REPORT

Work breakdown: We divided our work according to the front-/back-end of our application: 35% Carlos (mainly backend), 35% Jeffrey(mainly frontend) and 30% Jenny(mainly backend).

Features description:

1. User login with username/password with different users (student, instructor)

We have two types of users stored in our sqlite database: student and instructor. 'get request' is supported for both types of users.

Some pre-defined student users in our database:

	student_id	username	password	firstName	lastName	lectureSe
1	121	Chavez	Chavez00	Tarek	Chavez	
2	128	Zhang	Zhang00	Larry	Zhang	
3	321	Williams	Williams00	Allan	Williams	
4	345	Shankar	Shankar00	Tom	Shankar	
5	543	Brown	Brown00	Marsha	Brown	
6	553	Peltier	Peltier00	Ashton	Peltier	

Some pre-defined instructor users in our database:

	instructor_id	username	password	lecture_section
1	10000	LyndaBarnes	GraduateOffice	1
2	10001	SteveEngels	sengels	1
3	10002	PaulGries	pgries	2
4	10003	DanHeap	heapify	2
5	10004	KarenReid	reid	3

2. Register a new user. Try creating a new user of both types (student, instructor).

We decided to disable this functionality (Do you see the required changes on the database? If you change any of the hard-coded values in the database, does it get reflected again on the front end? If you change the password or the user name, can you log in again?) for security reasons, because once the course begins, the instructors should be fixed and not subject to change. More importantly, we expect that the instructor will get all the student users from the database directly instead of allowing for other users outside of the students list to be created by random users from outside.

Once a student or instructor enters correctly their username and password, they would be able to see the course page. Different types of users will see slightly different course pages due to their different roles.

3. Login using a student account, the route: "/scores" allows the student to view his/her mark. Our link is working as well. If you change any of the hard-coded values in the database, it will get reflected again on the front end. If you change the student mark in the database manually, it will also get reflected in the frontend.
4. Students are able to submit a remark request, by clicking on the remarking button. They are provided with a form and will be prompted of the result of the submission.
5. Students can submit anonymous feedback to the instructor. This information will be stored in the "Feedback" table of the database. Students can press the "anonymous feedback" button to submit this request.
6. An instructor can see all the grades of his students by clicking on a special link. All the information is shown on the webpage, and only users of type "instructor" can see this information.
7. An instructor can see all the anonymous feedback given by the students.
8. An instructor can see all the remarking requests and can enter marks for the students. All the changes can be correctly reflected in the database.