1. **What are the referenced key milestones from 1978, 2002, 2007, 2015, and what were their contributions to the current MVC development framework?**

1978: The Model-View-Controller (MVC) architectural pattern was developed as part of the Smalltalk project at Xerox PARC.

2002: The original ASP.NET was introduced in 2002. ASP.NET is a way to host .NET apps, letting you interact with HTTP requests and responses. Web Forms were built on top of ASP.NET to provide a set of UI components (pages, buttons, etc) plus a stateful, object-oriented GUI programming model.

2007: ASP.NET MVC Framework was introduced in 2007. Built on the existing ASP.NET platform, the MVC Framework was intended as a direct response to the criticisms of Web Forms and the popularity of competing platforms such as Ruby on Rails. However, compromises were required to graft the MVC Framework onto a platform originally designed for Web Forms.

2015: ASP.NET Core MVC was developed. ASP.NET Core is built on .NET Core, which is a cross-platform version of the .NET Framework without Windows specific APIs. Makes it possible to deploy ASP.NET Core apps to a broader set of hosting environments such as containers or cloud platforms.

1. **What is the difference between the MVC Framework and the MVC architectural design pattern?**

The ASP.NET MVC Framework implements the MVC Architectural Pattern. More specifically, the MVC Architectural pattern was designed in 1978 to follow a natural cycle: user takes an action, and in response, the application changes its data model and delivers an updated view to the user. Then the cycle repeats. This is a convenient fit for web applications delivered as a series of HTTP requests and responses.

The MVC Framework implements the MVC pattern by combining several technologies (databases, HTML, and executable code) that are usually split into a set of tiers or layers, and reflects emerging trends in web app development such as HTML and CSS standardization, RESTful web services, effective unit testing, and the idea that developers should embrace a stateless nature of HTTP.

1. **What is meant by the term *View State,*and what does the new approach intend to improve?**

View state is the mechanism for maintaining state across requests. The old approach resulted in large blocks of data being transferred between the client and server, and the new approach is intended to decrease bandwidth demands and improve response times of the server.

1. **Briefly describe ASP.NET Web Forms.**

Web Forms were built on top of ASP.NET to provide a set of UI components (pages, buttons, etc) plus a stateful, object-oriented GUI programming model. The main building blocks of Web Forms pages are server controls, which are reusable components responsible for rendering HTML markup and responding to events. View state is used to persist the state of server controls between normally stateless HTTP requests.

1. **Briefly describe ASP.NET.**

ASP.NET is a way to host .NET apps, letting the client interact with HTTP requests and responses. ASP.NET Core is a cross-platform implementation of ASP.NET.

1. **Briefly describe .NET Core.**

.NET Core is a cross-platform version of the .NET Framework without Windows specific APIs. Makes it possible to deploy ASP.NET Core apps to a broader set of hosting environments such as containers or cloud platforms

1. **What is Representational State Transfer (REST)? This will take some independent research.**

REpresentational State Transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services (RWS), provide interoperability between computer systems on the internet and are characterized by how they are stateless and separate the concerns of client and server.

1. **What is Agile Development? This will take some independent research.**

**Agile** software **development** is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for **Agile** Software **Development** and the 12 Principles behind it. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

1. **What is unit testing?**

Unit testing is supplying fake or mock implementations of framework components to simulate any scenario, using a variety of testing and mocking strategies, which can involve individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, or operating procedures.

1. **Describe URL routing. As part of your answer, discuss the concept of “clean URLs.”**

Uniform Resource Locator (URL) Routing is the process of intercepting an incoming Web request and automatically redirecting it to a different URL. Clean URLs, are uniform URLs intended to improve the usability and accessibility of a website or web service by being immediately and intuitively meaningful to non-expert users.

1. **Chapter 1 does not discuss the MVC pattern specifically, and in a sense this entire course is an extended examination of the MVC pattern. Using an independent resource (such as Wikipedia) briefly state the responsibility of (a) the model, (b) the controller, and (c) the view.**
2. The model represents the state of a particular aspect of the application.
3. The controller handles interactions and updates the model to reflect a change in state of the application, and then passes information to the view.
4. The view accepts necessary information from the controller and renders a user interface to display that information.