

CS4540 A4

Ken Wang

[Github Link](#)

Note

- There's no need to do any migrations or database-update before you start the program. It will be done at runtime when the program starts.
- For the Instructor test link, please use professor_mary@cs.utah.edu to test, since other instructor won't have access to mary's classes.
- !!! PLEASE use the school network to test the Email functionality, it would require authentication if using the AnyConnet VPN.
- In order to show the Collapsable function, in CourseList, the card besides the Fall-2019 one are still static, due to lack of seeding courses. Will consider adding more next assignment.
- Progress bar now is still generated randomly, to show some variants of the table.

Authentication and Authorization

It's nice to have this membership system to handel password and login informations. Where it would take forever to make a secure password storing place if we need to do it our own. I found it interesting that most of the data stored in the User Tables are hashes, not literal values. Which means most of the things are "Secure" in most of the times. I am curious on how did they encode and decode those hashes, and I've took some really general overview readings about it.

For Autohorization, there's Role based, Claim based, and Policy based Authorizations. Which surprises me that there's that many different ways to do one simliar thing in one framework. But since Role-based is what we're using, and it turns out to make the most sense to me after reading their introductions. Role-based is simple to understand and straight forward, while others like Claim-based can be used in different situations where we're not authorizing basing on the "Object" but part of the "Value" of the "Object". Which is really interesting to think about the use cases of it.

And the really convinient thing I learned about .net core Authentication is that Even if you do a Authorize at the Class level, you can still make changes in the method level. This way you can expose some of the Actions in the controllers to the ones you want, which become pretty nice in a situtaion like this where the controllers are kind of used by many roles, but not just a single role.

Above and Beyond

- At register time, I made the user to put their First and Last name to register, so that now instead of having "Professor professor_mary@cs.utah.edu" , we got "Professor Mary Hall". Which is much nicer to everyone.
- Courses are listed in Numeric order, just a programmer habbit. Though it needs to be in Chronical order later...

- I think I did well on the Role Status page, where I really like this UI I tried out. It's straight forward and it looks cool! I would like the Admin page to look more deep and dark, because it's always more serious to be a Admin...
- Also the "bonus karma" (a list of all courses an instructor is teaching) mentioned in the Assignment is what I've done for Above and Beyond.

Improvements

- Now it's using "Migration-based" DB instialization, instead of using database.ensureCreated.
- Made ALL the UI pages dynamic pulling from the database and transfered it to the UI I made for A1/A2. (Except for the static data to show the "collapsible" functionality in course list)
- Improved the footer so that when there's less content in the page it won't go to the middle of the page but will stay at the "bottom"
- Improved Create for LOS to have the user select "Course Names" instead of "CourseCid".
- Removed all the hard-coded "routes" and all links are now Controller based.
- Changed some UI some all pages should all look like a simliar style.

Next-Step

- Make Admin life easier to beautify a little on the CRUD pages. Considering to integrate LOS CRUD to Course CRUD, so Admin can directly edit LOSs in a Course, instead have to go to LOS CRUD.
- More seeds to display the "Collapsible"