

Report – Assignment 3

Wilson Mak & Hassan Tariq

Prior to beginning, we roughly planned out the general design aspect and functionality we wanted to add for assignment 3. The features we were able to include were:

- Pre-existing features from a2: We extended and kept these features concurrent in this assignment as well. This includes the overall layout; the navigation menu in the header, hover effect on the links/buttons, flexbox layout for webpages allowing for versatile and clean UI experience, and effective implementation of organizing data on each webpage for the web application.
- Login page: The user is prompted to log in before navigating any webpage, with their username and password. We were able to achieve this using a built-in function of flask known as session. The username and password are also checked against the SQL database, that holds data for both students, and instructors, such as their name, username, and password.
- Registration Menu: A new student, or instructor, is able to create a new profile if they are new to the site and are given the option to register below the submit button on the login webpage. The new user is prompted to enter in a unique username, which is checked against the database and prevents duplicate usernames across all users (instructors and students), along with a password which must be at least 8 characters long. Once registered, the user is redirected back to the login page to log back in and is also shown a “successfully registered” text when on the log in page.
- Error Messages: When the user enters a duplicate or invalid username, or an invalid password, when registering and/or logging in, an error message appears on the same page at the top, and allows them to attempt logging in or registering, again.
- Security Feature: The user is not able to navigate to any webpage by modifying the URL and accessing pages through the URL route, as each @app.route checks to see if the user is still logged in, before redirecting them to any webpage.
- Instructor Grade view/edit: Instructors are able to edit, view, and manage grades and remark requests for all grades, **for their particular students only**. The instructor is also able to generally see averages of each student for labs, assignments and term tests, for all evaluations. JavaScript is used to allow the instructor to click on any evaluation average, and it expands to show all grades for each evaluation for that particular student. Remark requests are shown at the end of each row, that display the evaluation requested to be remarked, and the reasoning given by the student. The instructor can edit the grade if the remark is approved.

Note: default grades for new students registered is -1.

- Student Grade view/remarks: The student can view their grades, all of which are outlined clearly on a div-table. The remark form is below the listed grades, where the student can select the evaluation from a dropdown menu, with the exception of the final exam (assuming that regrading for final is done through registrar) and submit a reason in the text box along with it, outlining reason for remark. Once submitted, a message of confirmation is displayed, and the evaluation is removed from the remark’s dropdown menu. Duplicate remarks are not allowed for any evaluation, even after remarks are granted.

- Student Feedback: The student can fill out a form for the instructor of their choice, and answer questions to provide feedback to the instructor.
- Instructor Feedback: Instructor can click and view on individual feedback provided from each anonymous student. Feedback is shown to the particular instructor for whom the feedback was provided to by the student's choice indicated on the form.
- Greeting Text/Log out Option: When the user logs in, there is text in the header, displaying a welcome message with their name, along with an option for them to log out. Their name is stored as a session variable (i.e. session['user_name']). This is shown on all webpages.
- Database: All user data on the website is stored, displayed, and updated through SQL queries and SQLite3, allowing interaction with the external database.
- Miscellaneous: Other features include the use of jQuery, to expand and collapse div-table elements of feedback sent by anonymous students to the instructors. JavaScript for instructors to view and edit grades.

Being able to implement all features we had planned for, there were a few added features that we were not able to add due to time constraints and complexity:

- Stylistic Elements: Animated effects to webpages, such as fade in and fade out, for all webpages.
- Login Enhancements: When the user selects their user type as 'student', as opposed to 'instructor', The subsequent dropdown menu which prompts the user to pick their instructor should only be shown to users selecting their type as 'student', and not for all users including instructors.
- User and Grades Enhancements: Adding a field that records the users utoronto email, in the registration menu. Any changes made to grades and remark requests, can be pushed to their email.

% of work:

Following from a2, where we had an overall even split in workload, we followed the same general rule.

- Workload for this assignment was evenly distributed, 50%/50%.
 - member 1 took on the grading aspect. This includes; instructor viewing/adding grades for individual students, instructors managing remark requests and editing grades accordingly, and students viewing grades and submitting remarks requests.
 - Member two did feedback, login page, registration page, and design/implementation of the session object in flask to track session data for users currently logged in. This includes; logging in, registering new users, student form to submit feedback for particular instructor, displaying feedback to particular instructor.
- Both team members planned the general design for the added features.