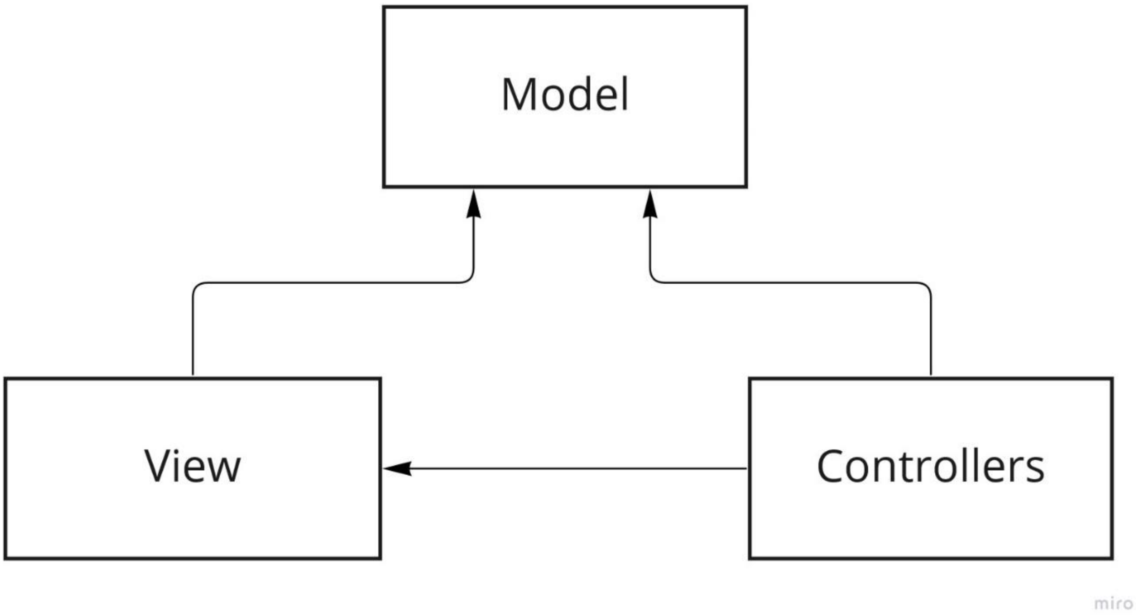
1 :- What is model?

Ans :- A model is a class with . cs (for C#) as an extension having both properties and methods. Models are used to set or get the data. If your application does not have data, then there is no need for a model.

2 :- Explain working of MVC Pattern.

Ans :- The Model-View-Controller is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application.



3 :- what is routing in MVC?

Ans :- Routing is the process through which the application matches an incoming URL path and executes the corresponding action methods. ASP.NET Core MVC uses a routing middleware to match the URLs of incoming requests and map them to specific action methods.

4 :- What is a View?

Ans :- A view is an HTML template with embedded Razor markup. Razor markup is code that interacts with HTML markup to produce a webpage that's sent to the client. In ASP.NET Core MVC, views are .cshtml files that use the C# programming language in Razor markup.

5 :- What is the importance of Controller?

Ans :- Controllers are the brain of an ASP.NET Core application.

Controllers are stored inside the Controllers folder in the root of the app. They are basically C# classes whose Public methods are called as Action Methods. These Action Methods handle the HTTP requests and prepare the response to be sent to the clients.

6 :- What is difference between MVC and Web Forms?

Ans :- Asp.Net MVC has Partial Views for code re-usability. Asp.Net Web Form has built-in data controls and best for rapid development with powerful data access.

7 :- Explain what is Model.

Ans :- The model binding system: Retrieves data from various sources such as route data, form fields, and query strings. Provides the data to controllers and Razor pages in method parameters and public properties. Converts string data to . NET types.

8 :- How is Model different from DataLayer of WebForm?

Ans :-

9 :- Explain the concept of MVC Scaffolding?

Ans :- Scaffolding is a technique used by many MVC frameworks like ASP.NET MVC, Ruby on Rails, Cake PHP and Node. JS etc., to generate code for basic CRUD (create, read, update, and delete) operations against your database effectively. Further you can edit or customize this auto generated code according to your need.

10 :- What is RouteData and how to access its values?

Ans :- RouteData is a property of the base Controller class, so RouteData can be accessed in any controller. RouteData contains route information of a current request. You can get the controller, action or parameter information using RouteData as shown below

12 :- Differences between Razor and ASPX View Engine in MVC?

Ans :- The Razor View Engine is a bit slower than the ASPX View Engine. Razor provides a new view engine with streamlined code for focused templating. Razor's syntax is very compact and improves readability of the markup and code. By default MVC supports ASPX (web forms) and Razor View Engine.

13 :- What is Razor in MVC?

Ans :- Razor is a markup syntax that lets you embed server-based code into web pages using C# and VB.Net. It is not a programming language. It is a server side markup language. Razor has no ties to ASP.NET MVC because Razor is a general-purpose templating engine.

14 :- Why is Layout page important?

Ans :- An effective layout not only looks attractive, but also helps the viewer understand the message the design is conveying. In other words, understanding layout is key when it comes to creating user-friendly, engaging designs, particularly in the realms of web design and advertising.

19 :- What are sections?

Ans :- Sections provide a way to organize where certain page elements should be placed. Each call to RenderSection can specify whether that section is required or optional: HTML Copy. <script type="text/javascript" src="~/scripts/global.js"></script> @RenderSection("Scripts", required: false)