1. Explain MVC (Model-View-Controller) in general?

Model-View-Controller (MVC) is an architectural software pattern that basically decouples various components of a web application. By using the MVC pattern, we can develop applications that are more flexible to changes without affecting the other components of our application.

- "Model", is basically domain data.
- "View" is the user interface to render the domain data.
- "Controller" translates user actions into appropriate operations performed on the model.

2. What is ASP.NET MVC?

ASP.NET MVC is a web development framework from Microsoft that is based on the Model-View-Controller (MVC) architectural design pattern. Microsoft has streamlined the development of MVC based applications using the ASP.NET MVC Framework.

3. What are the differences between ASP.NET MVC and ASP.NET WebForms?

ASP.NET Web Forms use the Page Controller Pattern for rendering layout, whereas ASP.NET MVC uses the Front Controller approach. In the case of the Page Controller Pattern, every page has its own Controller, in other words a code-behind file that processes the request. On the other hand, in ASP.NET MVC, a common Controller for all pages processes the requests.

Follow the link for the difference between the ASP.NET MVC and ASP.NET WebForms.

4. What are the Core features of ASP.NET MVC?

The core features of the ASP.NET MVC Framework are:

- Clear separation of application concerns (Presentation and Business Logic)
- An extensible and pluggable framework
- Extensive support for ASP.NET Routing
- Support for existing ASP.NET features

Follow for detailed understanding of above mentioned core features.

5. Can you please explain the request flow in the ASP.NET MVC framework?

The request flow for the ASP.NET MVC Framework is as follows.

The request hits the Controller coming from the client. The Controller plays its role and decides which model to use to serve the request, further passing that model to the View that then transforms the model and generates an appropriate response that is rendered to the client.

6. What is Routing in ASP.NET MVC?

In the case of a typical ASP.NET application, incoming requests are mapped to physical files such as an .aspx file. The ASP.NET MVC Framework uses friendly URLs that more easily describe the user's action but are not mapped to physical files.

The ASP.NET MVC Framework uses a routing engine that maps URLs to Controller classes. We can define routing rules for the engine, so that it can map incoming request URLs to the appropriate Controller.

Practically, when a user types a URL in a browser window for an ASP.NET MVC application and presses the "go" button, the routing engine uses routing rules defined in the Global.asax file to parse the URL and determine the path of the corresponding Controller.

7. What is the difference among ViewData, ViewBag and TempData?

To pass data from a Controller to a View and in a subsequent request, The ASP.NET MVC Framework provides various options, in other words ViewData, ViewBag and TempData.

Both ViewBag and ViewData are used to to communicate between Controller and corresponding View. But this communication is only for the server call, it becomes null if the redirect occurs. So, in short, it's a mechanism to maintain state between Controller and corresponding View.

ViewData is a dictionary object while ViewBag is a dynamic property (a new C# 4.0 feature). Since viewData is a dictionary object it is accessible using strings as keys and also requires typecasting for complex types. On the other hand, ViewBag doesn't have typecasting and null checks.

TempData is also a dictionary object that persists for the life of a HTTP Request. So, Tempdata can be used to maintain data between redirects, in other words from one Controller to another Controller.

8. What are Action Methods in ASP.NET MVC?

As previously stated about request flows in the ASP.NET MVC Framework, a request coming from a client hits the Controller first. Actually the MVC application determines the corresponding Controller using routing rules defined in Global.asax. And Controllers have specific methods for each user actions. Each request coming to the Controller is for a specific Action Method. The following code example, "ShowBooks" is an example of an Action Method.

```
public ViewResult ShowBooks(int id)
{
   var computerBook = db.Books.Where(p => P.BookID == id).First();
   return View(computerBook);
}
```

9. Explain the role of Model in ASP.NET MVC?

One of the core features of ASP.NET MVC is that it separates the input and UI logic from business logic. The role of the Model in ASP.NET MVC is to contain all application logic including validation, business and data access logic except View. In other words input and Controller; in other words UI logic.

The Model is normally responsible for accessing data from some persistent medium such as a database and manipulates it.

10. What are Action Filters in ASP.NET MVC?

If we need to apply some specific logic before or after action methods then we use action filters. We can apply these action filters to a Controller or a specific Controller action. Action filters are basically custom classes that provide a way for adding pre-action or post-action behavior to Controller actions.

For example:

- An Authorize filter can be used to restrict access to a specific user or a role.
- An OutputCache filter can cache the output of a Controller action for a specific duration.

11. Explain about 'page lifecycle' of ASP.NET MVC ?

The page lifecycle of an ASP.NET MVC page is explained as follows:

1. **App Initialisation**
In this stage, the aplication starts up by running Global.asax's Application_Start() method.
In this method, you can add Route objects to the static RouteTable.Routes collection.
If you're implementing a custom IControllerFactory, you can set this as the active controller factory by assigning it to the System.Web.Mvc.ControllerFactory.Instance property.

2. **Routing**
Routing is a stand-alone component that matches incoming requests to IHttpHandlers by URL pattern.
MvcHandler is, itself, an IHttpHandler, which acts as a kind of proxy to other IHttpHandlers configured in the Routes table.

3. **Instantiate and Execute Controller**
At this stage, the active IControllerFactory supplies an IController instance.

4. **Locate and invoke controller action**
At this stage, the controller invokes its relevant action method, which after further processing, calls RenderView().

5. **Instantiate and render view**
At this stage, the IViewFactory supplies an IView, which pushes response data to the IHttpResponse object.

What are the 3 main components of an ASP.NET MVC application?
1. M - Model
2. V - View
3. C - Controller

In which assembly is the MVC framework defined?
System.Web.Mvc

Is it possible to combine ASP.NET webforms and ASP.MVC and develop a single web application?
Yes, it is possible to combine ASP.NET webforms and ASP.MVC and develop a single web application.

What does Model, View and Controller represent in an MVC application?
Model: Model represents the application data domain. In short the applications business logic is contained with in the model.

View: Views represent the user interface, with which the end users interact. In short the all the user interface logic is contained with in the UI.

Controller: Controller is the component that responds to user actions. Based on the user actions, the respective controller, work with the model, and selects a view to render that displays the user interface. The user input logic is contained with in the controller.

What is the greatest advantage of using asp.net mvc over asp.net webforms?
It is difficult to unit test UI with webforms, where views in mvc can be very easily unit tested.

Which approach provides better support for test driven development - ASP.NET MVC or ASP.NET Webforms?
ASP.NET MVC

What are the advantages of ASP.NET MVC?
1. Extensive support for TDD. With asp.net MVC, views can also be very easily unit tested.
2. Complex applications can be easily managed
3. Seperation of concerns. Different aspects of the application can be divided into Model, View and Controller.
4. ASP.NET MVC views are light weight, as they donot use viewstate.

Is it possible to unit test an MVC application without running the controllers in an ASP.NET process?
Yes, all the features in an asp.net MVC application are interface based and hence mocking is much easier. So, we don't have to run the controllers in an ASP.NET process for unit testing.

Is it possible to share a view across multiple controllers?
Yes, put the view into the shared folder. This will automatically make the view available across multiple controllers.

What is the role of a controller in an MVC application?
The controller responds to user interactions, with the application, by selecting the action method to execute and alse selecting the view to render.

Where are the routing rules defined in an asp.net MVC application?
In Application_Start event in Global.asax

Name a few different return types of a controller action method?
The following are just a few return types of a controller action method. In general an action method can return an instance of a any class that derives from ActionResult class.
1. ViewResult
2. JavaScriptResult
3. RedirectResult
4. ContentResult
5. JsonResult

What is the significance of NonActionAttribute?
In general, all public methods of a controller class are treated as action methods. If you want prevent this default behaviour, just decorate the public method with NonActionAttribute.

What is the significance of ASP.NET routing?
ASP.NET MVC uses ASP.NET routing, to map incoming browser requests to controller action methods. ASP.NET Routing makes use of route table. Route table is created when your web application first starts. The route table is present in the Global.asax file.

What are the 3 segments of the default route, that is present in an ASP.NET MVC application?
1st Segment - Controller Name
2nd Segment - Action Method Name

3rd Segment - Parameter that is passed to the action method

Example: http://pragimtech.com/Customer/Details/5
Controller Name = Customer
Action Method Name = Details
Parameter Id = 5

ASP.NET MVC application, makes use of settings at 2 places for routing to work correctly. What are these 2 places?
1. Web.Config File : ASP.NET routing has to be enabled here.
2. Global.asax File : The Route table is created in the application Start event handler, of the Global.asax file.

What is the adavantage of using ASP.NET routing?
In an ASP.NET web application that does not make use of routing, an incoming browser request should map to a physical file. If the file does not exist, we get page not found error.

An ASP.NET web application that does make use of routing, makes use of URLs that do not have to map to specific files in a Web site. Because the URL does not have to map to a file, you can use URLs that are descriptive of the user's action and therefore are more easily understood by users.

What are the 3 things that are needed to specify a route?
1. URL Pattern - You can include placeholders in a URL pattern so that variable data can be passed to the request handler without requiring a query string.
2. Handler - The handler can be a physical file such as an .aspx file or a controller class.
3. Name for the Route - Name is optional.

Is the following route definition a valid route definition?
{controller}{action}/{id}
No, the above definition is not a valid route definition, because there is no literal value or delimiter between the placeholders. Therefore, routing cannot determine where to separate the value for the controller placeholder from the value for the action placeholder.

What is the use of the following default route?
{resource}.axd/{*pathInfo}
This route definition, prevent requests for the Web resource files such as WebResource.axd or ScriptResource.axd from being passed to a controller.

What is the difference between adding routes, to a webforms application and to an mvc application?
To add routes to a webforms application, we use MapPageRoute() method of the RouteCollection class, where as to add routes to an MVC application we use MapRoute() method.

How do you handle variable number of segments in a route definition?
Use a route with a catch-all parameter. An example is shown below. * is referred to as catch-all parameter.
controller/{action}/{*parametervalues}

What are the 2 ways of adding constraints to a route?
1. Use regular expressions
2. Use an object that implements IRouteConstraint interface

Give 2 examples for scenarios when routing is not applied?
1. A Physical File is Found that Matches the URL Pattern - This default behaviour can be overriden by setting the RouteExistingFiles property of the RouteCollection object to true.
2. Routing Is Explicitly Disabled for a URL Pattern - Use the RouteCollection.Ignore() method to prevent routing from handling certain requests.

What is the use of action filters in an MVC application?
Action Filters allow us to add pre-action and post-action behavior to controller action methods.

If I have multiple filters impleted, what is the order in which these filters get executed?
1. Authorization filters
2. Action filters
3. Response filters
4. Exception filters

What are the different types of filters, in an asp.net mvc application?
1. Authorization filters
2. Action filters
3. Result filters
4. Exception filters

Give an example for Authorization filters in an asp.net mvc application?
1. RequireHttpsAttribute
2. AuthorizeAttribute

Which filter executes first in an asp.net mvc application?
Authorization filter

What are the levels at which filters can be applied in an asp.net mvc application?
1. Action Method
2. Controller
3. Application
[b]Is it possible to create a custom filter?[/b]
Yes

What filters are executed in the end?
Exception Filters

Is it possible to cancel filter execution?
Yes

What type of filter does OutputCacheAttribute class represents?
Result Filter

What are the 2 popular asp.net mvc view engines?
1. Razor
2. .aspx

What symbol would you use to denote, the start of a code block in razor views?
@

What symbol would you use to denote, the start of a code block in aspx views?
<%= %>

In razor syntax, what is the escape sequence character for @ symbol?
The escape sequence character for @ symbol, is another @ symbol

When using razor views, do you have to take any special steps to proctect your asp.net mvc application from cross site scripting (XSS) attacks?
No, by default content emitted using a @ block is automatically HTML encoded to protect from cross site scripting (XSS) attacks.

When using aspx view engine, to have a consistent look and feel, across all pages of the application, we can make use of asp.net master pages. What is asp.net master pages equivalent, when using razor views?
To have a consistent look and feel when using razor views, we can make use of layout pages. Layout pages, reside in the shared folder, and are named as _Layout.cshtml

What are sections?
Layout pages, can define sections, which can then be overriden by specific views making use of the layout. Defining and overriding sections is optional.

What are the file extensions for razor views?
1. .cshtml - If the programming lanugaue is C#
2. .vbhtml - If the programming lanugaue is VB

How do you specify comments using razor syntax?
Razor syntax makes use of @* to indicate the begining of a comment and *@ to indicate the end. An example is shown below.
@* This is a Comment *@