# CSCB20 A3

## Overview

The CSCB20 website and dashboard is developed by Rikin Katyal and Sajid Rahman. This website allows instructors to send class-wide announcements, view and update student marks, receive anonymous feedback about their class and teaching, administer their class, and view & respond to remark requests for assessments.

The website is hosted on Mathlab here: <https://mathlab.utsc.utoronto.ca/cscb20/katyalri/>. You should run this locally on your machine as PHP is causing problems on Mathlab that do not occur on your local machine. Everything is fully functional locally using MAMP.

## Challenges

* One challenge was deciding what info on a single page to hide or display, depending on who is logged in. We had looked at how much common code and elements we had on a single page, and if it was not much overlap, we made a new page for the students and instructors. The student would be directed to the student marks page for example, and the instructor would be directed to a different page.
* Another challenge was building on top of our old website. We needed to make the new website easily accessible from the old website but also keep it separate as it contained account-related content. We also wanted to make sure only people logged in could access the old site. To tackle this we used a *session* to keep track of who’s logged in, and redirected them to the login page if they weren’t. We also created a dashboard page that housed all the new additions linking to the db, and made this dashboard accessible via the old website and vice versa. The dashboard is the main control center to navigate between pages.
* Another challenge we faced was debugging our code as PHP is not really well-suited for that. When something went wrong, it would just not load the page instead of telling us what line the problem was caused. In order to fix the error, we would have to try to *echo* every line until we came across the line causing problems and debug that in depth. There is no console where we can print the value of variables, so they must be echoed which is time consuming and a hassle.

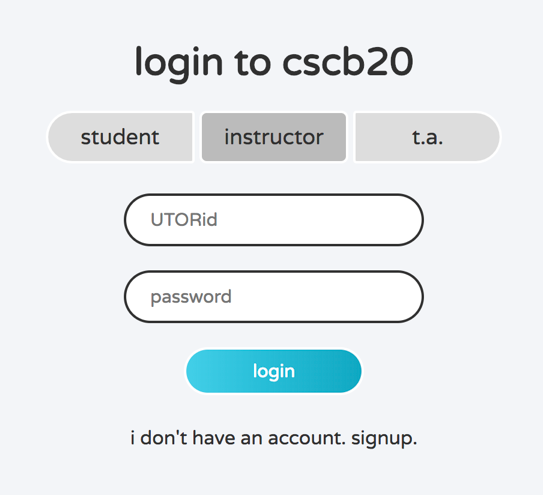
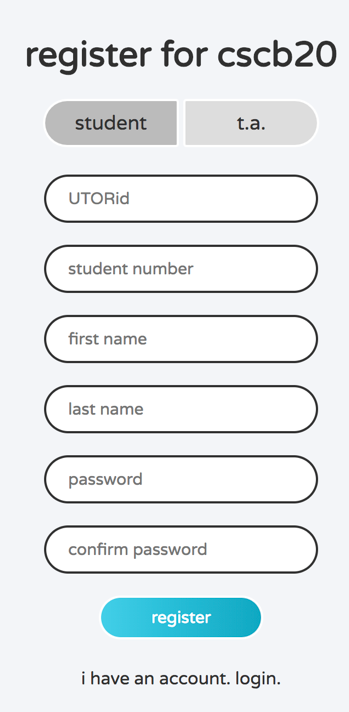
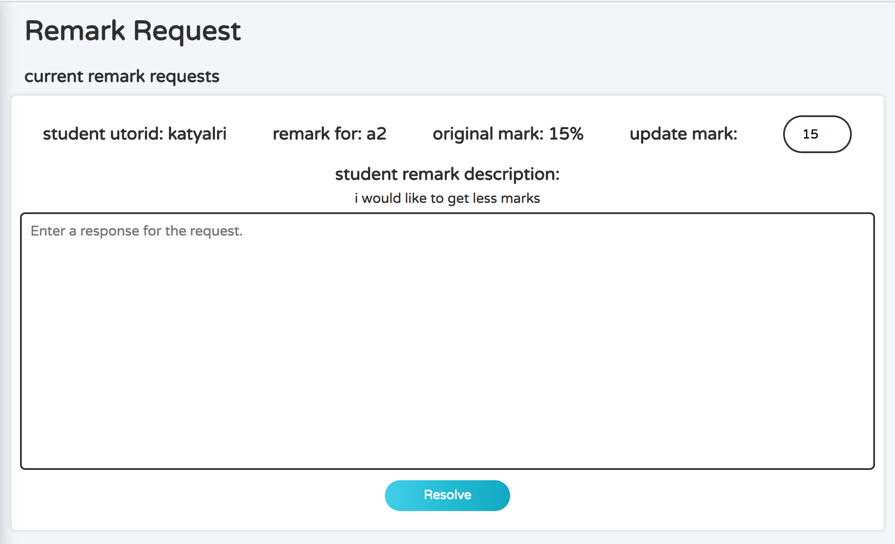
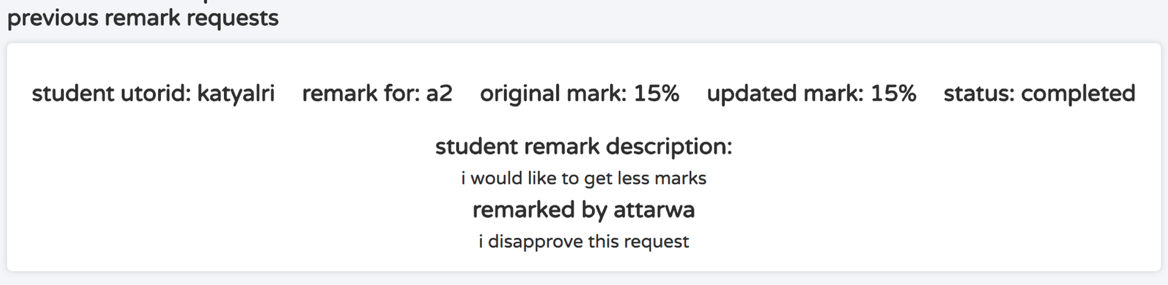
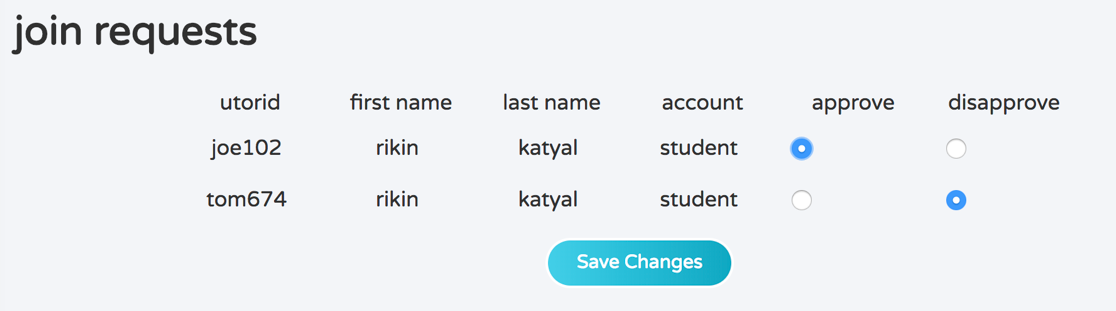
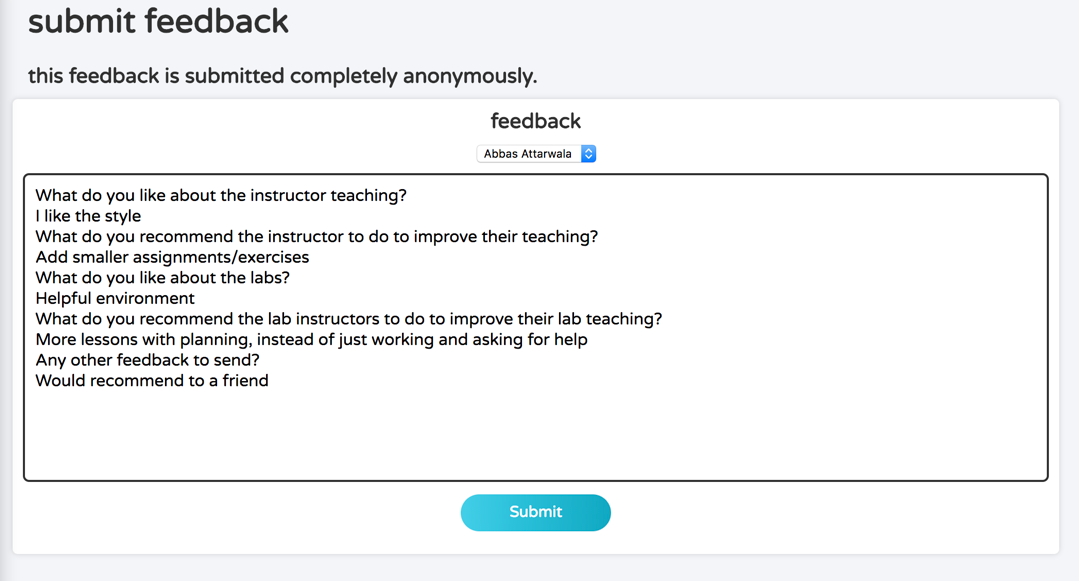
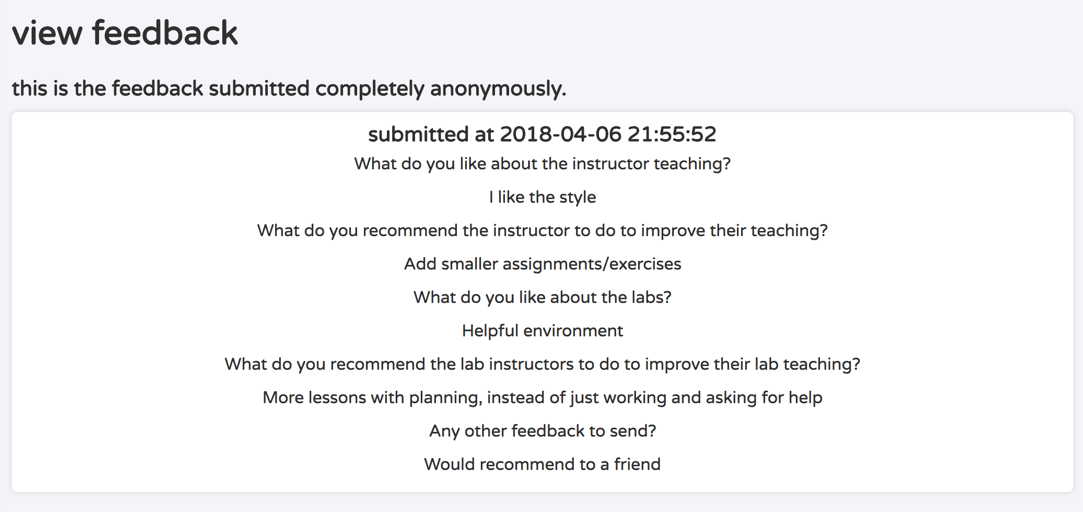
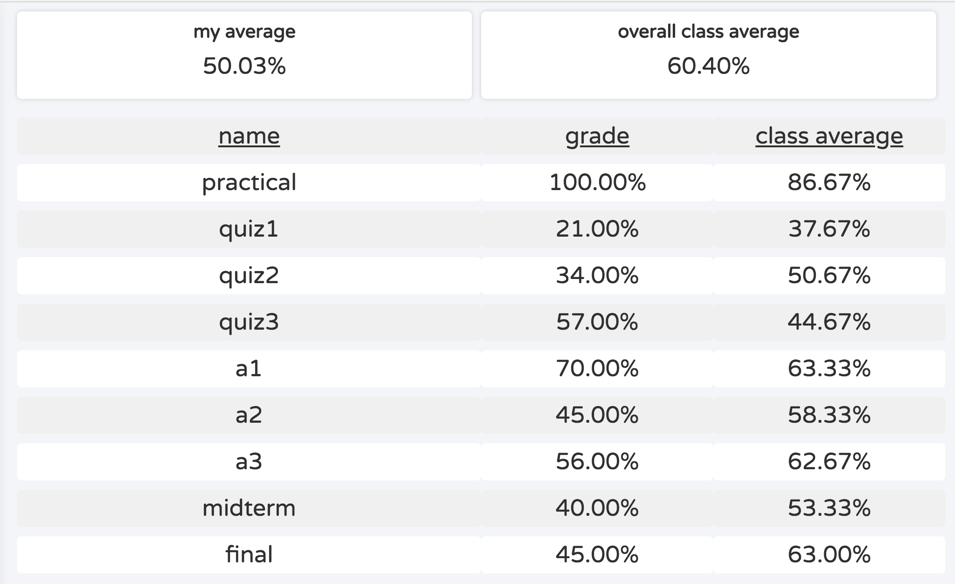
## Challenges

* One challenge was deciding what info on a single page to hide or display, depending on who is logged in. We had looked at how much common code and elements we had on a single page, and if it was not much overlap, we made a new page for the students and instructors. The student would be directed to the student marks page for example, and the instructor would be directed to a different page.
* Another challenge was building on top of our old website. We needed to make the new website easily accessible from the old website but also keep it separate as it contained account-related content. We also wanted to make sure only people logged in could access the old site. To tackle this we used a *session* to keep track of who’s logged in, and redirected them to the login page if they weren’t. We also created a dashboard page that housed all the new additions linking to the db, and made this dashboard accessible via the old website and vice versa. The dashboard is the main control center to navigate between pages.
* Another challenge we faced was debugging our code as PHP is not really well-suited for that. When something went wrong, it would just not load the page instead of telling us what line the problem was caused. In order to fix the error, we would have to try to *echo* every line until we came across the line causing problems and debug that in depth. There is no console where we can print the value of variables, so they must be echoed which is time consuming and a hassle.
* One technical challenge we faced was getting all the marks updated for each student in the instructor view. When hitting the save button we needed to get all the student id’s and their respective marks from text input fields and write an SQL query so that each student’s marks were properly updated. This was a challenge to do as PHP and JavaScript each had their advantages. JavaScript lets you easily get the input values inside a div representing each student’s marks and PHP lets you write queries to update these marks. We ended up solving this by writing a nested for loop in PHP that would allow us to get all the students, and within each student get their marks. Then we constructed an SQL query to update these given marks for the given student. This looped for every student in the class, regardless of if the marks were changed or not to ensure no data was lost.

## Learning

* One new thing we learn was how nicely MySQL integrates with PHP. All the commands are built in and connecting and querying is much simpler than expected.
* Another thing we learnt was how websites approach security, as we also integrated auth tokens in our login and hashed passwords with SHA256, to ensure maximum security.
* We also learned how websites keep us logged in, as we also integrated the same technique. We created a session and stored all the user info in the session, which we could access on any page of the website.
* We learned how to properly use POST and GET requests to share and send data to and from other pages within the website and to the DB. We used POST requests to send form data and GET from URL params for referral paths to know where the user came from, ex. Sign up, logout, etc.
* We also learned how to integrate PHP with HTML by creating HTML DOM elements within PHP as well as getting HTML element values from PHP.
* We also learned how to create alert dialogs in a *hacky* way as PHP does not natively support them, so we had to inject JavaScript to do that for us.

## Instructions

* When accessing the website, you will automatically be redirected to the login page. You must login before you can access anything on the website. The current instructor account that has been setup has the following credentials: *username:* **attarwa** *password:* **ilovephp**. Make sure to select the type of account when logging in at the top.  
  
* You can sign up for a new account by clicking the text below the button. Your account will need to be approved by an instructor before you will be able to login. To approve requests, login as an instructor and click on the approve checkbox for each student you wish to approve.  
  
* Once you are logged in as instructor or TA, you can view remark requests submitted by students and respond to them right away. You can decide if you want to update their mark and send them feedback right and also evaluate the request with the given info.  
  
* Once you mark it complete, it shows up below so you always have a record of the requests and the student also sees the same view on their end. The student will also see when the request is pending for when you haven’t responded to yet.  
  
* When a new student or TA requests to join the class by signing up, only an instructor can approve or disapprove requests. In the join requests page, you will see the following when there are students or TAs waiting for a response.  
  
* Submitting feedback as a student, you can choose which instructor it goes to like this.  
  
* When submitted, the instructor can view the feedback on the same page and it will show this. Note that it does not show the name of who submitted it, and the name or any info is not even stored in the database, making it truly anonymous.  
  
* Students can view their current mark in the class, along with a break down of each assessment mark and the overall class average to see where they stand. It is shown to the student like this.  
  
* Instructors and TAs have access to everyone’s mark and can add marks or update them all at once.  
  