* Microservices – This is an architecture that splits needed services into its own separate container. Each is run as a service, but what does run as a service mean? Elaborate on this. They communicate via APIs, but what if I didn’t know anything about APIs. Give me a short sentence or two on what an API is? Code is not shared between services and you are able to write code specifically to the type of service offered like if this one service needs a certain language, you can write in that language, but if another would work better with another language or technology then you can use those in the other microservice container. Splitting this up means that different people can work on different microservices. Is this because they’re independent so what you do in one doesn’t affect another. I like the example from amazon. They use microservices in many different ways, but maybe explain why we would want to split things up into microservices and apply this concept more to the project in your knowledge transfer.
* AWS Knowledge Check – This looks like some kind of copy and paste maybe from the Amazon website. This would need to be cut down and summarized and applied to the project more. How are we using AWS in the project? Why are we using it in the project? What do we need to know of how it applies and is used for the project?
* MVC – State explicitly how this applies to our project. Summarize this into things that are easier to digest by applying this to our project. Provide sources.