CSC 210 Homework 4 Calculator and Lotto Endpoints

Assignment Description

For this assignment you will be required to write your first server side application using Python Flask. You must create three endpoints on your server with the routes /add /multiply and /lotto. Each will be required to receive and parse input from the URL query string. These endpoints must return the appropriate calculation or results, however there is no requirement to return a properly formed HTML page. The purpose of this assignment is to get you started with Flask and demonstrate that you can receive and process user input on your server. I encourage you to simply return the result as a string rather than attempting to construct a full HTML response. We will be covering generating proper HTML responses from Flask soon. Similarly for this assignment you may assume the presence of the necessary variables in the query string and that they will be integers.

Add

- Your add endpoint must be available at the route /add on your server
- It must receive two integers, x and y, as query string input and return the sum of those integers as a string.
- The following is an example URL and the desired resulting response (you should use the default server location of localhost port 5000 for your app)
  
  http://localhost:5000/add?x=40&y=2 ==> 42

Multiply

- Your multiply endpoint must use the route /multiply on your server
- It must receive two integers, x and y, as query string input and return the product of those integers as a string.
- The following is an example URL and the desired resulting response (you should use the default server location of localhost port 5000 for your app)
  
  http://localhost:5000/multiply?x=40&y=2 ==> 80

Lotto

- Your lotto endpoint will use the route /lotto
- It must accept as input two integers, n and k as input from the query string
- It should return as a string a space separated list of k integers which have been sampled from the interval [1,n) without replacement.
• Please note that if \( k \geq n \) this is impossible, and your server should instead return a warning of some kind with an appropriate HTTP response code. In this case I would recommend 422 Unprocessable Entity.
• The following is an example, however due to the random nature of the endpoint your tests will almost certainly not be identical.

http://localhost:5000/lotto?k=5&n=100 \Rightarrow 7 \ 46 \ 65 \ 57 \ 26

Recommended Reading

• Chapters 1-2 of your textbook
• Flask Mega-Tutorial
• Query Strings
• Interval Notation
• Random Package
• Response Codes