1. **Describe what a controller does in the MVC design pattern.**

Controllers are C# classes that handle incoming requests, and are usually inheriting from the Microsoft.AspNetCore.Mvc.Controller class, which is the built-in MVC controller base class. Each public method in a controller is known as an action method, meaning you can invoke it from the Web via some URL to perform an action.

1. **What is the ASP.NET MVC convention in naming controllers? What does HomeController.cs do?**

The convention for naming controllers is a name followed by the word Controller. For example, HomeController.cs is the controller called Home, which is the default controller and contains the default “Index” action method that handles the browser’s HTTP request. The result from the method will be sent back to the browser.

1. **What is the name of the routing configuration file? Where is it located?**

ASP.NET Routing is enabled in your application's Web configuration file (Web.config file). The web.config will be placed in the root folder of your website or project. ASP.NET Routing routes incoming requests to controller actions using a route table that is created when the web application first starts. The route table is setup in Global.asax.

When an MVC application first starts, the Application\_Start() method is called. This method, in turn, calls the RegisterRoutes() method. The RegisterRoutes() method creates the route table.

1. **What is Razor? How does Razor treat an expression beginning with the at symbol (@)?**

Razor creates a modern platform associated with the older Web Pages development style. It is ASP.NET programming syntax used to create dynamic web pages with the C# or Visual Basic.NET programming languages. You add code to a page using the @ character when using razor.

1. **How do View methods work?**

The View tells MVC how to generate an HTML response from a browser request. A ViewResult Index() action method returns a ViewResult object instructing the MVC to render a view. The contents of the view file are mostly HTML, except the part immediately following @ which processes the contents of views and generates HTML that is sent to the browser.

1. **What is the purpose of MVC models?**

The purpose of the MVC model is to display dynamic output by separating your views from your controller so that one can change without affecting the other.

1. **What is a strongly typed view and why do we use strongly typed views?**

Strongly typed views are used for rendering a specific model type, instead of using the general ViewData structure. By specifying the type of data, you get access to IntelliSense for the model class.

1. **What is the purpose of setting a start page URL?**

The start page URL is a fully qualified URL and is not used very often. It is meant for testing web services by opening a page that will call into the service, but isn't part of the application itself.

1. **Describe the differences between HTTP GET and HTTP POST.**

HTTP GET displays the initial blank form when someone first visits a web app. It is what a browser issues normally each time someone clicks a link. HTTP POST is responsible for receiving submitted data and deciding what to do with it.

The client “gets” a form from the server so that the user can input information. After the information is entered into the form, the client “posts” the data to the server.

1. **Describe the two approaches to validation in web applications.**

Server side Validation is performed by a web server, after input has been sent to the server. Client Side Validation is performed by a web browser, before input is sent to a web server.

1. **What is the role of Cascading Style Sheets (CSS) in web development?**

CSS is a language used for describing the look and formatting of a document written in HTML. It tells the browser how HTML elements are to be displayed (property: value; background: blue).