Data at Your Command

Due Apr 10 by 11:59pm **Points** 100 **Submitting** a file upload
Available until Apr 28 at 11:59pm

This assignment was locked Apr 28 at 11:59pm.

Data at Your Command

For this lab, you are going to create a contact database using file I/O that uses your own CLI (command line interface). It will expose its own command prompt and accept commands until the user quits. You will store first name, last name, and phone number as a record in the database.

The following commands will work if typed in the command line:

add [fname] [lname] [phone] (add a new contact record)

list (list all records)

find [value] (find and show the first record that matches the search)

del [value] (delete the first record that matches the search)

quit (quit the CLI)

Lab Requirements:

- The database will store all data in a JSON formatted text file called *db.json.* The structure should be set up as a single list containing dictionaries. Each dictionary should hold the data for one person.
- If the user passes in too few or incorrect arguments on the CLI, give them an appropriate error message and then print the proper usage for the application on the screen.
- When finding or deleting a record, use the search string to match either the first name or last name exactly. Make this search case-insensitive by converting the search string and the names you are searching for to lower case using *lower().* Don't bother with RegEx on this assignment, we will look at that in the future.
- Keep errors from showing up from usage syntax or if a database file doesn't exist yet use a *try/except*.
- Code DRY (Don't Repeat Yourself). If you find you are writing the same code more than once, you should be making it into a function
- When adding or deleting a user, print a message that says the name of person and whether they were added or deleted. (e.g. Scott Wood was added.)
- When using the 'list' command to list all of the records, print a separator between records to keep it organized.
- When using the 'quit' command exit the application

Submission Requirements:

- Submit only the single python script, no other files should be needed or included.

Additional Resources:

- JSON data in Python: **https://www.w3schools.com/python/python_json.asp (https://www.w3schools.com/python/python_json.asp) (Links to an external site.) (https://www.w3schools.com/python/python_json.asp)**

Examples:

Testing should be done with python and look similar to this:

To add a person to the database

>>>add Scott Wood 801-302-2891

Scott Wood added.

To delete the first person found that has "Scott" as a first name or last name

>del Scott

Scott Wood deleted.

To find the first person that has "Wood" as a first or last name

>>>find Wood

Scott

Wood

801-302-2891

To list all the records in the database

>>>list

Scott

Wood

801-302-2891

Bob

Robertson

801-555-8888

Sally

Silverstone

801-555-7474

If the user does not use the right command or arguments, respond with something like this:

usage:

add [fname] [lname] [phone] (add a new contact record)

list (list all records)

find [value] (find and show the first record that matches the search)

del [value] (delete the first record that matches the search)

quit (quit the CLI)

Some Rubric (1)

Criteria	Ratings		Pts
Accept command: add [fname] [lname] [phone] (add a new record)	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Accept command: list (list all records)	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Accept command: find [search] (find and show the first record that matches the search)	**5 pts Full Marks**	**0 pts No Marks**	5 pts
Accept command: del [search] (delete the first record that matches the search)	**5 pts Full Marks**	**0 pts No Marks**	5 pts
The database will store all data in a JSON formatted text file called db.json. The structure should be set up as a single list containing several dictionaries. Each dictionary should hold the data for one person.	**20 pts Full Marks**	**0 pts No Marks**	20 pts
If the user passes in too few or incorrect arguments, print the proper usage for the application on the screen and repeat the prompt to receive their next command.	**10 pts Full Marks**	**0 pts No Marks**	10 pts
When finding or deleting a record, use the search string to match either the first name or last name exactly. Make this search case-insensitive by converting the search string and the names you are searching for to lower case using lower().	**10 pts Full Marks**	**0 pts No Marks**	10 pts
Keep errors from showing up from usage syntax or if a database file doesn't exist yet use a try/except.	**10 pts Full Marks**	**0 pts No Marks**	10 pts
Code DRY (Don't Repeat Yourself). If you find you are writing the same code more than once, show some effort by putting duplicated code into a function.	**10 pts Full Marks**	**0 pts No Marks**	10 pts

Criteria	Ratings		Pts
When adding or deleting a user, print a message that says the name of person and whether they were added or deleted. (e.g. John Smith was added.)	**10 pts Full Marks**	**0 pts No Marks**	10 pts
When using the 'list' command to list all of the records, print some sort of separator between records.	**10 pts Full Marks**	**0 pts No Marks**	10 pts
			Total Points: 100