

CSCI 2133 — Rapid Programming Techniques for Innovation

Assignment 4

Due: *by midnight, Friday, April 5, 2019*

Worth: 100 marks

Instructor: Gang Liu

Assignment Instructions:

For submission, firstly login into Gitlab using your CSID and password. Create a private repository called “Assignment-4” and share it only with the instructor and TA/marker of this course.

All files with answers must be added to Gitlab and committed as per instructions in the lab. You can make changes and commit the solution files as many times as you want but only the latest copy submitted before the deadline will be taken into consideration. **Please note that you will also have to submit the assignment on Brightspace by zipping all the necessary files.** Failure to do so will result in heavy loss of points.

To clarify the terminology: When we say submit, it means you add the file to Git using the command “git add”, and commit it to the repository using “git commit”.

Please note that the solution to each question needs to be placed under a separate folder for itself under “Assignment-4”. For this assignment, please make sure you include a README file which provides instructions to the marker on how to run each and every solution. Please note that insufficient details provided will cost you a lot of marks.

For questions involving the Django framework, it is mandatory to write a lot of backend validations. If you are using a virtual environment, please make sure you do not push it to Git. **Please keep track of the number of requests you make while querying the Twitter API as it might cost you if you cross the daily limit. Late submissions will be penalized by 10% with each passing day after the due date and submissions will not be accepted after April 7, 2019.**

Questions:

1. (10 marks) Complete the lab 10 exercise and push the sample project to your Git repository under a directory called a4q1.
2. (20 marks) Develop a web portal for a school where both the students and teachers can register, login and logout of their user accounts. Once the user is logged in,

display a simple profile page which displays the details they provided in the registration form. You can use the same registration and login page for both type of users if you want. Appropriate back-end validations are a MUST. For instance, if a user who already has an account tries to register, appropriate error messages need to be shown on the web page. Push the Django project to your Git repository under a directory called a4q2.

3. i) (10 marks) Write a python script in jupyter notebook to extract 50 tweets containing the word “Halifax” using your Twitter Developer account and push the results in CSV format to your Git repository along with the .ipynb notebook under a directory called a4q3. The CSV file should contain the tweet, the tweet id, the screen name and the time of creation of the tweet.

ii) (10 marks) Write a python script in jupyter notebook to read the CSV file you generated in the previous question and extract the hashtags present in the tweets, if any. Also, print the frequently occurring hashtag in the tweets along with its frequency. Place the solution under a folder called a4q4 and push it to your Git repository. It is mandatory that you use the third-party python package “pandas” for this question.
4. (20 marks) Develop a single-page web application using Django which uses the Twitter API and retrieves 10 tweets from a given user’s timeline on clicking a button called “Get Tweets”. Please also make sure you save the tweets in your sqlite database once you receive the API response. Register your model with the Django admin console as this will be used to check if you save the tweets to your database on every API call. Place the folder under a folder called a4q5 and push it to your Git repository.
5. (15 marks) Write a python program which satisfies the following:
 - i) An “Employee” class that stores employee details like salary, full name, age, and position during initialization.
 - ii) A class variable “raise_amount” which provides the percentage by which the salaries of all the employees under this class will be raised. (set it as 2 initially i.e. 200% raise)
 - iii) A method to retrieve the salary of any employee after applying the raise.
 - iv) A method to re-assign the class variable “raise_amount”.

Finally, create a list of 6 objects for the “Employee” class and write a function which takes the list as an argument and returns a sorted list of objects based on the salaries of the employees in the descending order.

6. (15 marks) README.txt is a MUST and it needs to provide detailed instructions on how to set up Django for every question, which URLs to visit (For instance, for a4q5, which URL should be used to visit the only web page present in the application?), the commands to make migrations to database, run the local development server, and also on starting a jupyter notebook.