Xiao Ting Chen, Emily Dang, Sophia Kardaras
Final Python Project
IST 256

Project Summary

The Problem

Deciding where to spend the next four years of your life is a big decision that many struggle with during senior year of high school. The college application process gets even more complicated when students need to sit down and compare all of their options through different avenues such as books and websites.

The Solution

The goal of this program is to help college students compare different colleges in one place. Factors such as tuition, SAT scores, average temperatures, location and school URL are listed in the output so students can compare all of their options. The program is designed so students and parents don't need to spend hours doing research on their own on different websites. This program utilizes multiple APIs to deliver the information students need to compare their prospects.

- **College Scorecard API:** https://collegescorecard.ed.gov/data/documentation/
- **Google Maps Geolocation API:** https://developers.google.com/maps/documentation/geolocation/intro
- **Weather Time Machine API:** https://darksky.net/dev/docs/time-machine

How does it work?

Students start off by being prompted with a menu list with the following options: A (add), R (remove), V (view) and Q (quit). If a user inputs A, then they will need to input a college name. If they input R, then they will need to input a college name that they previously entered. If they input V, then the list of colleges they entered into the program will be printed in a dataframe. If the user inputs Q, then the program will quit and stop asking the user for further inputs. Users can add as many colleges as they want into their school list. Each school that they enter is checked using the College Scorecard API. If the school is identified, then the geographic information is further identified in the Google Maps Geolocation API and the Weather Time Machine API. These APIs help gather information pertaining to SAT scores, weather, location and tuition rates.

Code Explanation

The first four lines import requests, json and pandas. We included the “from IPython.core.display import display” line to ensure that our data frame shows up nice and clean with lines like an excel sheet. The first function named **“college_check”** checks whether or not the inputted college name exists in the database. If the college name exists in the database, then it returns true. Otherwise it returns as false. The second function named **“college_score”** calls to the College Scorecard API. It includes an API key and the URL that is called to for college data retrieval. The third function named **“college_detail”** prints out the results and information pertaining to each college input. Based on the data that was available through College Scorecard, our group decided to retrieve information pertaining to the college name, SAT scores for reading, writing and mathematics, tuition cost, school URL, city, state, zip code and geographic coordinates for weather conditions. The data frame is structured with the following columns: college name, tuition, SAT(R), SAT(M), SAT(W), URL, City, State, Zip and Temp.

The fourth function named **“google_geocode”** calls to the Google Maps API to get the location of each college. The fifth function named **“darksky_weather”** calls to the Dark Sky Weather API to get the current weather conditions of each college. The sixth function named **“add_college”** allows users to add colleges to their college list. If the user enters a college that has already been added to the list, they will be told that the specific college is already in the list. The seventh function named, **“remove_college”** allows users to remove colleges to their college list. If they try to remove a college that is not on the list yet, they will be prompted with a message that says, “[college name] is not in the list.” The eighth function named **“list_menu”** prompts users with a list of different options they can choose from to create their personal college list. The list options include A for add college, R for remove college, V for view college list and Q for quit program. The main part of the program prompts users to input a college name. The program is able to handle uppercase and lowercase letters as input. If the user enters A, then the user is prompted to enter a college name. If the user enters a college name is recognized in the College Scorecard database, then the program informs the user with the following message, “Sorry, this college can NOT be found in the data set.” If the user enters R, then they are asked which college they would like to remove. If they input V, then the whole college list is printed out in a data frame. If the user inputs Q, then the program quits and stops asking the user for further inputs.

Responsibilities

Each person in the group had different strengths and weaknesses. We divided up the project responsibilities based on everyone's interests, strengths and weaknesses. Emily came up with the idea of College Collider based on the obstacles she saw high school students face during the college application process. While everyone worked on the actual python code portion of the project, Xiao Ting really took the lead. She is the most skilled python coder in the group so she helped take on the more challenging parts of the program. All of the written documents such as the proposal and the final document were written by Emily with help from Xiao Ting. Sophia took the lead on creating and designing the poster as she had a lot of experience from her advertising classes. Emily helped Sophia ensure that all of the details on the poster aligned with the code. All of the project tasks are broken down in the two tables below.

Task	**Who**
Python Code	Xiao Ting, Sophia, Emily
Written Report	Xiao Ting, Emily
Poster	Sophia, Emily

The table below breaks down who coded each specific part of the program.

Function Name (in order of appearance in code)	**Who**
college_check	Xiao Ting Chen
college_score	Emily Dang
college_detail	Xiao Ting Chen
google_geocode	Sophia Kardaras
darksky_weather	Sophia Kardaras
add_college	Emily Dang
remove_college	Xiao Ting Chen
list_menu	Emily Dang
Main Program	Xiao Ting Chen