All 3 of us contributed 33.33% to the entire project.

Summary of Features

General

- Pages don't crash when refreshing as we handle both GET and POST requests.
- We used flexboxes and div elements to make tables and our header/footer, no table elements were used.
- We used a proper schema to make the database along with a blueprint for the application on how the logic is supposed to work.

Types of Users

- We have two different types of users: Student and Instructors
  - **Students** have access to all the content from our website from A2 (things that were meant for students) along with a new Page called "Grades Summary" where they can view their grades for assessments, along with any re-mark requests they would've submitted and a form to submit remark requests.
  - **Students** also have another new page called "Send Anonymous Feedback", where a form lets them a course instructor to send feedback to and then can respond to 4 questions and submit the feedback directly to that instructor.
  - **Instructors** have a page titled "Check/Update Grades", here they can view the grades of every student registered to our web app and can modify them using an update form. They also have a page titled "Re-Mark Requests" where they can view all the remark requests submitted by students registered to this web app and they can resolve those requests and add comments along with rejecting or accepting the requests. As per the actual re-mark they can use the "Check/Update Grades" to change the grades.
  - **Instructors** also have a page titled "View Feedback" where they can view all the anonymous feedback addressed to them by their students.

Registration and Login

- No one can access any of the course content or the instructor pages unless they are registered on the website. When people first access the website they are shown a reduced version of our home page with hidden content that requests them to sign in or register before moving forward.
- People have the ability to choose what type of user to register, the passwords are hashed into the database using flask-bcrypt. Email addresses and usernames are unique, no student or instructor can have the same username or email (that is checked).
- While logging in, users select whether they want to login as students or instructors.
- Depending on how you login in, you are shown respective content

- We have put restrictions to ensure instructors can only view content they are supposed to see and the same for students, if someone tries to change the url to open pages that they’re not authorised for or when they are not logged in, they will be given a flash message and be asked to login in.

Work Divided Amongst Us

- Emin and Varun: Designed the schema
- Alejandro and Varun: Designed the logical blueprint of the website
- Emin and Alejandro worked on the registration and login
- Varun worked on the instructor special pages (“Check/Update Grades”,and “Re-Mark Requests, “View Feedback”)
- Emin and Varun worked on the CSS and HTML for the template, created the div table elements, header, footer and the code to restrict access to certain content or pages depending on what type of user is signed in or if no one is signed in.
- Varun, Emin and Alejandro worked on Student pages “Send Anonymous Feedback” and “Send Anonymous Feedback”
- Varun and Alejandro worked on integrating the content from our old website from A2.
- We all tested the web app multiple times