## Notes on Final Exam

### What should you expect on the Final exam?

* Hands-on programming exam
* Focused on chapters since midterm:
  + Validation
  + EF Core & LINQ
  + xUnit
    - Note: Custom services won’t be on the final
  + Componentization - i.e. tag helpers, partial views & view components
  + Authentication & authorization
* Just to be clear, that means:
  + Cookies and Sessions are ***not*** on the final
  + And neither is anything about *async* - i.e. asynchronous operations - except as it might pertain to authentication/authorization
* You will be building 1 app, same as with the midterm, and there will be a series of parts, basically 1 per topic in the previous list, that specify what you need to do.
* You will be provided with a functioning app as a starting point that you will be asked to build on. It has a significant amount of code already implemented.
* The latest state of our Movies app (that we have continually evolved in class) and the apps of assignments #2 and #3 all represent something very close to the kind of complexity you should expect on the exam.
* While the focus is on these later topics you can never forget the foundational topics from before the midterm!

### Good apps for review:

* The continually modified version of the authors Movies App (built on since ch4):
  + Good for EF Core:
    - full 1-to-many between Movies & Genres with a link on Genre in all movies table that takes user to Movies of that genre
    - full many-to-many between Movies & Actors (called Casting), namely an entity class that records the fact a given Actor was cast on a given Movie + add action link to all Movies view to take user to view all the actors that were cast in that movie
  + Plus both assignments #2 and #3 good for EF Core
  + Also good for tag helpers (last action message), partial views (Movie action links) & view component (TopRated movies)
  + We also went through the process of adding authentication & authorization to the Movies app
* In terms of xUnit, it will be very light - the test infrastructure will already be in place and you will simply have to complete a test
* Version of Murach Registration App:
  + Does most if not all of the validation attributes
  + Plus asst3
* A good suggestion might be to think up a simple app that involves a primary entity that has a dynamic 1-to-many relationship with another entity
  + E.g. a Team has any number of Players
    - Remember the guiding code example here is a Movie with any number of Reviews