**Where we are:**

|  |  |
| --- | --- |
| Intro to MVC | Navigation, Searching and Filtering |
| Controllers and Views | Forms and Tag Helpers |
| Unit Testing, Publishing to Azure | Annotation and validation  Azure SQL Database |
| Dependency Injection, Models | Authentication |
| Entity Framework and Scaffolding | Authorization |

**Announcements**  
A beta version of your term project is due Tuesday of Week 10 (3/14).  
In addition to the feature requirements listed in the Term Project Requirements document, I expect you to use “best practices”, which means using all the things you’ve learned in this course.

**Publishing to Azure with a Database**

* Using the Azure Portal:
  + Create a SQL Server in your resource group
  + Create a database for your server
  + Copy the connection string
  + Optional: Enable error pages
* In Visual Studio
  + Modify appsettings.json with the connections string for the Azure database:  
    Server=tcp:briansqlserver.database.windows.net,1433;Initial Catalog=BookInfoDb;Persist Security Info=False;User ID=profBird;Password=LCC123lcc;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
  + Modify project.json so that appsettings.json will be published to Azure:
    - publishOptions

"publishOptions": {

"include": [

"wwwroot",

"web.config",

"\*\*/\*.cshtml",

"appsettings.json"

]  
}

* + Add the connection string to the publish dialog in *Settings / Entity Framework Migrations*. This will create tables in your database(s).
    - Or, you can do the same thing by running update-database on your local machine. For example, in the VS Package Manager Console, you could execute:  
      Update-Database -Context ApplicationDbContext -Environment Production
  + Now publish it!