# MVC Project

## Blog

## Task

Build a fully featured blog for your personal site.

Every Friday from this point forward, your homework will include a blog post to your site about what you've learned that week. This will become a sort of progress journal of your experiences in the course, as well as a means by which potential employers and the staff at Coder Foundry may observe and monitor that progress.

## Requirements

This project will be a new Visual Studio MVC project with individual user authentication. You will add a link to this project through a menu item on your personal site. Make sure you use the same Bootstrap template for this project as you did for your personal site.

1. Use Entity Framework for user authentication and management.
2. You will implement two Roles: Admin and Moderator.
    a. Admin users may post new blog entries, edit existing entries, as well as add, edit, and delete comments to entries.
    b. Moderators may add, edit, or delete comments to entries.
    c. All other *authenticated* users may leave comments to entries.
        i. Users wishing to leave comments may login with a 3rd party account only (Facebook, Twitter, Google, LinkedIn, GitHub, etc.). You may use any of these 3rd party options, and as many as you like, but please include at least three options for users.
    d. Anonymous users may view entries and comments, but not submit comments.
3. Seed the database with the following data:
    a. Create your own personal login (email, username, password) and assign it to the Admin role.
    b. Create the following Coder Foundry login (email and username, with password "Password-1") and assign it to the Moderator role:
        i. moderator@coderfoundry.com
    c. BONUS: Try to figure out how to require the Moderator to change his/her password on first login.
4. Your data source is a new database associated with your personal blog.
    a. Blog entry table should have fields such as the following:
        i. Id
        ii. Creation date

        iii. Updated date
        iv. Title
        v. Body text
        vi. Media URL (if you want to include pictures/video/etc.)
    b. Comment entry table should have fields such as the following:
        i. Id
        ii. PostId
        iii. AuthorId
        iv. Body
        v. Creation date
        vi. Updated date
        vii. Update reason
    c. You may feel free to design your database differently if you so desire.
5. Database queries not generated by VS should be handled through SQL stored procedures.
6. Provide a text-based search feature (call a stored proc) that allows a user to search through all blog posts and comments for the search criteria.
7. You may include JavaScript, jQuery, or AngularJS for front end dressing if you desire, but it is certainly not required.
8. You may design and test your database in SQL Server Management Studio, but you must also construct a Code-First Database model in Visual Studio.

## Questions

1. How will you display search results?
2. How will you handle multiple matches?
3. How will you design your comment interface?
4. How will you design your blog post creation interface?
5. How will you order the posts on the page?
6. How many posts will you display at one time, and how much of each?
7. Will you display all posts on a single page, or will you provide paging controls and only display a set number of posts on the page at a given time?
8. How will you display a detailed single-post view?