Assignment 2

Billy Davies

S230525

## Github Link : <https://github.com/BillyDavies1122/hit339Assignment2>

## Features and how they are implemented:

**As a member enroll in a class**

I used a class in the Schedules controller to do this. First a user clicks enroll and it loads the schedules index view. Then they click the enroll button, which is sent to the schedules controller. Then the date in the session is compared to the current date time, if the current time is less than the session time it’s allowed otherwise nothing happens.

If the enrollment is allowed then a new scheduleMember is created and added to the database , then the user is sent back to the index page.

**As a coach view my classes and who is enrolled**

The user clicks the My sessions(coach) button , then the schedules controller gets the current users email and sets the view bag variable to it. Then sends them to the mySchedules view, in this view it loops through all of the schedules and only displays the ones with the same email as the logged in user (the coach). It also adds a button to check who’s enrolled.

This link calls the checkEnrolled function in the ScheduleMembers controller which then adds the id in the url to the viewbag variable and sends it to the checkEnrolled view. Then it loops through each item in the ScheduleMembers model but only displaying the data from the relevant schedule.

**Create an account**

This uses the inbuilt identity framework, and the data imported from the azure server.

**Edit coach profile**

Uses an edited version of the scaffolded edit page for the coach model, but only displayed the logged in user. Did this by obtaining the current users id and only displaying that user in the list

**View coaches & members**

Uses the scaffolded index page to display a list of either.

**Create new Schedule**

Uses the scaffolded create page to create a new page.

**Roles**

Failed to implement these , you can see the attempt to add an admin role in startup.cs in the CreateUserRoles function. The function is then called on like 97 , which runs but doesn’t seem to have any effect as I cannot user [Authorize(roles=”Admin”)] to specify which role should have access.

**Mobile View**

Each view has the viewport tag on it so that is responsive and resizes depending on screen size. Tables are tricky on mobile, but you can scroll left and right to view the extra data that isn’t displaying.