Introduction

The assignment was given to us in 3 parts (Login page, Student page, Instructor page). So each of us took a part and worked on it. Login page was done by Skandan vecham, Student page was done by Fardin Alam and Instructor page was done by Danyal Rana.

Features that worked and didn't work

Most of the features we set out to implement worked, the Student page and Instructor were implemented as expected. However when it came to the login/register pages we could've done better when a user would create a password and set some conditions so that the password is secure(For example including a capital in the password and a symbol and digits). For the instructor page we did what the handout expressed where we had a form like structure where the instructor can edit the grades and when the instructor submitted the changes the information would update in the database. What took us 3 the most amount of time to do was figure out how to update the information in the database and how to implement that code in the python.

Fardin Alam (33%)

Student.html - After student logs in, he is greeted by a welcome back message along with their name. It then gives two options to view grades and provide feedback. If a student clicks grades then they will be presented with their grades form a1-a3 and underneath that he will see a drop down menu where he can choose which assessment he would like regraded, and a text box underneath asking why he thinks it should be regraded. Once student submits the feedback he should be redirected to a page saying that its successful. If student chooses to leave feedback, he will be taken to the feedback webpage where he will be prompted to choose which instructor to leave feedback for. When the feedback is left and student press submit the feedback will be updated in the database.

Helped create report, and created corresponding student functions in app.py

Danyal Rana (33%)

Instructor.html- After the instructor logs in, it gives options to show regrade requests, feedback, to view students grades and to change them. If instructor chooses to view grades, they will be able to see all students grades in a grid and if they choose to see regrade requests, they should be able to see all the requests from students and then go to change grades accordingly by entering the username and choosing the specified assignment and then inputting the newgrade. The changes should also be viewable on the students side. If the instructor chooses to view feedback he should be able to see all the feedback given by students Anonymously.

Helped create report and created corresponding instructor functions in app.py

Skandan Vecham (33%)

Login - Followed from by flask website and lecture. When a user enters their username and password. It checks that the username/password pair are in the database. If it is NOT then the user is redirected to a page that says invalid username or password and has a link that takes them back to the login page. If it is in the database then the user is taken to the respective instructor/student page.
Logout- When the user presses logout it clears the session and takes back to the login screen. The flask runs in an infinite loop until they close the window
Register- When you press create account there's a check box where you can select instructor account or student account. If a username is taken it does not create the account and says invalid username.
Created the database and helped with the report.