CSE 337: Homework Assignment 4

Instructions

Please read the the following instructions carefully before coding. You may lose points if you fail to follow these instructions.

- Deadline for this assignment is **Dec 13, 11:59PM. No late submissions accepted**.
- You will need create a sub-folder for each question, since your answers to first two questions will have a complicated directory structure.

  Note: Please rename your folder in the right way before zipping! For example, if your name is Ying Lu and your student ID is 111345678, put all submission files in a folder named Ying_Lu_1112345678, then compress that folder to Ying_Lu_1112345678.zip.

1 Delete Functionality for Wiki[30 points]

Your task in this question is to add a remove page to the Flask example provided in Lecture 22 such that the users are able to remove an article `<postid>` by navigating to `/remove/<postid>`. You will need to create an additional method inside the `routes.py`, create a new template named `remove.html` for the page that displays acknowledment for the delete operation, and you will need to implement the functionality to delete an entry in the database. Also, you will need to add a link to the corresponding edit page in `/posts/<postid>`(*e.g.,* there should be a link to `/delete/3` in `/posts/3`)

2 Text Analyzer[40 points]

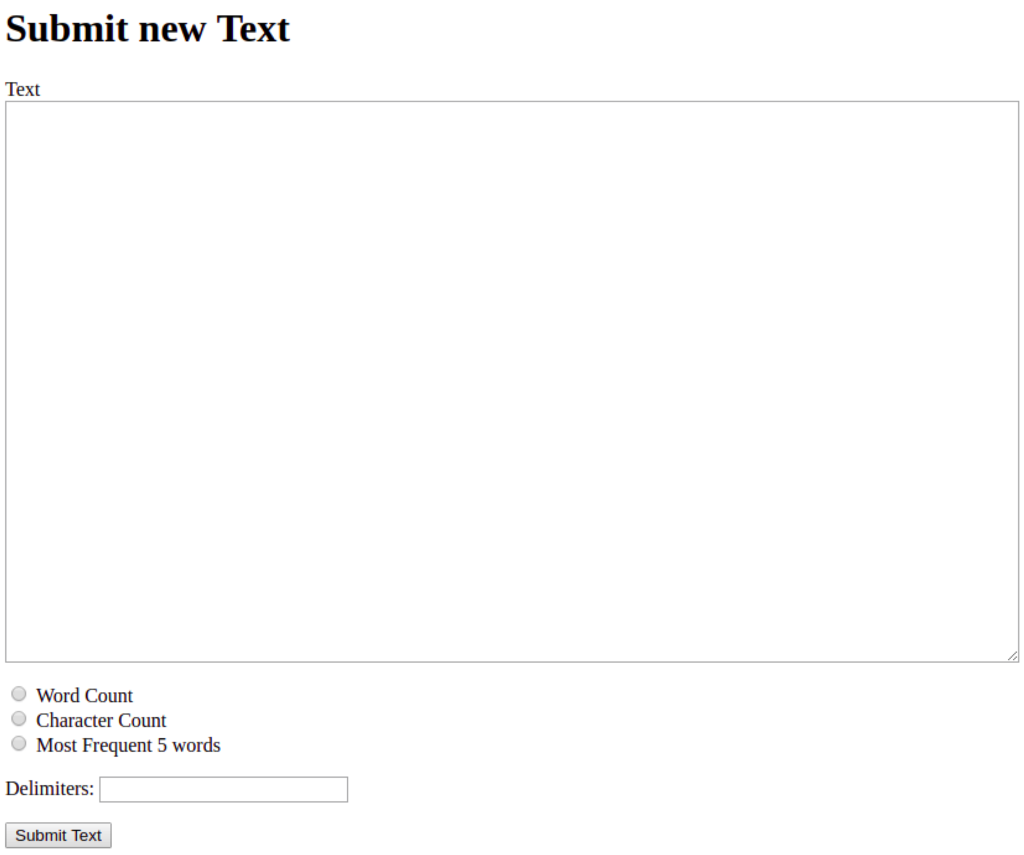
Write a Flask application that can be used to analyze a given text. The user should be able to enter a text, and pick an operation to perform on the text. Once the user clicks submit, your application should redirect to `/result/<operationname>` (*e.g.,* `/result/wordcount`) and there it present the results of the chosen operation. Delimiters field is used to denote the characters to be used as delimiters(In addition to spaces). You can ignore the delimiters field in character count use-case. For example, if the user enters ".,;" to the delimiters field, delimiter characters will be space( ), dot(.), comma(,) and semicolon(;). Your application should display the result to the user, along with a link back to the index page. Your application should check for invalid inputs(*i.e.,* check for no input etc.) and redirect to another page(*e.g.,* `/result/error/`) where an error message will be displayed to the user, with a link back to the index page. Your application should also feature custom error pages for 404, 403 and 500 status codes(Contents of these pages are up to you). Your design elements should be decoupled from the Python code, and the overall design needs to be modular. You should concentrate on avoiding repeating code as much as possible.

3 GUI Programming [30 points + 10 bonus points]

An Improved Phonebook[30 points]

The phonebook example we have seen in the class can be further improved to be more user-friendly. For example, it can store more information related to the users, and there could be a separate window for each and every functionality. Your task in this question is to implement a multi-window phonebook with additonal functionality. Figure 2 provides the windows your application should feature. Your application does not have to look exactly the same as the examples presented below, however, it should implement the same functionality. You should check for empty inputs in all screens, and display message boxes to the user in the even of invalid input and successful record creation/deletion.

Bonus [10 points]: Add another functionality(and the corresponding button/window in the application) to list all existing records to the user. Your window size should change according to the number of records.



Submit new Text

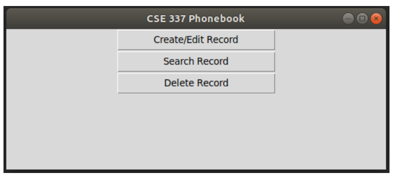
Text

Word Count
Character Count
Most Frequent 5 words

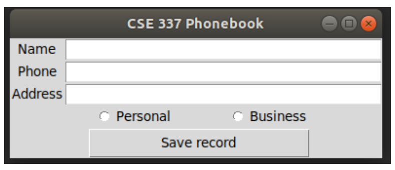
Delimiters:

Submit Text

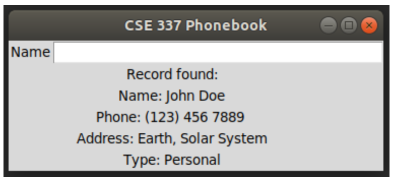
Figure 1: Sample Input Form for Question 2.



(a) Main Screen



(b) New Record Screen

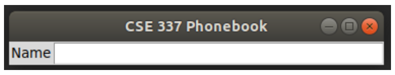


CSE 337 Phonebook

Name

Record found:
Name: John Doe
Phone: (123) 456 7889
Address: Earth, Solar System
Type: Personal

(c) Search Screen



(d) Delete Screen

Figure 2: Pictures of animals