BSCCS2003: Week-3 Lab Assignment

In this assignment, you have to use Python to take user inputs as command line arguments to generate specified output file(s). We list below the instructions to be followed in preparing and submitting the solution.

Instructions:

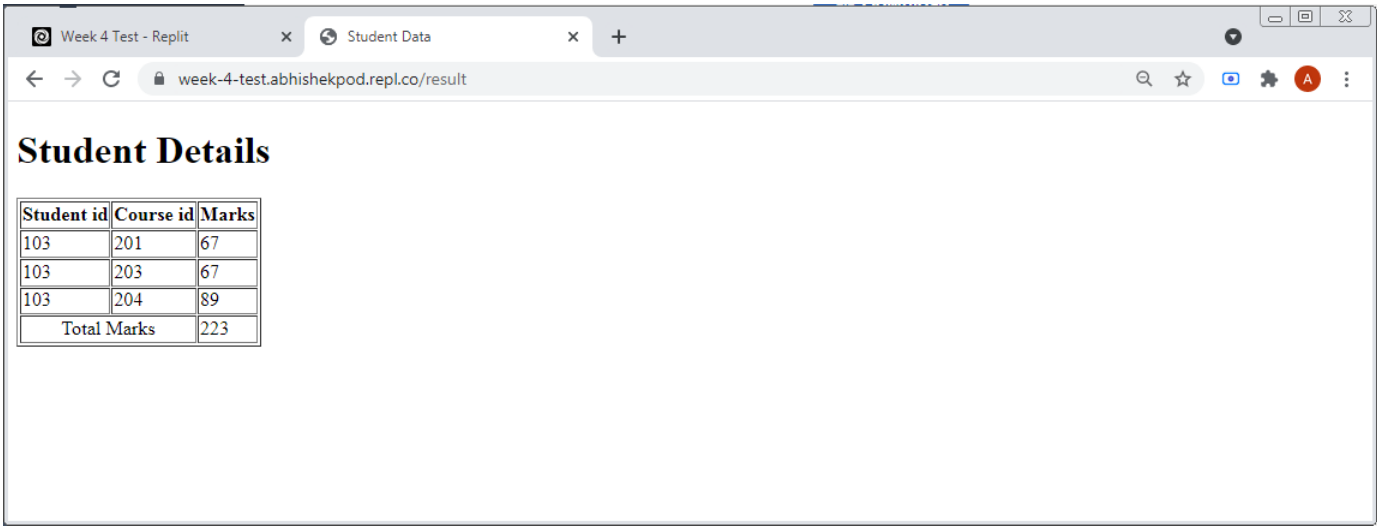
- The solution for this assignment must be submitted as a single Python file. The name of this Python file should be app.py.
- Submit a single .zip file containing app.py (only), the name of which should be "<roll_number>.zip". E.g.: *21f1000000.zip* (Note that the file app.py must reside in the root directory of your submission zip file)
- The Python code must generate an output HTML document named "output.html" and an image (as is required by the problem statement) in the current working directory.
- Allowed Python packages: pyhtml, jinja2, matplotlib or any standard Python3 package.
- The output page should be HTML5 compliant, e.g., the file should begin with the declaration <!DOCTYPE html>.
- Use of any HTML code generator may lead to rejection of your submission.
- You must use relative paths only. To access the csv file in python program, you must use the name "data.csv" only and should not use any directory path.
  What should not be used:
  C:\Week3\data.csv
  Week3\data.csv
- You will be given as input, a CSV file named *data.csv* that contains the marks of some students in few courses. The fields of the CSV file are given below. You have to write a Python program assuming that the CSV file is residing in the current working directory and the name of the CSV file is data.csv.

Student ID	**Course ID**	**Marks**

Problem Statement:

- The Python program must take as input two parameters as command line arguments.
- The first parameter is either '-s' or '-c'. "-s" specifies that the second parameter is a student ID and "-c" specifies that the second parameter is a course ID.
- The second parameter will be the corresponding student ID or course ID.
- Example input: `python app.py -c 201`, where 201 is a course ID.
- If the first parameter is '-s', then the second parameter should be a student ID. From the input CSV file, the program must extract marks, for each course, of the student whose ID is given as the second parameter. It must create an HTML page that displays the output in a tabular form (the order of columns must remain the same) as shown in Figure 1. The table should be titled as "Student Details" and should have column headers: "Student ID", "Course ID" and "Marks". The table should also display the total marks of that student in the last row as shown.

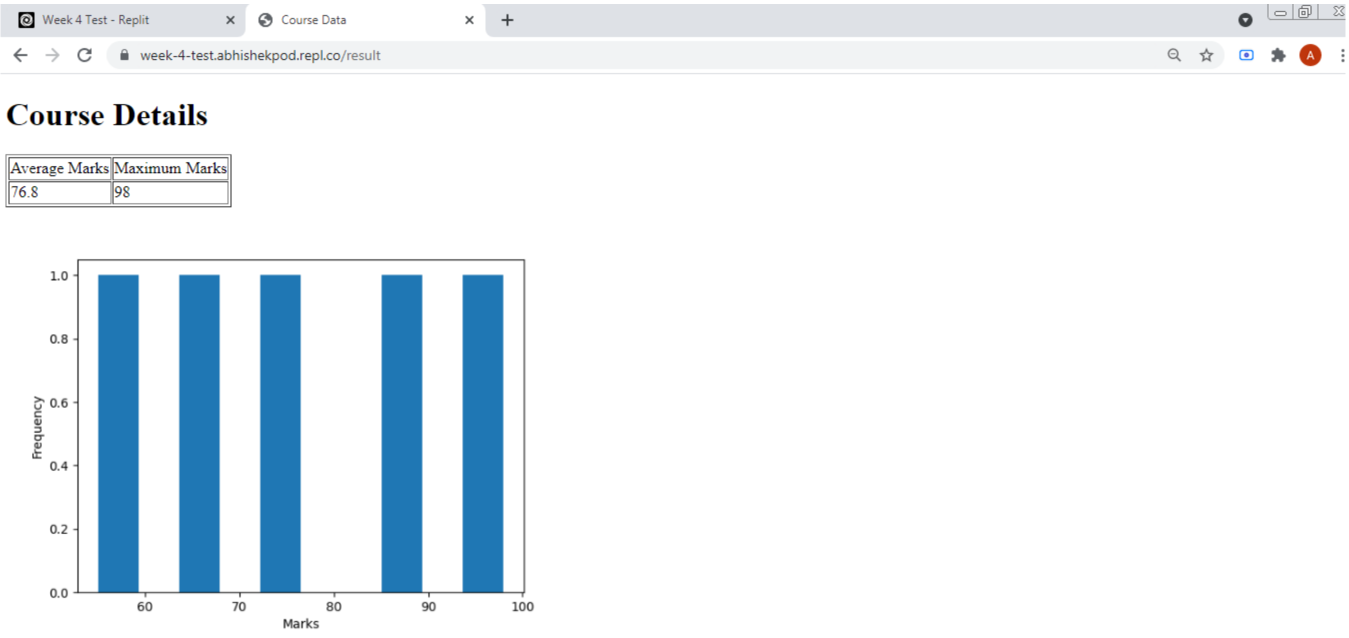
- If the first parameter is '-c', then the second parameter should be a course ID. The Python program must find the highest and the average marks for that course and display it on an HTML page as shown in Figure 2. The title of the table must be "Course Details" and the column headers must be "Average Marks" and "Maximum Marks" (the order of columns must remain the same) as shown. The Python code must also display the histogram of marks for the given course ID as shown in the figure.
- The program should display an error message if there is a deviation from the expected input. For instance, if "-s" is followed by a course ID or any invalid student ID, then a message as shown in Figure 3 must be displayed on the HTML page.



Student Details

Student id	Course id	Marks
103	201	67
103	203	67
103	204	89
Total Marks		223

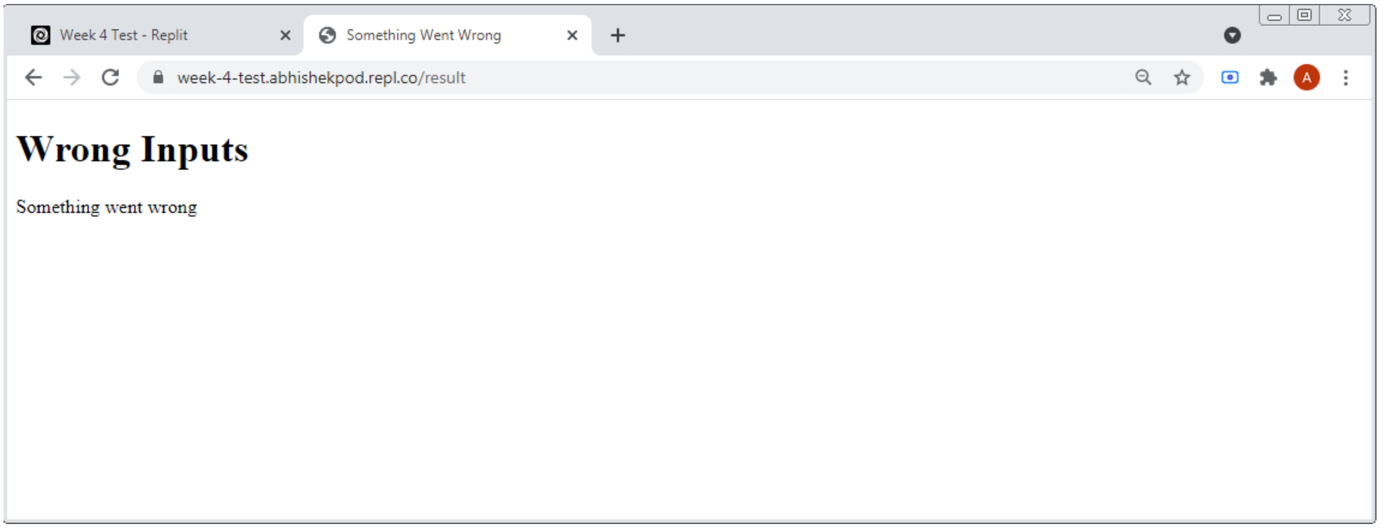
Figure 1: Student Details



Course Details

Average Marks	Maximum Marks
76.8	98

Figure 2: Course Details



Wrong Inputs

Something went wrong

Figure 3: Warning message