Assignment 3: REST API POST, GET, PUT, DELETE
Release Date: March 25, 2020
First Deadline: April 4, 2020 (Saturday, 11:59 pm)
Due Date: April 6, 2020 (Monday, 11:59 pm)
Total points: 100
(This is an individual assignment)

Problem Description:
In this assignment you will build a Python based REST API for Pool data. You do not need to access any external data source for this assignment.

Develop a Python Flask based REST API that supports following endpoints.

1. POST http://localhost:5000/pools

```
{
    "pool_name": "Barton Springs",
    "status": "Open",
    "phone": "512-974-6300",
    "pool_type": "Community"
  }
```

pool_name: Any string. Needs to be unique in the REST API (use this as natural key)
status: Valid values: Closed, Open, In Renovation
phone: 10 digits in following format: xxx-xxx-xxxx
pool_type: Valid values: Neighborhood, University, Community

Output:

- Status: 201 created
- Status:
    - 400 Bad Data
      Message: Pool with name <pool_name> already exists
    - 400 Bad Data
      Message: phone field not in valid format
    - 400 Bad Data
      Message: pool_type field has invalid value
    - 400 Bad Data
      Message: status field has invalid value

2. GET http://localhost:5000/pools/<pool_name>

Output:

- Status: 200 OK

{
"pool_name": "<pool_name>",
"status": "<status>",
"phone": "<phone-number>",
"pool_type": "<pool-type>"
}

- Status: 404 Not Found
  Message: Pool with name <pool_name> does not exist

3. PUT http://localhost:5000/pools/<pool_name>

{
"pool_name": "<pool_name>",
"status": "<status>",
"phone": "<phone-number>",
"pool_type": "<pool-type>"
}

Pass in update values as request body.

Output:

- Status: 200 OK
  {
  "pool_name": "<pool_name>",
  "status": "<status>",
  "phone": "<phone-number>",
  "pool_type": "<pool-type>"
  }

- Status: 404 Not Found
  Message: Pool with name <pool_name> does not exist

Output example:

- Status: 400 Bad Data
  Message: Update to pool_name field not allowed

Note: Updates are allowed only on status, phone, and pool_type fields. All the validations that are for POST requests should be applied to PUT requests as well.

4. DELETE http://localhost:5000/pools/<pool_name>

Output:

- Status: 200 OK
- Status: 404 Not Found
  Message: Pool with name <pool_name> does not exist

Notes:

1. You will need to use Database for this assignment. Spin up a MySQL server on your machine. Check Piazza post: https://piazza.com/class/k5k9xzk2v4a68x?cid=73
2. Check https://github.com/devdattakulkarni/elements-of-web-programming/tree/master/greetings for an example of connecting to a local database server from within a Python program.
3. For the table, feel free to use any name.
4. Check Canvas for screenshots of how to use curl to send data to REST API

Submission Instructions:
Name your submission folder as “assignment3”
Name your program pools.py
Structure of assignment3 directory should be as follows:
assignment3/
    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 json library for generating json

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 a 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 (April 4, 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 the last minute. It will be graded for 3 points. If your first commit is on or before April 4 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 (April 6, 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.