

BSCCS2003: Week-4 Lab Assignment

In this assignment, you have to create a web application, using flask. We list below the instructions to be followed in preparing and submitting the solution.

General instructions:

- Submit a single .zip file containing all your submission files and folders, the name of which should be “<roll_number>.zip”. E.g.: 21f1000000.zip
- The folder structure inside the zip file should be as follows:
 - The Python program must be written inside a file named “app.py”. This file must reside inside the root directory of submission.
 - All the HTML files should be kept inside a folder named “templates” and the images or CSS files (if any) should be kept in “static” folder and this “static” folder must reside inside the root directory of submission.
 - You are not required to submit the data.csv file.
- You should not keep any code inside the scope of the condition “if name == ‘main’” except run() call.
- Allowed Python packages: jinja2, matplotlib, flask, or any standard Python3 package.
- The web pages should be HTML5 compliant, e.g., the file should begin with the declaration <!DOCTYPE html>.
- 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 the application assuming that the CSV file is residing in the current working directory (same as app.py) and the name of the CSV file is data.csv.

Student ID	Course ID	Marks

Problem Statement:

- Using a standard flask template, you have to create an application that displays an index page as the first webpage. The index page should have an HTML form with two radio buttons: one with the label “Student ID” and the other with the label “Course ID”. It must also have an input text field to enter the ID and a button with label “Submit”, as shown in Figure 1. The HTML form should be same as mentioned below:

```
<form method="POST" action="/" id = "data-form">
    <input type="radio" name="ID" value="student_id" />
    <label>Student ID</label>
    <input type="radio" name="ID" value="course_id" />
    <label>Course ID</label>
```

```

        <input type="text" name="id_value" />
        <input type="submit" value="Submit" />
    </form>

```

- Suppose the radio button for student ID is selected and the user enters a student ID in the input text field as shown in Figure 2. Then, on clicking the “Submit” button, the application should send a POST request to the application’s URI = “/” and must extract marks of the specific student, for each course, from the input CSV file. It must display a table titled “Student Details”, having columns with headers: “Student ID”, “Course ID” and “Marks”. The table should also display the total marks of that student in the last row. It must also display a link to navigate back to the HTML form, as shown in the Figure 3.

The HTML for the table which displays the student details should look like the following. Its id should be “student-details-table”.

```

<table border = "2" id = "student-details-table">
    <tr>
        <th>Student Id</th>
        <th>Course Id</th>
        <th>Marks</th>
    </tr>
    <tr>
        <td>student_id_data</td>
        <td>course_id_data</td>
        <td>marks_data</td>
    </tr>
    ...
    ...
    ...
    <tr>
        <td>student_id_data</td>
        <td>course_id_data</td>
        <td>marks_data</td>
    </tr>
    <tr>
        <td colspan = "2" style="text-align:center"> Total Marks </td>
        <td>total_marks_data</td>
    </tr>
</table>

```

Note: student id data, course id data, marks data, total marks data should be populated according to the data extracted from the CSV file.

- Suppose the radio button for course ID is selected and the user enters a course ID in the input text field as shown in Figure 4. Then, on clicking the “Submit” button, the application should send a POST request to the application’s URI = “/” and must find the highest and the average marks for that course and display it in a table. The title of the table must be “Course Details” and the column headers must be “Average Marks” and “Maximum Marks” as shown in Figure 5. The page must also display the histogram of marks for the given course ID as shown. It must also display a link to navigate back to the HTML form. The HTML for the table which displays the course details should look like the following. Its id should be “course-details-table”.

```

<table border = "2" id = "course-details-table">
    <tr>
        <th>Average Marks</th>
        <th>Maximum Marks</th>
    </tr>
    <tr>

```

<td>average_marks</td>

Page 2

<td>maximum_marks</td>
</tr>
</table>

Note: average marks, maximum marks should be populated accordingly.

- The application should display an error message if there is a deviation from the expected input. For instance, suppose the radio button for student ID is selected, and the user enters a course ID or any other invalid text in the input text field or leaves it empty, then a message as shown in Figure 6 must be displayed. It must also display a link to navigate back to the HTML form.

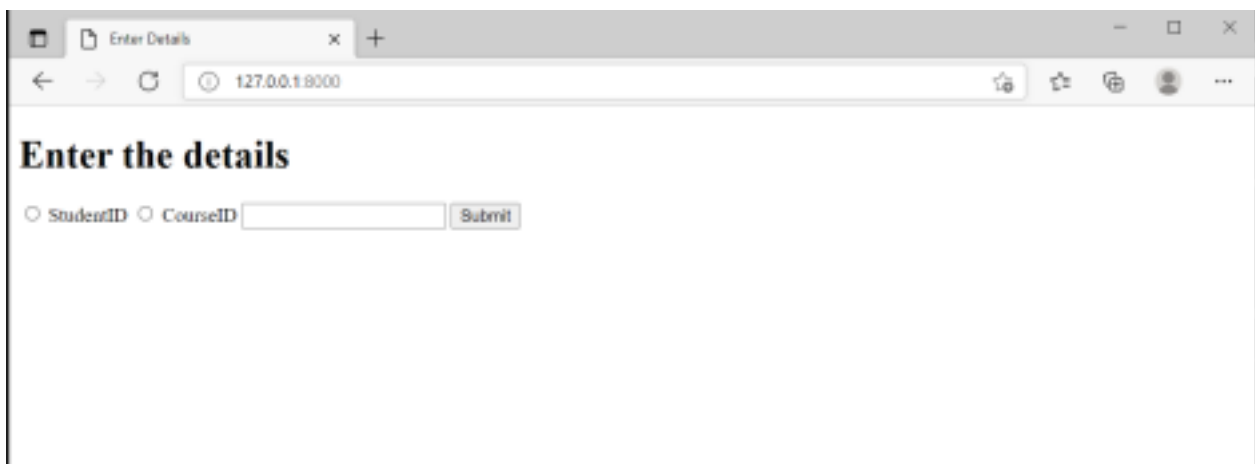
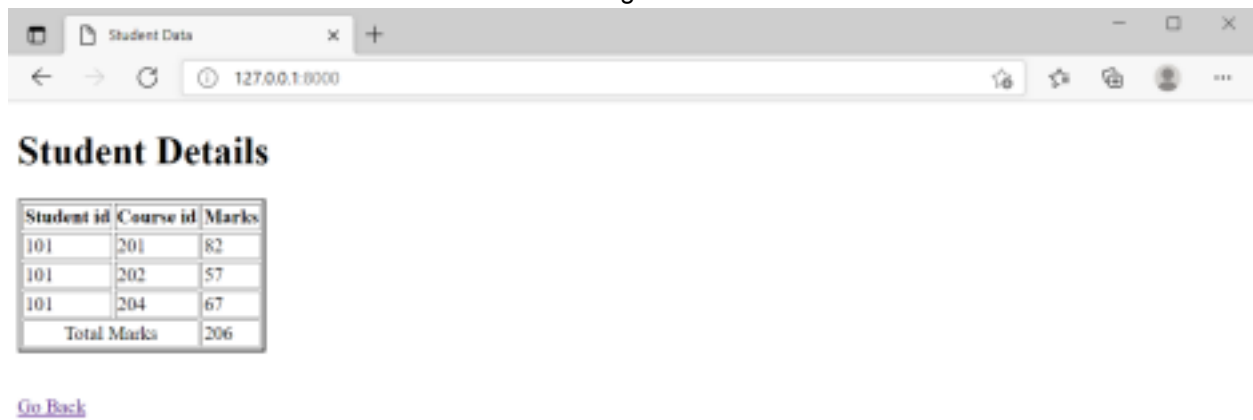


Figure 1: Form



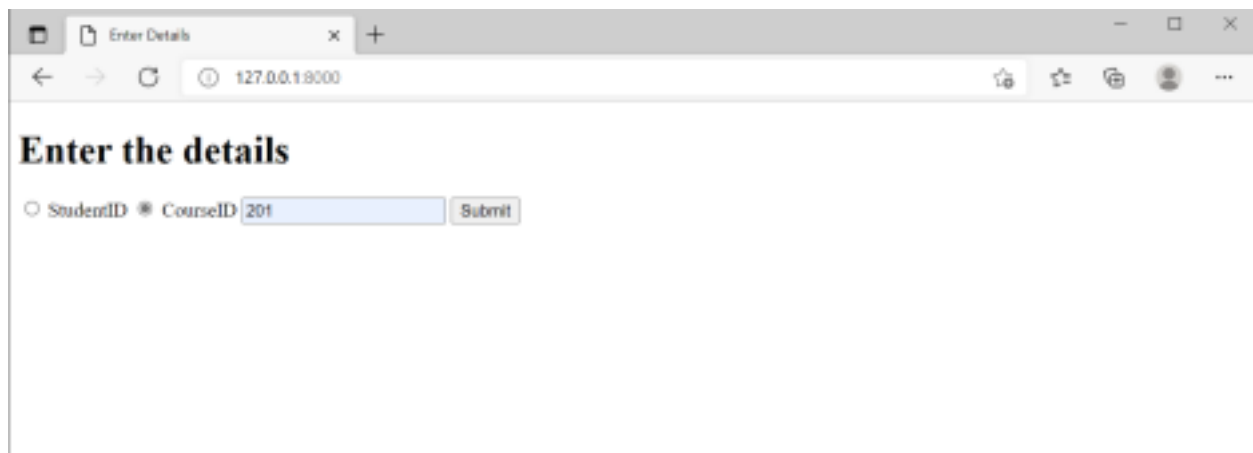
Figure 2: Student input in form



Student id	Course id	Marks
101	201	82
101	202	57
101	204	67
Total Marks		206

[Go Back](#)

Figure 3: Student Details



Enter the details

☐ StudentID * CourseID

Figure 4: Course input in form

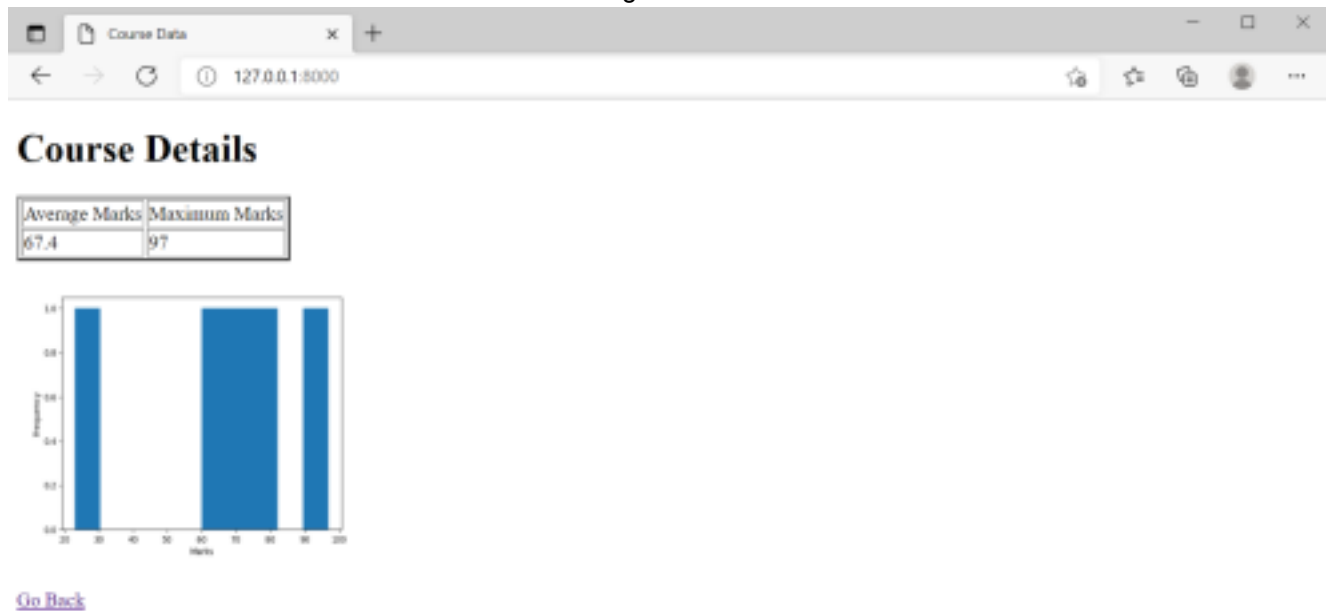


Figure 5: Course Details



Figure 6: Warning message

