

Configuration Manual

MSc Research Project
Utilizing NLP and Hybrid Recommender Approach to Enhancing
Personalized Travel Planning

Shih-Ting Huang
Student ID: x23266287

School of Computing
National College of Ireland

Supervisor: Victor del Rosal

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Shih-Ting, Huang
Student ID: X23266287
Programme: MSc in Artificial Intelligence for Business **Year:** 2025
Module: MSc (Research) Practicum/Internship Part 2
Lecturer: Victor del Rosal
Submission Due Date: Utilizing NLP and Hybrid Recommender Approach to Enhancing Personalized Travel Planning
Project Title: 283 8
Word Count: **Page Count:**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shih-Ting, Huang
Student ID: X23266287

1 Introduction

The configuration manual provides a walk-through of the research study setup, including system requirements and implementation. Both of them would be described below.

2 System Requirements

2.1 Hardware Specification

- Device name: MacBook Air
- Processor: Apple M2
- Installed RAM: 16 GB
- Memory: 16GB unified memory

2.2 Software Specification

- Programming Language: Python
- Libraries Used:

Data Handling and Analysis

- pandas: Data manipulation and analysis.
- json: Parse and serialize JSON (JavaScript Object Notation) data.
- argparse: Handle command-line arguments.

Machine Learning / NLP

- sklearn.feature_extraction.text.TfidfVectorizer: Convert a collection of text documents to a matrix of TF-IDF features.
- sklearn.metrics.pairwise.cosine_similarity: Measure similarity between TF-IDF vectors.
- spacy: Advanced Natural Language Processing (NLP).
- re: Regular expressions for text pattern matching and cleaning.
- dateparser: Parse human-readable date strings into datetime objects.

Web and API Communication

- requests: Make HTTP requests to APIs.
- urllib.parse.quote: URL-encode strings for safe inclusion in a URL.

OpenAI Integration

- openai: Interface with OpenAI API (e.g., ChatGPT, DALL·E, Whisper).

Time and Date Utilities

- time: Track and manage delays, timeouts, or timestamps.
- datetime, timedelta: Work with dates and time spans.

Web App Development

- flask: Lightweight web framework for building APIs or web services.

- flask.request, flask.jsonify: Handle HTTP requests and return JSON responses in Flask.
- flask_cors.CORS: Allow cross-origin resource sharing (CORS) in web APIs.

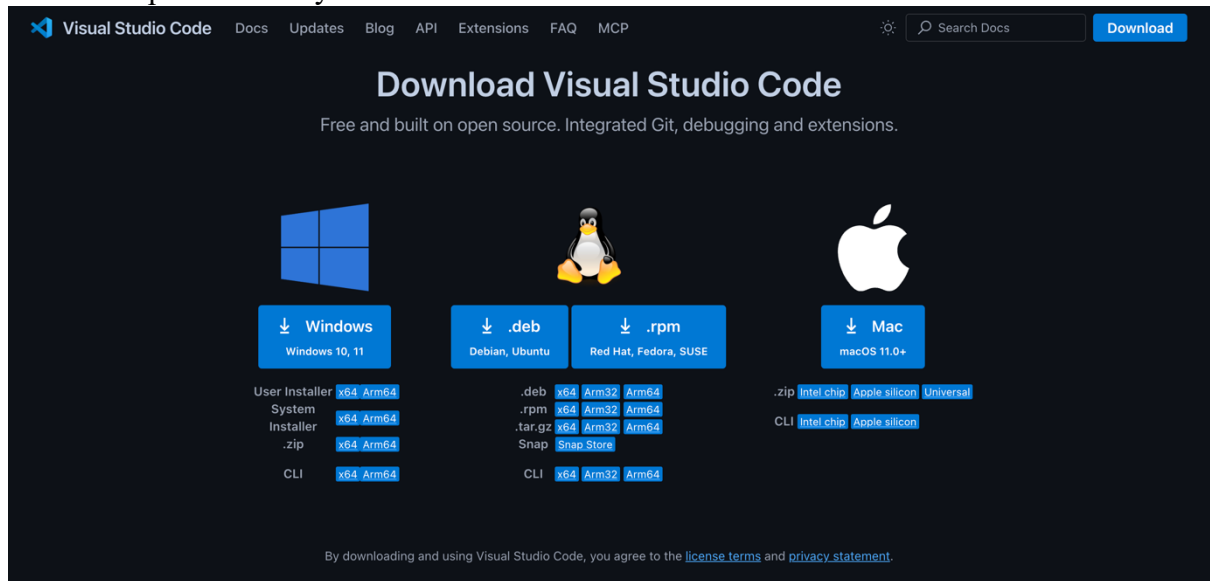
System Tools and Logging

- sys: Access command-line args, manipulate Python runtime behavior.
- subprocess: Run external programs or commands from Python.
- logging: Log events for debugging or monitoring.
- Development Environment: Visual Studio Code

3 Implementation

3.1 Installing Visual Studio Code

Click the option that fit your environment and download



3.2 Code Description

Travel_Recommender.py: Extract travel related information from user input.

```
Departure City: NYC, Arrival Country: Mexico, Arrival City: Mexico City, Travel Period: 20250819 - 20250821, Travel Days: 3 days, Budget: 500 euros, Preferences: None [] Mexico City ('20250819', '20250821') Mexico City NYC 500.0 None
```

chatbot.html: The interference for user to interact with.

Travel Recommender Chatbot

Flight: BR112 | Price: 201.00 USD | Flight: BR112 |
 Airline: BR | TPE->OKA | Dep: 2025-08-19T06:45:00
 | Arr: 2025-08-19T09:15:00 | Price: 224.63 USD
 Flight: JL8664 | Airline: JL | TPE->NRT | Dep: 2025-
 08-19T12:40:00 | Arr: 2025-08-19T17:05:00 | Price:
 264.40 USD

I live in Taipei and want to go to Okinawa for 3 days travel, start from 20250819, my budget is around 1000 euro, can you help me organize this trip?

Departure City: Taipei, Arrival Country: Japan, Arrival City: Okinawa, Travel Period: 20250819 - 20250821, Travel Days: 3 days, Budget: 1000 euros, Preferences: None Attraction Recommendation from Google: A-side Smash Burger Okinawa World Hiipii Beach Campground (Happiness Marine Club) Exchange San-A Naha Main Place AEON Naha Yakena Strait Observatory Cape Kyan CLUB CLUTCH Bokunen Museum Lips Tamaudun Gallery On The Go with Annalisa Okinawa Fruits Land Lane to Central Attraction Recommendation from GPT: Shurejo Castle Kakumai Street Okinawa Churaumi

chat_backend.py: Use for connecting front end and all the APIs.
 Attractions.py: For collecting attraction data from Google API.

```

INFO:root:Collecting data for Okinawa with type 'tourist_attraction'
INFO:root:Collecting data for Okinawa with type 'museum'
INFO:root:Collecting data for Okinawa with type 'amusement_park'
INFO:root:Collecting data for Okinawa with type 'aquarium'
INFO:root:Collecting data for Okinawa with type 'zoo'
INFO:root:Collecting data for Okinawa with type 'park'
INFO:root:Collecting data for Okinawa with type 'stadium'
INFO:root:Collecting data for Okinawa with type 'art_gallery'
INFO:root:Collecting data for Okinawa with type 'church'
INFO:root:Collecting data for Okinawa with type 'mosque'
INFO:root:Collecting data for Okinawa with type 'hindu_temple'
INFO:root:Collecting data for Okinawa with type 'synagogue'
INFO:root:Collecting data for Okinawa with type 'library'
INFO:root:Collecting data for Okinawa with type 'casino'
INFO:root:Collecting data for Okinawa with type 'night_club'
INFO:root:Collecting data for Okinawa with type 'movie_theater'
INFO:root:Collecting data for Okinawa with type 'bowling_alley'
INFO:root:Collecting data for Okinawa with type 'campground'
INFO:root:Collecting data for Okinawa with type 'natural_feature'
INFO:root:Collecting data for Okinawa with type 'shopping_mall'
INFO:root:Collecting data for Okinawa with type 'historical_place'
INFO:root:Collecting data for Okinawa with type 'point_of_interest'
INFO:root:Collecting data for Okinawa with type 'establishment'
INFO:root:Collecting data for Okinawa with type 'landmark'
  
```

Traveler_Review.py: For collecting traveler review from Google API.

```
INFO:root:Collecting data for Croatia with type 'tourist_attraction'
INFO:root:Collecting data for Croatia with type 'museum'
INFO:root:Collecting data for Croatia with type 'amusement_park'
INFO:root:Collecting data for Croatia with type 'aquarium'
INFO:root:Collecting data for Croatia with type 'zoo'
INFO:root:Collecting data for Ireland with type 'tourist_attraction'
INFO:root:Collecting data for Ireland with type 'museum'
INFO:root:Collecting data for Ireland with type 'amusement_park'
INFO:root:Collecting data for Ireland with type 'aquarium'
INFO:root:Collecting data for Ireland with type 'zoo'
INFO:root:Collecting data for Ireland with type 'park'
INFO:root:Collecting data for Ireland with type 'stadium'
INFO:root:Collecting data for Ireland with type 'art_gallery'
INFO:root:Collecting data for Ireland with type 'church'
INFO:root:Collecting data for Ireland with type 'mosque'
INFO:root:Collecting data for Ireland with type 'hindu_temple'
INFO:root:Collecting data for Ireland with type 'synagogue'
```

Flight_data_Amaduse.py: For collecting flight data from Amadeus API.

```
Found flights: [{'Flight Number': 'BR156', 'Airline': 'BR', 'Departure Airport': 'TSA', 'Arrival Airport': 'GMP', 'Departure Time': '2025-08-19T14:40:00', 'Arrival Time': '2025-08-19T18:10:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '356.30'}, {'Flight Number': 'JL98', 'Airline': 'JL', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T14:20:00', 'Arrival Time': '2025-08-19T18:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '360.40'}, {'Flight Number': 'JL96', 'Airline': 'JL', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T09:10:00', 'Arrival Time': '2025-08-19T13:20:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '360.74'}, {'Flight Number': 'JL98', 'Airline': 'JL', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T14:20:00', 'Arrival Time': '2025-08-19T18:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '360.74'}, {'Flight Number': 'JL96', 'Airline': 'JL', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T09:10:00', 'Arrival Time': '2025-08-19T13:20:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '360.74'}, {'Flight Number': 'JL98', 'Airline': 'JL', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T14:20:00', 'Arrival Time': '2025-08-19T18:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '360.74'}, {'Flight Number': 'NH852', 'Airline': 'NH', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T13:35:00', 'Arrival Time': '2025-08-19T17:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '396.31'}, {'Flight Number': 'NH852', 'Airline': 'NH', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T13:35:00', 'Arrival Time': '2025-08-19T17:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '396.31'}, {'Flight Number': 'NH854', 'Airline': 'NH', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T16:50:00', 'Arrival Time': '2025-08-19T20:50:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '396.31'}, {'Flight Number': 'NH854', 'Airline': 'NH', 'Departure Airport': 'TSA', 'Arrival Airport': 'HND', 'Departure Time': '2025-08-19T16:50:00', 'Arrival Time': '2025-08-19T20:50:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '396.31'}]
Found flights: [{'Flight Number': 'VZ568', 'Airline': 'HR', 'Departure Airport': 'TPE', 'Arrival Airport': 'OKA', 'Departure Time': '2025-08-19T14:45:00', 'Arrival Time': '2025-08-19T17:05:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '110.91'}, {'Flight Number': '7C6102', 'Airline': '7C', 'Departure Airport': 'TPE', 'Arrival Airport': 'ICN', 'Departure Time': '2025-08-19T11:50:00', 'Arrival Time': '2025-08-19T15:15:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '161.89'}, {'Flight Number': 'HX283', 'Airline': 'HR', 'Departure Airport': 'TPE', 'Arrival Airport': 'HKG', 'Departure Time': '2025-08-19T21:45:00', 'Arrival Time': '2025-08-19T23:45:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '162.46'}, {'Flight Number': 'HX285', 'Airline': 'HR', 'Departure Airport': 'TPE', 'Arrival Airport': 'HKG', 'Departure Time': '2025-08-19T20:15:00', 'Arrival Time': '2025-08-19T22:10:00', 'Flight Status': 'Scheduled', 'Ticket Price (USD)': '162.46'}, {'Flight Number': 'HX283', 'Airline': 'HR'}
```

Weather_data.py: For collecting weather information from open weather API.

```

2025-08-19 | few clouds | Temp: 28.6°C | Humidity: 83%
2025-08-20 | few clouds | Temp: 28.6°C | Humidity: 83%
2025-08-21 | few clouds | Temp: 28.6°C | Humidity: 83%
Found flights: [{'Flight Number': 'BR156', 'Airline': 'BR

```

travel_info_analyzer.py: Collect attraction data from OpenAI.

```

{
  "Day": 1,
  "Name": "Shurijo Castle",
  "Address": "1 Chome-2 Shurikinjocho, Naha, Okinawa 903-0815, Japan",
  "Latitude": "26.2124",
  "Longitude": "127.7196",
  "Price Level": "Moderate",
  "Availability": "Open",
  "Country": "Japan",
  "Type": "Historical Site"
},
{
  "Day": 1,
  "Name": "Kokusai Street",
  "Address": "3 Chome-10-1 Makishi, Naha, Okinawa 900-0013, Japan",
  "Latitude": "26.2123",
  "Longitude": "127.6795",
  "Price Level": "Low",
  "Availability": "Open",
  "Country": "Japan",
  "Type": "Shopping District"
},
{
  "Day": 1,
  "Name": "Okinawa Churaumi Aquarium",
  "Address": "424 Ishikawa, Motobu, Kunigami District, Okinawa 905-0206, Japan",
  "Latitude": "26.6947",
  "Longitude": "127.8770",
  "Price Level": "Moderate",
  "Availability": "Open",
  "Country": "Japan",
  "Type": "Aquarium"
}

```

Recommender_Analysis.py: Use hybrid recommendation algorithm to provide attraction data collected from Google.

	Name	...	Place_ID
0	A-side Smash Burger	...	ChIJZ8VPQwAT5TQRtBOQvS0MQ08
1	Okinawa World	...	ChIJza_GaWlv5TQRy52qkW4L2P4
2	Hiipii Beach Campground (Happiness Marine Club)	...	ChIJNWcEKfwq5TQRYMWe4GuFfB4
3	Exchange	...	ChIJW7nKfjwS5TQRX_nk8xzHtzA
4	San-A Naha Main Place	...	ChIJFwVGNpr5TQRWa4uFVWLhxx
5	AEON Naha	...	ChIJnyJKV7Fp5TQRHWdGau5JmQk
6	Yakena Strait Observatory	...	ChIJSWAg11sJ5TQRLTFzzXVxbzk
7	Cape Kyan	...	ChIJy0UrTzBn5TQRrBNMV2bXBns
8	CLUB CLUTCH	...	ChIJVVUlu2Rp5TQRf2cZECy7ztQ
9	Bokunen Museum	...	ChIJ02AM-Q8T5TQRpczm9aZRRcw
10	Lips	...	ChIJwe3oKAAT5TQRWUBXYqZ-RRM
11	Tamaudun	...	ChIJLZXj0P1r5TQRTvnj1SMSPTs
12	Gallery On The Go with Annalisa	...	ChIJCddnizIF5TQRJhzuA9Q55-M
13	Okinawa Fruits Land	...	ChIJ__8_i1L_5DQR7PrCr2WtP8E
14	Lane to Central	...	ChIJ0WY1eWQS5TQRZ3mKRDiqCmc

merge_attractions.py: Combine all the collected travel information.

呼叫 merge_attractions.py 合併景點資料 ...

Merged 24 attractions to /Users/sarenah/Documents/ST/愛爾蘭/Practicum/temp/attractions_data_merged.json

airport_iata_lookup.py: Use for switch city to IATA code.

```
1,"Goroka Airport","Goroka","Papua New Guinea","GKA","AYCA",-6.081688834580001,145.391998291,5282,10,"U","Pacific/Port Moresby","airport","OurAirports"
2,"Madang Airport","Madang","Papua New Guinea","MAG","AYMD",-5.28707988739,145.789001465,20,10,"U","Pacific/Port Moresby","airport","OurAirports"
3,"Mount Hagen Kagamuga Airport","Mount Hagen","Papua New Guinea","HGU","AYMH",-5.826789855957031,144.29600524902344,5388,10,"U","Pacific/Port Moresby","airport","OurAirports"
4,"Nadzab Airport","Nadzab","Papua New Guinea","LAE","AYNZ",-6.569803,146.725977,239,10,"U","Pacific/Port Moresby","airport","OurAirports"
5,"Port Moresby Jacksons International Airport","Port Moresby","Papua New Guinea","POM","AYPY",-5.443380355834961,147.22000122070312,146,10,"U","Pacific/Port Moresby","airport","OurAirports"
6,"Wewak International Airport","Wewak","Papua New Guinea","WMK","AYWK",-3.58383011818,143.660006348,19,10,"U","Pacific/Port Moresby","airport","OurAirports"
7,"Narsarsuaq Airport","Narsarsuaq","Greenland","UAK","BGBM",61.1604995728,-45.4259986877,112,-3,"E","America/Godthab","airport","OurAirports"
8,"Godthaab / Nuuk Airport","Godthaab","Greenland","GOH","BGGH",64.19090271,-51.6781005859,283,-3,"E","America/Godthab","airport","OurAirports"
9,"Kangerlussuaq Airport","Sondrestrom","Greenland","SFJ","BGSF",67.0122218992,-50.7116031647,165,-3,"E","America/Godthab","airport","OurAirports"
10,"Thule Air Base","Thule","Greenland","THU","BGTU",76.5311965942,-68.7032012939,251,-4,"E","America/Thule","airport","OurAirports"
11,"Akureyri Airport","Akureyri","Iceland","AEY","BIAR",65.66000366210930,-18.07270050048928,6,0,"N","Atlantic/Reykjavik","airport","OurAirports"
12,"Egilsstaðir Airport","Egilsstaðir","Iceland","EGS","BIEG",65.2833023071289,-14.401399612426758,76,0,"N","Atlantic/Reykjavik","airport","OurAirports"
13,"Hornafjörður Airport","Hofn","Iceland","HFN","BIHN",64.295601,-15.2272,24,0,"N","Atlantic/Reykjavik","airport","OurAirports"
14,"Húsavík Airport","Husavik","Iceland","HZK","BIHU",65.952301,-17.426001,48,0,"N","Atlantic/Reykjavik","airport","OurAirports"
15,"Ísafjörður Airport","Isafjörður","Iceland","IFJ","BIIS",66.058009783935547,-23.135299682617188,8,0,"N","Atlantic/Reykjavik","airport","OurAirports"
16,"Keflavík International Airport","Keflavík","Iceland","KEF","BIKF",63.985000610352,-22.605600357056,171,0,"N","Atlantic/Reykjavik","airport","OurAirports"
17,"Patreksfjörður Airport","Patreksfjörður","Iceland","PFJ","BIPA",65.555801,-23.965,11,0,"N","Atlantic/Reykjavik","airport","OurAirports"
18,"Reykjavík Airport","Reykjavík","Iceland","RKV","BIRK",64.1299972534,-21.9405994415,48,0,"N","Atlantic/Reykjavik","airport","OurAirports"
19,"Siglufjörður Airport","Siglufjörður","Iceland","SIJ","BISI",66.133301,-18.9167,10,0,"N","Atlantic/Reykjavik","airport","OurAirports"
20,"Vestmannaeyjar Airport","Vestmannaeyjar","Iceland","VEY","BIWM",63.42430114746094,-20.278900146484375,326,0,"N","Atlantic/Reykjavik","airport","OurAirports"
21,"Sault Ste Marie Airport","Sault Sainte Marie","Canada","YAM","CYAM",46.48500061035156,-84.5093994140625,630,-5,"A","America/Toronto","airport","OurAirports"
22,"Winnipeg / St. Andrews Airport","Winnipeg","Canada","WN","CYAW",50.0564002991,-97.03259122070001,760,-6,"A","America/Winnipeg","airport","OurAirports"
23,"Halifax / CFB Shearwater Heliport","Halifax","Canada","YN","CYAM",44.639702,-63.499401,144,-4,"A","America/Halifax","airport","OurAirports"
24,"St. Anthony Airport","St. Anthony","Canada","YAY","CYAY",51.3918991089,-56.083099365200006,108,-3,5,"A","America/St Johns","airport","OurAirports"
25,"Tofino / Long Beach Airport","Tofino","Canada","YAZ","CYAZ",49.079833,-125.775583,80,-8,"A","America/Vancouver","airport","OurAirports"
26,"Kuguaruk Airport","Pelly Bay","Canada","YBB","CYBB",68.534401,-99.888009,156,-7,"A","America/Edmonton","airport","OurAirports"
27,"Baie Comeau Airport","Baie Comeau","Canada","YBC","CYBC",49.13249969482422,-68.20439910888672,71,-5,"A","America/Toronto","airport","OurAirports"
28,"CFB Bagotville","Bagotville","Canada","YBG","CYBG",48.33060073852539,-70.99639892578125,522,-5,"A","America/Toronto","airport","OurAirports"
29,"Baker Lake Airport","Baker Lake","Canada","YBK","CYBK",64.29889678960001,-96.077796936,59,-6,"A","America/Winnipeg","airport","OurAirports"
30,"Campbell River Airport","Campbell River","Canada","YBL","CYBL",49.950801849365234,-125.27100372314453,346,-8,"A","America/Vancouver","airport","OurAirports"
31,"Brandon Municipal Airport","Brandon","Canada","YBR","CYBR",49.91,-99.951897,1343,-6,"A","America/Winnipeg","airport","OurAirports"
32,"Cambridge Bay Airport","Cambridge Bay","Canada","YCB","CYCB",69.1081000911,-105.138000468,90,-7,"A","America/Edmonton","airport","OurAirports"
33,"Nanaimo Airport","Nanaimo","Canada","YCD","CYCD",49.054970224899996,-123.869862556,92,-8,"A","America/Vancouver","airport","OurAirports"
34,"Castlegar/Mest Kootenay Regional Airport","Castlegar","Canada","YCG","CYCG",49.2963981628,-117.632003784,1624,-8,"A","America/Vancouver","airport","OurAirports"
35,"Miramichi Airport","Chatham","Canada","YCH","CYCH",47.007801,-65.449203,108,-4,"A","America/Halifax","airport","OurAirports"
36,"Charlo Airport","Charlo","Canada","YCL","CYCL",47.990799,-66.330299,132,-4,"A","America/Halifax","airport","OurAirports"
37,"Kugluktuk Airport","Coppermine","Canada","YCO","CYCO",67.816704,-115.143997,74,-7,"A","America/Edmonton","airport","OurAirports"
```

Itinerary_organize.py: Organize all collected data and generate itinerary.

```
{
  "itinerary": [
    {
      "day": 1,
      "date": "2025-08-19",
      "activities": [
        {
          "time": "09:00-12:00",
          "activity": "Visit Okinawa World",
          "attraction": {
            "name": "Okinawa World",
            "address": "1336 Tamagusuku Maekawa, Nanjo, Okinawa 901-0016, Japan",
            "rating": 4.3,
            "latitude": 26.1444,
            "longitude": 127.7302,
            "estimated_cost": "€15"
          },
          "transportation": {
            "mode": "Taxi",
            "duration": "30 mins",
            "cost": "€10"
          },
          "restaurant": {
            "name": "A-side Smash Burger",
            "address": "1 Chome-10-15 1F, Okinawa, Japan",
            "estimated_cost": "€20"
          }
        },
        {
          "time": "14:00-18:00",
          "activity": "Visit Shikinaen Garden",
          "attraction": {
            "name": "Shikinaen Garden",
            "address": "421-7 Maaji, Naha, Okinawa 902-0072, Japan",
            "rating": 4.1,
            "latitude": 26.1898,
            "longitude": 127.6693,
            "estimated_cost": "€10"
          },
          "transportation": {
            "mode": "Bus",
            "duration": "1 hour",
            "cost": "€5"
          },
          "restaurant": {
            "name": "Exchange",
            "address": "Kadena Air Base, Okinawa, Japan",
            "estimated_cost": "€25"
          }
        }
      ],
      "accommodation": {
        "name": "Luxury Hotel Okinawa",
        "address": "1 Chome-1-2 Yogi, Naha, Okinawa 902-0076, Japan",
        "estimated_cost": "€100"
      },
      "flight": {
        "flight_number": "BR112",
        "departure": "06:45",
        "arrival": "09:15"
      }
    }
  ]
}
```

3.3 Installing Libraries

```
import requests
import logging
import time
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer

import numpy as np
import pandas as pd
import requests
import logging
import time
from urllib.parse import quote
import sys
import openai
import spacy
import re
import subprocess
import json
import logging
import sys

import openai
import pandas as pd
from datetime import datetime, timedelta
import io
import time
import json
import argparse

from flask import Flask, request, jsonify
from flask_cors import CORS
import sys
```

3.4 Loading the API

```
def get_attractions(city, place_type):
    """
    Fetches attractions in a given city using Google Places API.
    """
    base_url = "https://maps.googleapis.com/maps/api/place/textsearch/json"
    encoded_query = quote(f"{place_type} in {city}") # Ensure proper encoding
    url = f"{base_url}?query={encoded_query}&key={API_KEY}"
```

```
# ♦ Initialize Amadeus API Client
amadeus = Client(
    client_id=AMADEUS_API_KEY,
    client_secret=AMADEUS_API_SECRET
)

# ♦ Get Flight Information & Ticket Prices
def get_flights(departure_airport, arrival_airport, departure_date):
    try:
        response = amadeus.shopping.flight_offers_search.get(
            originLocationCode=departure_airport,
            destinationLocationCode=arrival_airport,
            departureDate=departure_date,
            adults=1, # Number of passengers
            max=10 # Number of flight results
        )
```

```
start_dt = datetime.strptime(normalize_date(start_date), "%Y-%m-%d")
end_dt = datetime.strptime(normalize_date(end_date), "%Y-%m-%d")
num_days = (end_dt - start_dt).days + 1
weather_info = []
for i in range(num_days):
    date = (start_dt + timedelta(days=i)).strftime("%Y-%m-%d")
    base_url = f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}&units=metric"
    try:
        response = requests.get(base_url)
        if response.status_code == 200:
            data = response.json()
            weather_info.append({
                "Date": date,
                "City": data.get("name", city),
                "Weather": data["weather"][0]["description"] if "weather" in data and data["weather"] else "",
                "Temperature (°C)": data["main"]["temp"] if "main" in data else "",
                "Feels Like (°C)": data["main"]["feels_like"] if "main" in data else "",
                "Min Temperature (°C)": data["main"]["temp_min"] if "main" in data else "",
                "Max Temperature (°C)": data["main"]["temp_max"] if "main" in data else "",
                "Humidity (%)": data["main"]["humidity"] if "main" in data else "",
                "Wind Speed (m/s)": data["wind"]["speed"] if "wind" in data else "",
            })
```

```
openai.api_key =
```

3.5 Run the Program

Run chat_backend.py

```
[(base) sarenah@Mac temp % python chat_backend.py
```

```
Index: []
* Serving Flask app 'chat_backend'
* Debug mode: on
INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment
. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5001
* Running on http://192.168.0.160:5001
INFO:werkzeug:Press CTRL+C to quit
INFO:werkzeug: * Restarting with watchdog (fsevents)
```

Run chatbot.html

Travel Recommender Chatbot

19110.45.00 | Price: 201.03 USD | Flight: BR112 |
Airline: BR | TPE->OKA | Dep: 2025-08-19T06:45:00
| Arr: 2025-08-19T09:15:00 | Price: 224.63 USD
Flight: JL8664 | Airline: JL | TPE->NRT | Dep: 2025-
08-19T12:40:00 | Arr: 2025-08-19T17:05:00 | Price:
264.40 USD

I live in Taipei and want to go to Okinawa for 3 days
travel, start from 20250819, my budget is around
1000 euro, can you help me organize this trip?

Departure City: Taipei, Arrival Country: Japan, Arrival
City: Okinawa, Travel Period: 20250819 - 20250821,
Travel Days: 3 days, Budget: 1000 euros,
Preferences: None Attraction Recommendation from
Google: A-side Smash Burger Okinawa World Hiipii
Beach Campground (Happiness Marine Club)
Exchange San-A Naha Main Place AEON Naha
Yakena Strait Observatory Cape Kyan CLUB
CLUTCH Bokunen Museum Lips Tamaudun Gallery
On The Go with Annalisa Okinawa Fruits Land Lane
to Central Attraction Recommendation from GPT:
Shurijo Castle, Kikunai Street, Okinawa, Churaumi

References

Akhitha Babu. (2021). "Install Visual Studio Code" Available at: <https://medium.com/nerd-for-tech/install-visual-studio-code-fe3908c5cf15> [Accessed on 30 July 2025].