Ch 6 – Discussion Questions – ASP.NET

1. Why do we want to validate user input on the client? Why do we want to validate user input on the server?

**Answer:** Because you must have validation logic running on the server and client validation logic gives users instant feedback on the information they enter into a form. The server validation logic is in place because you should never trust information arriving from the network.

2. In ASP.NET MVC, what data items are the focus of validation?

**Answer:** The data items that are the focus of validation in ASP.NET MVC are model values.

3. Where do data annotations live in the .NET framework?

**Answer:** Data annotations live in the System.ComponentModel.DataAnnotations namespace.

4. Can you stack multiple validation attributes with regard to actions? If so, how many can you stack?

**Answer:** Yes, you can stack multiple validation attributes with regard to actions. No regard to how many.

5. Explain the syntax of [remote]. What does it do?

**Answer:** It enables you to perform client-side validation with a server callback.

6. What is a server callback?

**Answer:** A server callback is where a client-script function sends a request to an ASP.NET Web page. The Web page runs a modified version of its normal life cycle. The page is initiated and its controls and other members are created, and then a specially marked method is invoked. The method performs the processing that you have coded and then returns a value to the browser that can be read by another client script function.

7. What is a model builder? When does it run?

**Answer:** A model builder is a class that contains runs types and methods. It runs implicitly. (p.147)

8. What is the model state? When is it created? How is it created? What does it contain?

**Answer:** It comes from the request parameters

9. What does the controller action method generally do if the model state is not valid?

**Answer:** It gives the page back.

10. What are the two options for custom validation? Explain your answer.

**Answer:** Packaging validation logic into a custom data annotation and packaging validation logic into a model object itself.