Assignment 1: HTTP Query Parameters, XML Parsing, Flask, requests
Release Date: January 28, 2020
First Deadline: February 7, 2020 (Friday, 11:59 pm)
Due Date: February 9, 2020 (Sunday, 11:59 pm)
Total points: 100
(This is an individual assignment)

Problem Description:
As part of Open data access, City of Austin provides data about swimming pools in the city. This information includes things such as the location of a pool, when it is closed, type of pool, etc. This data is available here:
https://data.austintexas.gov/Recreation-and-Culture/Pool-Map/jfqh-bqzu
In this assignment you will build a Python based Web application to provide a queryable interface to this data.

For our assignment purpose I have copied this data at the following location:
https://raw.githubusercontent.com/devdattakulkarni/elements-of-web-programming/master/data/austin-pool-timings.xml

You will work with the above XML version of the pool data.

Develop a Python Flask based web application that supports following query parameters: weekend, pool_type, weekday_closure.
Your program should show the names of the pools that satisfy the **combined** query parameters that are passed in. Here are some sample inputs and outputs:

1. http://localhost:5000?weekday_closure=Monday

```
Bartholomew
Clarksville
Mabel Davis
Dick Nichols
Balcones
Rosewood
Murchison
Kennemer
West Austin
Civitan
```

2. http://localhost:5000?weekday_closure=Monday&pool_type=Neighborhood

Balcones
Rosewood
Murchison
Kennemer
West Austin
Civitan

3. http://localhost:5000?weekend=Closed

Northwest
Reed
Walunt Creek
:
Ramsey

Input/Output assumptions and constraints:

1. Query parameters will be supplied in correct case as is the data. So you can assume that weekday parameter will be provided with ‘Monday’ ‘Tuesday’ etc. and not ‘monday’ (or any other variation).
2. Not every input will contain will contain all the query parameters.
3. Present the pool names in whatever order they appear in the original data. You do not have to sort the output.

Submission Instructions:

Name your submission folder as “assignment1”
Name your program pools.py
Structure of assignment1 directory should be as follows:

```
assignment1/
    requirements.txt
    pools.py
    README.txt
```

Feel free to create additional Python packages and modules if required.

Use Flask library/framework for web application development.
Use Python requests library for retrieving XML data.
Use Python XMLElementTree library for parsing XML data (don’t need to add this to requirements.txt as it is part of Python).

You will have to include names of these libraries in requirements.txt.

Include following information in README.txt
name: <your name>
eid: <your ut eid>
bitbucketid: <your bitbucket id>
comments: <Comments, if any>

Create a *private* repository on bitbucket (https://bitbucket.org/product/pricing?tab=host-in-the-cloud sign up using the free option if you don't have bitbucket account). Name the repository assignment1 and push/submit all the files that you create for your assignment to this repository.

Grant "read" access to Chris (username: chrisayoub)

We will use the latest commit ID for grading. You don't need to submit anything on canvas.

Deadlines:

First deadline (February 7, 11:59pm): First submission due. Your assignment does not need to be finished at this point. This deadline is to ensure that you complete all the required setup (Python, Flask) early and not postpone it till last minute. It will be graded for 3 points. If your first commit is on or before February 7 11:59 pm, you will receive points out of 100. If your first commit is after that then you will receive points out of 97.

Due date (February 9, 11:59pm): Deadline by which your assignment needs to be completed.

Late Penalty

Every day after the final deadline will incur a 5 point penalty.

Collaboration policy:

This is an individual assignment. You are allowed to discuss concepts and high-level implementation questions with each other. But you are not allowed to copy or share code with each other or students who might have taken this class before. Final submission should reflect your own code.