Razor allows you to not only leverage existing tags but also co-mingle your Views with C# code. Provide 2 anti-patterns that could occur if too much C# code is present in your Views. Comment on at least two other students’ solutions.

The most common anti-pattern when looking at the Views portion of MVC is overloading the view with specific business data that would normally be contained within the models. There are some things which need to be rendered within a view such as taking in form data, but any code which takes that form data and modifies it should be done in a method within a model class as best practice to avoid this common anti-pattern. Even making a new model just for the specific purposes of the view objects to pass data to is a much better practice than bloating the view itself. Properly leveraging Action attributes and HTML helpers is another way to avoid excessive code within a view. HTML helpers and action attributes can be invoked to increase efficiency as well as preventing actions being hard coded inside the view each time that action needs to be completed.

Dresser, M. (2015, August 15). Anti-Patterns in ASP.NET MVC. Retrieved January 15, 2021, from <https://www.matthewdresser.com/asp.net-mvc/anti-patterns>

Galloway, J., Wilson, B., Allen, K. S., Matson, D., & Haack, P. (2014). Views. In *Professional ASP.NET MVC 5* (pp. 49–74). John Wiley and Sons.

Patel, P. (2014, February 24). Three Layer Architecture in C# .NET. Retrieved January 01, 2021, from https://www.codeproject.com/articles/36847/three-layer-architecture-in-c-net-2

Zhelev, D. (2020, September 08). The Role of Layers in Web Applications. Retrieved January 01, 2021, from https://mentormate.com/blog/the-role-of-layers-in-web-applications/