**Capstone**

## **Backend: C# API**

Groups must create a backend API that includes:

* At least 4 database tables
* At least 1 join in a query
* A Data Access Layer injected into the API controllers
* Dapper for Database ORM
  + Stored Procedures for some/most of the database work
* At least 3 API endpoints
* A mixture of HTTP methods including at least one example of GET, POST, PUT/PATCH, and DELETE

## **Front-End: Angular & CSS**

* At least 6 components
* Angular routing between components
* At least two services
* Uses your API
* Consistent, attractive styling beyond the default C# Angular project template
  + Use of Bootstrap or Spark optional

## **Workflow Expectations:**

* All tasks/user stories should be worked on a separate branch that is merged into main branch
* Branches should correlate with a user story or task found on your Trello Board
* All branches should have a pull request and be reviewed by one team member before merging to main branch

## **Steps to Take Beyond MVP:**

* Improve your API error handling – return more information in the response and/or HTTP error codes
* Further error handling in Angular – if you get an error from an API, don’t just send it to

the console, display information to the user

* Refactor your code – cleaner, more documented, more modular
* Improve your CSS – really make the sire pop. Dig into CSS animations
* If it makes sense with your project, incorporate a third-party API

Note: Please notify us of any major changes from your initial pitch

# **Fallback Project**

While we strongly encourage every group to develop a unique project that meets the above requirements, this serves as a fallback for any group that cannot come up with something.

* Select a specific user from a drop-down. An individual user has one of three levels of access:
* Regular users can:
  + View a list of courses
  + Enroll or unenroll from a course
  + Enter a course they are enrolled in and see modules, assignments, and resources
* Admin users can:
  + Create a course
  + Assign a teacher to a course
  + Create a new user
  + Assign teacher or admin rights to a user
* Teachers can:
* Add modules to a course
* Add assignments to modules
* Add resources to modules
* Rearrange the order of materials within modules and between modules
* Remove students from their courses

Ideas for stretch goals:

* Teachers can create quizzes in their courses, and regular users can take them (Sequence of multiple-choice questions?)
* Teachers can create slideshows in their courses and regular users can navigate through them
* Teachers can post YouTube videos in their courses
* Incorporate your own ideas

ram Managers.