Introduction:

We (Bomin Kim, Taemin Lee, Paul Nguyen), a team of first-year students, have dedicated the past three weeks to our first major project. Our goal was to create a website that closely mirrors real-world scenarios, particularly focusing on its relevance to the CSCB20 course and potential applications in future courses.

Tech stack:

Front end: HTML, CSS, Javascript, JQuery
Back end: Python Flask, Flask-SQLAlchemy, Flask-Bcrypt

User Design:

- We have significantly enhanced the navigation bar compared to our previous version. It now features a new color scheme and is designed to be more user-friendly and responsive. Additionally, we've added a secondary navigation bar within the footer to allow users to easily access other tabs without needing to scroll back to the top of the page.
- We've developed separate interfaces for instructors and students. Both types of users are required to have an account to access any content on the site. If they try to access pages without authentication, they will be redirected to a sign-in page. If authentication fails, users will be prevented from viewing the website's content, ensuring that only authorized users can access course materials.
- For students, there is also a dedicated page where they can view their grades, submit regrade requests, and see all their requests.
    - ‘My Grades’ page is categorized into two sections, by assessment type (i.e. assignment and exam) for the user's convenience, and the assessment name, mark that the user received, and assignment due date/exam date are the information provided to the student users
    - At the end of each row for assessment, the ‘Remark Request’ button is placed for the student users to send remark requests right in place, instantly after viewing their marks, without having to navigate further.
    - Once the ‘Remark Request’ button is clicked, a pop-up window with a brief instruction for remark requests and a textbox to write comments shows up for the users to express their concerns about grading, which are sent to the instructors.
    - The ‘My Remark Requests’ page is designed for the students to receive the instructors’ responses to their remark requests. Similar to the ‘My

Grades' page, the remark requests are displayed in rows, with a button at the end that shows a pop-up window for comments. Only the latest comment is displayed to prevent confusion.

- For the instructor side, they have a unique page for instructors to manage grades, view regrade requests, and view anonymous feedback
    - On the 'Manage Grades' page, the list of the assessments is displayed for the instructor to choose to manage grades for.
    - Once they click the 'manage grades' button beside each assessment, now the list of the students who have completed the evaluation is exhibited, so the instructors can view or change the grades of each student in that specific assessment.
    - To alter the individual student's grade, the instructor can simply click on the displayed current mark (i.e. click on the number '87' if the mark is displayed as 87/100) for the new grade input box to appear. The new grade that the instructor types in that box will get updated once the instructor finishes typing and clicks somewhere else out of the box so that the box is 'out focused', or the instructor presses the enter key to indicate that the new mark is inputted. This update of the grades will be immediately updated on the current page, as well as the database.
    - The 'View Regrade Requests' page is designed in a similar fashion to the 'Regrade Requests' page in the student interface, but with functionality to add comments.
    - In the 'View Anonymous Feedbacks' page, an instructor can view the anonymous feedback that is specifically sent to him/her. We listed the feedback for each of the four questions in four different columns, in the boxes with varying heights. This design was created not only for a visually appealing user interface, but also for the instructors to be able to check all the feedback on one screen, all at once, only by scrolling down without having to navigate through complicated paths, as we thought that this would help the instructor getting the idea of how and where to improve, or enhance about the course, by looking at multiple opinions from different students collectively.

Database Design:

In **modules/model.py**, we structured our database into seven schemas, with a focus on flexibility and ease of expansion. The main schemas are for assignments, exams, login authentication, and student feedback. Splitting assignments and exams into separate schemas allows for efficient querying, especially considering the potentially higher volume of regrade requests for assignments compared to exams. Additionally, the login

schema securely stores user authentication information, and the feedback schema anonymizes student comments to instructors.

Technical aspect:

Utilizing Blueprints for the backend architecture has proven instrumental in organizing our codebase. This modular approach streamlines development, making it easier to manage and scale the application. Comprehensive documentation for helper functions ensures that developers, including future contributors, can quickly understand and modify the codebase as needed. Moreover, integrating JavaScript for sending POST requests has significantly enhanced the website's interactivity, providing users with a more dynamic experience.

Website Limits:

Despite our efforts to create a robust website, we acknowledge certain limitations. Overall, our website lacks security. For example, our use of raw SQLite queries makes the site vulnerable to SQL injection attacks. Additionally, the absence of an Object-Oriented Programming (OOP) approach and UML diagram at the project's inception has resulted in less organized code, which could impact maintainability and scalability.

Challenges:

One of the significant challenges we faced was effective communication between frontend and backend developers. Differences in UI design choices often complicate backend implementation. To address this, we implemented regular meetings to collaboratively decide on the website's design and structure. This approach ensured that both frontend and backend developers were aligned in their efforts, leading to a more cohesive final product.

Future planning:

Moving forward, our goal is to further enhance the website's functionality to meet real-world requirements. One of the key features we plan to implement is a search function, allowing users to easily find specific content within the website. Additionally, we aim to expand the website's capabilities to support the upload of various file types, such as images, text files, and videos, for lectures. Furthermore, we plan to incorporate

a feature that enables instructors to view students' work alongside regrade requests, providing them with valuable context when evaluating these requests.

Work distribution:

We've included a screenshot from our GitHub repository that shows each team member's contributions to the project. However, we want to emphasize that the number of commits does not necessarily reflect each member's actual contribution to the final product. Each team member has taken on an equal amount of tasks based on their complexity.
Here is how tasks are distributed:

- Paul (nnhphong):
    - Design database schemas
    - Build and architect the backend
    - Create helper functions for modifying the database
    - Connect the backend with the frontend by handling POST/GET request
- Taemin (taeminlee0601):
    - Redesigned the navbar to make it responsive
    - Designed the UI of the instructor remark request page
    - Designed the UI of the instructor feedback page
- Bomin (Bomin75):
    - Improved the overall UI of the original website from the second assignment
    - Designed the UI of sign in/sign up page
    - Created a html template for student/instructor interface (user_template.html)
    - Designed the UI of My Grades/My Regrade Requests/Manage Grades page

Feb 25, 2024 – Apr 5, 2024

Contributions: **Commits**

Contributions to main, excluding merge commits