

CSCB20, Assignment3 (HTML, CSS, JavaScript, SQLite and Flask)  
Due date 5th April @ 11:59pm. YOU MUST WORK TEAMS OF 3  
FOR THIS ASSIGNMENT. If you choose not to do this, then we will  
apply some penalty on your assignment mark.

## 1 Introduction

In this assignment, you will continue working on your CSCB63 course website from assignment2. However, in this assignment3, you must also use JavaScript, SQLite, and Flask. This assignment is challenging, and you are asked to get started early on it. You can use JQuery for this assignment. However, you cannot use any 3rd party CSS frameworks. You can work in teams of 3 for this assignment. The website must be usable, responsive, and visually appealing. There are some minimum requirements, but for the most part, there is a lot of creativity! This assignment gets a complete closure on everything we have been learning in this course, starting from databases and concluding with web technologies.

## 2 What is expected in Assignment3?

Now that you have your CSCB63 course website developed for assignment2, you are asked to add the following new features to your website:

1. You must create a login page for CSCB63. i.e., users can only access all contents of CSCB63 (i.e., lecture material, lab material, course instructor, TA information, etc., etc.) after logging in. You may want to redesign your homepage, i.e., the first page of your website, such that it allows someone without a username and password to create an account first. I will leave it up to you what information you like to collect when the user does not have an account. Again, you do not want to worry about the security of this website. However, at the very least, there should be an option on the user account creation whether the account is being created for a **student**, or **instructor**. All two types of users, i.e., **student**, or **instructor** must have access to all the contents of your website. If anyone tries logging in with an incorrect username and password, no content of your website must be accessible. Instead, you must give them an appropriate message of incorrect username/password and ask them to try again. You are free to design this homepage in any way you like. Feel free to use the University of Toronto Scarborough logo or related picture on your website. Please keep it simple, clear, and appealing.

### 2. Student Users

- When a user of type **student** logs into your website, you must have a special link for this student shown that displays his/her current marks from all assignments, midterm exam, labs, and the final exam. You may want to debate first with your team member, what is the best way to show this information to the student? Table form (created using div elements?), or in some other form? Let us assume that a student named 'Bill' has logged in. The idea here is that there is some link such as 'Welcome Bill, click here to see your current grades.' When Bill clicks on the link, you pull whatever information from the SQLite database you have about Bill, i.e., his

grades, and display it back to him. It is perfectly fine to hardcode some dummy data into your database for some students with certain marks. Make sure that Bill only sees his information and no other information about other students. This link must also be visible to users of type **students**.

- The student must be able to view their mark and have the option to submit a remarking request. To send a remark request, you may want to have a small button such as ‘submit remark request’ next to each mark, along with a small textbox that allows the student to explain why they are asking for a remarking request.
- Create a special page somewhere on your website that allows students to submit anonymous feedback back to the instructor. The anonymous feedback form can be designed in HTML/CSS/JavaScript or whatever you feel is appropriate. However, it should contain at least the following kind of questions:
  - (a) What do you like about the instructor teaching?
  - (b) What do you recommend the instructor to do to improve their teaching?
  - (c) What do you like about the labs?
  - (d) What do you recommend the lab instructors to do to improve their lab teaching?

You can add more questions as you see fit. However, keep in mind that this link must only be visible to users of type **student**. No one else must have access to this link. There should also be an option for the student to select the different instructors present on the website. You should enumerate all the instructor users and display them to the students to select the instructor for whom the feedback is meant for. You do not want the feedback going to the wrong instructor!

### 3. Instructor Users

- When a user of type **instructor** logs into your website, you must have a special link for this instructor shown that displays all his/her students’ grades. You may want to debate first with your team member, what is the best way to show this information to the instructor? Table form (created using div elements?), or in some other form? The idea here is that there is some link such as ‘Welcome instructor Anna, click here to see all the grades of your class.’ When I click on the above link, you pull whatever information from the SQLite database you have about the entire class. It is perfectly fine to hardcode some dummy data into your database for some students with certain marks. Make sure that this information is only visible to users of type **instructor**. Ideally, you will like the instructor to see all students’ marks of his/her course only. However, to keep it simple here, it is fine if you instead show all the students’ marks for any user of type instructor. In other words, any user of type instructor can log in and see the marks of all students. Yes, this is silly. However, it does keep things simple.
- When a user of type **instructor** logs into your website, you must also provide this user with a special link so that the instructor can see all the anonymous feedback. It is OK to hardcode some data into your database. Please make sure the logged-in instructor only sees the feedback that is meant for them. You do not want some feedback shown to the wrong instructor!
- The instructor must have a special link that allows them to see all remarking requests. Ideally, you will like the instructor to see all students’ remarks of his/her course only. However, to keep it simple here, it is fine if you instead show all the remarks. In other words, any user of type instructor can log in and see the marks of all students. Yes, this is silly. However, it does keep things simple.
- The **instructor** must have the option to enter marks for students. Again, you will want to think about what is the best design to do this. There are multiple ways to do this. You may want to either have a simple form-like structure that allows the instructor to enter the mark and when the ‘submit’ button is clicked, the marks are stored in the database.

### 3 Requirements

- Before starting to code in HTML and CSS, you are required to first think about what kind of tables do you require in your SQLite database. You will also need to think about the schema, primary key, number of columns, etc. This is everything that we have looked at in the first half of the semester.
- Once you have your schema in place, you will first want to test this out. By testing, I mean writing simple SQL queries that allow you to create a user, log in a user, view student marks, enter information about anonymous feedback, etc.
- Remember the first two points on the top, do not require any Flask, HTML or CSS, or JavaScript. You can test the first two points simply using material from the first few weeks of this semester. Once you have tested your SQL queries and you know that they work fine, at this point, you can start thinking about integrating your SQL queries into your Flask code. Do not move to this step unless and until you have the first two points in this section completed and tested.
- You will want to use the in lecture code from the last two weeks of March ( I am working on this to have the code and have my videos). In the lecture, we talked/will talk about how to fetch data from SQLite database using Flask, we also talked about how to log in a user using Flask and SQLite. You should be able to take this code and customize it to meet the requirements of this assignment.
- Remember this is a fully functional website. We should be able to log in as an instructor on your website and enter some student dummy marks. And then log out and log in using that student and should be able to see the mark of that student. Etc. etc.

### 4 Deliverable

TBA