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.

Hello Class,

From what I have read, putting any sort of business login inside of a view is considered an anti-pattern. Adding business logic into your view would require data expensive api calls/requests which would cause high network latency, even more data transfers and cause the same data to be read than if the logic were already inside of the model. Another would be adding data access into the views ie. Database queries and such. This in itself can cause problems in the design making it rigid and decrease the overall performance of the application. When developers include this sort of code into the view creates tight coupling that makes it almost impossible for the view to update in response to the data code. This would also create many countless un-needed database calls, adding more performance issues due to connection to the database being opened and closed multiple times.

References:

Fowler, M. (n.d.). Separated Presentation. martinfowler.com. https://martinfowler.com/eaaDev/SeparatedPresentation.html

.Net MVC - Accessing the Database from within a View more than just Bad Practice? (n.d.). Stack Overflow. https://stackoverflow.com/questions/20248494/net-mvc-accessing-the-database-from-within-a-view-more-than-just-bad-practice

Writing business logic in view. (n.d.). Software Engineering Stack Exchange. https://softwareengineering.stackexchange.com/questions/335880/writing-business-logic-in-view