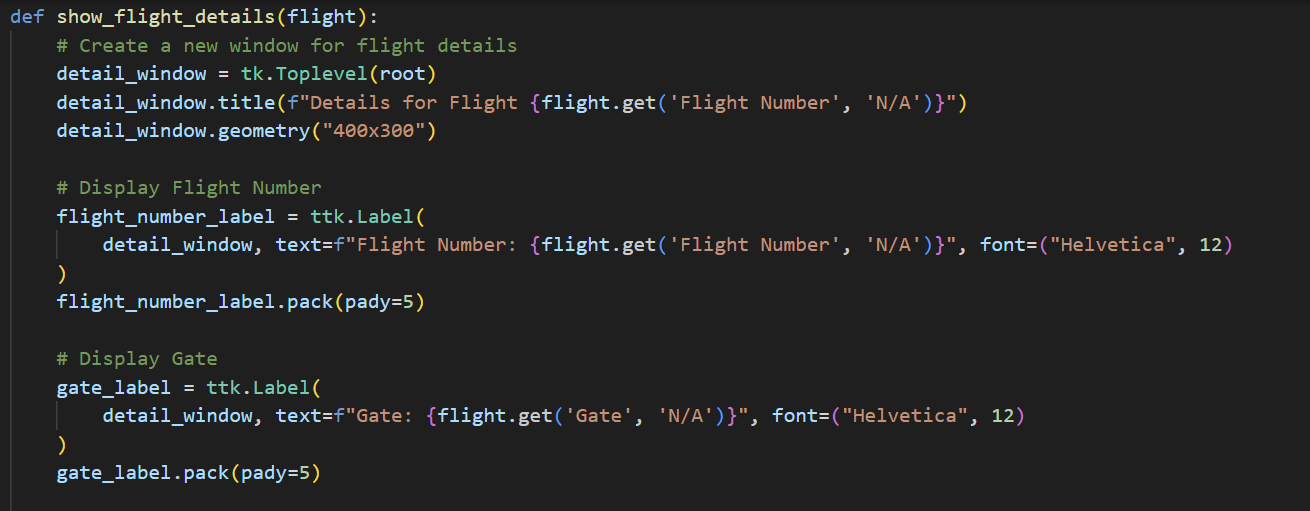
The time\_diff use to calculate the”time’difference between the current time and the flight departure time as well as the arrival time.

Showing The Flight Details

When user clicks each flight number it will pop up the more details for each flight number. Here’s the result.:



To make it work I make a code for the flight details. Like this:



I use the define function (def) and put flight for argument which is expected to be a dictionary containing flight details (e.g. Flight Number, gate, destination, etc.). it is used to open a new window and display these details.

For the ‘detail\_window’, I’ve input the ‘tk.Toplevel(root)’ it create a new top-level window within the main application window(root) This allows the flight details to be shown in a separate popup window. I’ve also input the ‘detail\_window.title’ and write the title of the window something like :”Details for Flight” Then I use the ‘flight.get(‘Flight Number’, ‘N/A’)’ to fetch the flight number from the flight dictionary. If the key is missing, I set the default to “N/A”. and lastly I want to set the size of the window to 400 pixels wide and 300 pixels tall by input the ‘detail\_window’ and add ‘.geometry(“400x300”)’.

To display the Flight Number, I create a code like this:

flight\_number\_label = ttk.Label(

detail\_window,

text=f”Flight Number: {flight.get(‘Flight Number’, ‘N/A’)}”,

font=(“Helvetica”, 12)

)

flight\_number\_label.pack(pady=5)

I put the flight number in the format “Flight Number: {Flight Number’, ‘N/A’}” so if the flight number is unavailable, I set to default to ‘N/A’. else, it will show the available Flight Number.

I also create a code to display the gate , Like This:

gate\_label = ttk.Label(

detail\_window,

text=f”Gate: {flight.get(‘Gate’, ‘N/A’)}”,

font=(“Helvetica”, 12)

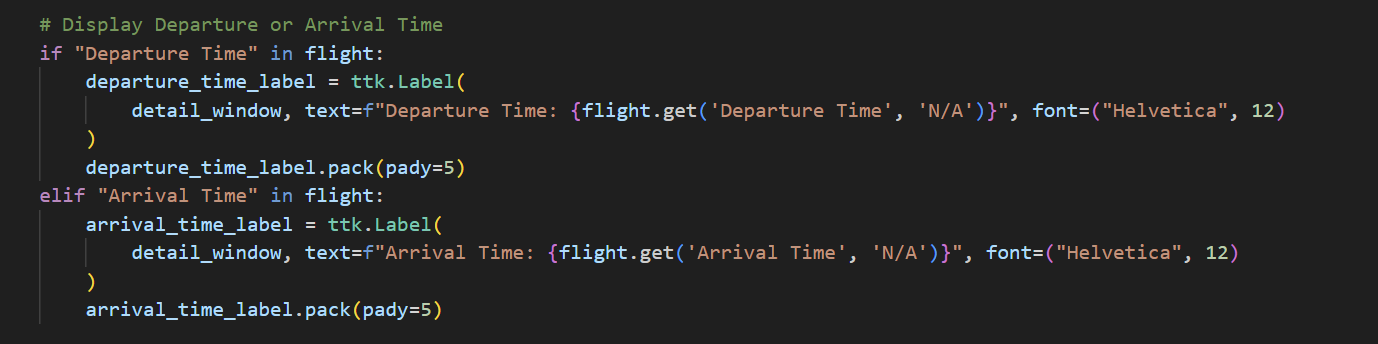
)

gate\_label.pack(pady=5)

I put the gate in the format ”Gate: {‘Gate’, ‘N/A’}” it will default to ‘N/A’ if the Gate is not available. I’m also make a font of ‘Helvetica’ and set the font size to 12.

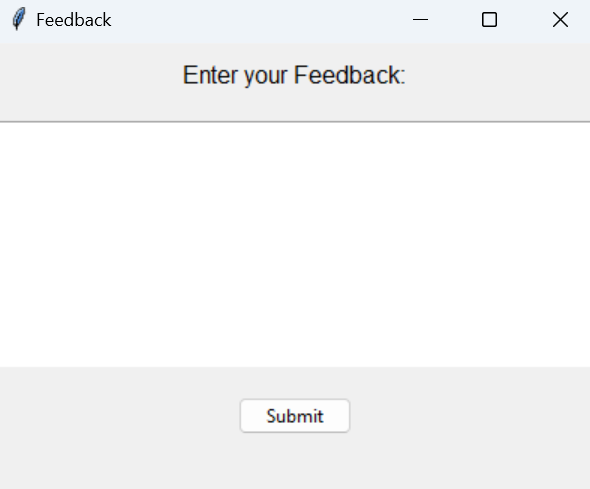
To Display the Destination or Origin I use the if function so it can ensure only Destination or Origin is shown. Like this:

  
I use the if function on the Destination and Origin flight to Check if the flight dictionary contains a “Destination” key. If it exists, it displays the destination. If “Destination” is not found but “Origin” exists, it displays the origin. So it can ensures only one of these is shown. The if function ensures that the program correctly displays Destination for outbound flights and Origin for inbound flights, depending on the flight type.



I also use the if function on the “Departure Time” and “Arrival Time” so if the flight dictionary contains the key “Departure Time” It will Displays the departure time. If the flight dictionary contain the key “Arrival Time” It will Displays the arrival time. And It can ensures only one of these is shown.

Feedback Page



To make user more convenient I create a feedback page for users to share their thoughts, opinions, and suggestions about a product, service, or experience. It allows users to share their improvement that we need to solve like solving a problem. And so on.

To make the feedback form I make the def function like this:



I used the def function a function help to organize the code by breaking it into smaller, reusable chunks. I can easily call the function (show\_feedback\_form) whenever I need to display the feedback form. I also can use this ame function in different parts of the program so I don’t habe to rewriting the feedback form code at each time. By using this function, it can make the code cleaner and more readable. By encapsulating the feedback form creation in a function, the main program remains concise and easier to follow.

def show\_feedback\_form(): This code defines a new function called show\_feedback\_form. Everything indented under this line is part of the function.

The Global Variables “global feedback\_window, feedback\_text” declares that feedback\_window and feedback\_text are global variables. This means that they can be accessed and modified outside the function.

To create a new window I make the feedback\_window code as well as the title & geometry :

feedback\_window = tk.Toplevel(root)

feedback\_window.title("Feedback")

feedback\_window.geometry("400x300")

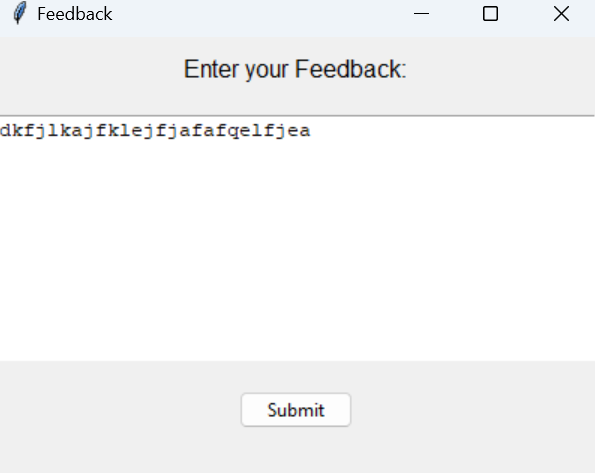
This code create a new top-level window as a child of the main window (root). The window is titled "Feedback" and given a size of 400x300 pixels.

And also to add a new label on the feedback, I make a feedback\_label code:

feedback\_label = ttk.Label(feedback\_window, text="Enter your Feedback:", font=("Helvetica", 12))

feedback\_label.pack(pady=10)

This creates a label widget with the text "Enter your Feedback:" and a font size of 12. The pack method is used to add the label to the feedback\_window, with 10 pixels of padding (pady=10) above and below it.



To make a Text box like this I use a code :

feedback\_text = tk.Text(feedback\_window, width=50, height=10)

feedback\_text.pack(pady=10)

By making this code It creates a text box where users can type their feedback. It is 50 characters wide and 10 lines tall. The pack method is again used to add the text box to the feedback\_window, with 10 pixels of vertical padding.

On the image above I’m also add a submit button to make it possible I make the submit\_button code below

submit\_button = ttk.Button(feedback\_window, text="Submit", command=submit\_feedback)

submit\_button.pack(pady=10)

This line creates a button with the text "Submit". When clicked, the button will call the submit\_feedback function. The pack method is used to add the button to the feedback\_window, with 10 pixels of vertical padding.

Conclusion

In conclusion, This Flight information that I made is to make user more easier to look their flight status anytime anywhere. To make user more convenient, I create a feedback page for users to share their thoughts, opinions, and suggestions about a product, service, or experience. I hope this will happen in the future.