CSE 4256

Due Sunday Apr 10 11:59 PM EST

Dr. Alan Weide

Spring Term 2022

Homework 10

Code Quality

Wile you do not need to correctly answer *every* question in order to earn a grade of "satisfactory" on assignments in this course, please at least *attempt* every question, keeping in mind that some questions carry more "weight" than others; a 60% or so is still required to earn a grade of satisfactory on the homework.[1] You may be surprised how quickly things click once you start working on them!

Assignment

This assignment should be completed as a collection of several Python source (`.py`) files, uploaded to Carmen as a **`.zip`** file. You may only use modules and packages from the standard library that we have covered in class, or that you have written yourself in prior homework assignments.

(1) Create a file called `runserver.py` that, when run, starts an HTTP server with CGI enabled on port 8000 from the current directory.

(2) In the same directory as `runserver.py`, create a new directory called `cgi-bin`. In that directory, place the file `cardutils.py` that was uploaded along with the assignment to Carmen.

(3) If you did not complete either of the two functions `play_game` or `review_game` in Homework 9, complete one of them now before proceeding with either step 4 or 5 as appropriate.

(4) If you completed the body of `play_game` in Homework 9, do the following.

   (a) Create a file called `playgame.py` in the `cgi-bin` directory. This script should respond to HTTP POST requests from a client to your server with data in CSV format following the requirements for the `play_game` function from last week's homework, except that it should *only* print the cards that were played, and not the scores or a game summary. The body of the request should have one field, "`nplayers`", containing the

[1] It should be obvious which are the heavily-weighted questions, but if you are unsure please ask me.

number of players to simulate, and it should use a standard card deck, as generated by the function `std_card_deck`.[2]

(b) Create a file called `main.py` in the same directory as `runserver.py`. From `main.py`, use `urllib` to submit a POST request to `http://localhost:8000/cgi-bin/playgame.py` with appropriate `data`, and print the response to the console along with the HTTP status code and reason phrase. See my Demos repl for a sample client script.

(c) Modify `main.py` so that instead of printing the results to the console, it stores the results locally in a CSV file.

(5) If you completed the body of `review_game` in Homework 9, do the following.

(a) Create a file called `reviewgame.py` in the `cgi-bin` directory. This script should respond to HTTP POST requests from a client to your server with an HTML webpage displaying the results of the game that was uploaded as a CSV file accompanying the request with key "`gamefile`".

Note: the file in the CGI `FieldStorage` instance is a sequence of *bytes*, not strings! You'll need to convert the file to an iterable of strings to use with a `csv` reader. Use the `decode(encoding='utf-8')` instance method on bytes objects to do so.

(b) To handle file uploads, create a file called `index.html` in the same directory as `runserver.py`, and in it place a form with a file upload field (see an example of such a form in my Demos repl in the file `webapp/index.html`). The form should also have a field for the number of players, named "`nplayers`". The HTML file should describe what the application does (*e.g.*, "Upload a CSV file containing the output from a card game and this web app will tell you the results of the game!"). When the user submits the form, it should send a POST request to `cgi-bin/reviewgame.py`.

(6) Start your server either by running `runserver.py` or by issuing the command `python3 -m http.server --cgi` from the terminal in the appropriate directory. Use

[2]Hint: the `csv` module can print CSV-like text to `sys.stdout` by importing `sys` and using `sys.stdout` as the "file name" when instantiating a CSV writer. The response from a CGI script is whatever that script printed to `stdout`.

this server to test your CGI script by submitting requests either from `main.py` or from the HTML webpage.

Challenge Activities

Some homeworks (such as this one) will have additional challenge activities. These activities **do not contribute to your grade**, but they are problems that I find interesting or challenging.

(7) Complete the other of the two CGI scripts (either `playgame.py` or `reviewgame.py`).

(8) Modify your `index.html` file to let the user decide whether to simulate a game or to review a game.

Submission

To submit this assignment, upload a **`.zip`** file named "lastnamefirstname.zip" (obviously replacing lastname with your own last name and firstname with your own first name) containing all of your Python and HTML files to the "Homework 10" assignment on Carmen. As always, be sure to note all group members who contributed to the assignment and what those contributions were.