

Python Programming Test - LeanKloud

Part 1

[Flask](#) is a framework to build web applications using python, and [flask-restplus is a framework to build RESTful APIs in flask](#). [This example tutorial builds a Todo application](#) (to keep list of todos). Your task is as follows:

1. Start with the flask-restplus tutorial, and enhance it as described below. The tutorial code base **must** be the starting point for this exercise.
2. Implement the storage for tasks, and associated state, in a database such as sqlite or mysql. Preferably, don't use SQL Alchemy, instead, use the [native DB API](#) for that database.
3. Add 2 new fields to a task:
 - a. Due by, type date, of when this task should be finished
 - b. Status - Not started, In progress, and Finished
4. Add or modify web methods to change the status of tasks.
5. Implement the following additional end points:
 - a. "GET /due?due_date=yyyy-mm-dd" - this gets a list of tasks which are due to be finished on that specified date
 - b. "GET /overdue" - this gets all tasks which are past their due date, as of the date this query is run
 - c. "GET /finished" - this gets all tasks which are finished
6. **For extra credit**, implement authorization for this web app, with both read and write access. Users with write access can create and change the status of tasks. Users with read access can only view tasks, via the above APIs, but cannot create or modify them.

Part 2

The attached csv file contains the marks scored by students in a class, in different subjects. Each row has the marks scored by the student in the subjects of Maths, Biology, English, Physics, Chemistry, and Hindi. Write a python program, **without using pandas**, to efficiently

1. Find the topper in each subject.
2. Find the top 3 students in the class, based on their marks in all subjects.
3. The results should be printed on the console upon running the program with the csv file as the argument, and look as below:

Topper in Maths is (*name of student*)
Topper in Biology is (*name of student*)

...

Best students in the class are (*student first rank, student second rank, student third rank*)

Here, the actual student names should be output. Also state complexity of your algorithm in the Big-O asymptotic notation.

NOTE: pandas should **not** be used.

Rules

1. Each candidate must attempt this on their own, and should not discuss or ask for help from any other person. We will rely on an honour system and trust that you will not ask others for help. If we determine that this has happened, serious action will be taken.
2. You are free to use any other help, especially, use the web to search for ideas, solutions to sub-problems, etc.
3. Create a free account on github or bitbucket, and check your code in there.
4. When you've finished the assignment, create a screencast of using the system, performing various activities, and upload to some web location like google drive or even check it in.
5. Assignments are **due one week after receiving it**. To submit the assignment, send a link to the git repository, and a link to the screenshot, to vasanth@leancloud.com and vijay@leancloud.com
6. If there are any questions, they can be emailed to the same address.