Final Project Report

Project Code:

Github Repo: https://github.com/dengchaofan12/SI507_final_project.git

Package Required: bs4 requests time json os.path pandas flask JSONEncoder from json

README included in the repo

Data sources:

https://www.lakepedia.com/

https://www.lakepedia.com/continent/north-america-lakes.html

https://www.lakepedia.com/lake/michigan.html

This project focus on crawling multiple pages, all links are public, no authentication required. The first HTML link is the main page, and second, third HTML link is just an example. First link will have continent related links which leads to second page which includes all lakes in desired continent and relevant links, and these leads to third page showing detailed description of the lake.

There are 6 different links for second level, and each lead to dozens of links for third level. In total, there are 165 links to process.

In my program, first program should run is data_fetch.py which scan all three pages and cache relevant information into json file for future usage. There are two cached json files, first is final_project_source file which contains all HTML link source information used, second file is data_strucutre which contains all necessary information extracted from HTML links.

Data Structure:

My data structure is a tree, instead of a binary tree, my tree has multiple number of children nodes, and they are included in the list of the parent node. My tree has four layers, first layer is root node, and followed by continent nodes, then country nodes and next are lake nodes. Each node has three properties (Name, Children, Values), only at lake layer, there are values, values are information the program will present to user, name is the identification for every node, for example, continent will have “North America” or “Canada”, and children is a list of all its children node,

Root

Continent

Country

Lake (Values)

Tree is created by tree_generator.py included in the repo

Tree is stored in tree_structure.py under data_storage folder

The file that reads the tree json is user_interact_load_tree.py included in the repo

Interaction and presentation:

Instead of letting user click and search around webpages, this program can quickly locate the desired lake the user wants to see and display relevant information to user.

There are four questions user need to answer:

1. Which continent you are choosing?
2. Which Country you are choosing?
3. Which lake you are choosing?
4. Which format you want to display (text or flask table)?

In this project, I have used command line prompts and flask at data presentation. For every question, the list of options will be given, and user can either enter corresponding number or specific name to move forward. Also, user is able to exit at any time, and if user enters an invalid information, user will be asked to try again.

Below are examples of how to display looks like, first picture is text table and second picture is flask table.

```
Display in Text or Flask Table 1. Text 2.Flask: 1
           Lake Michigan
Max_Depth          281.0
Altitude           175.0
Volume          4860.000
Country    United States
Continent  North America
```

Lake Michigan!

Lake Name	Lake Michigan
Max_Depth	281.0
Altutude	175.0
Volume	4860.000
Country	United States
Continent	North America

Demo Video:

https://www.youtube.com/watch?v=uTaY_0BQaDQ