

Assessment: ASP.Net Core MVC

Duration: 30 Minutes

Max Marks: 25

1. Which of the following is responsible for handling user interactions in the MVC pattern?
 - a) Model
 - b) View
 - c) Controller
 - d) Service

2. In ASP.NET Core MVC, where are view files typically stored?
 - a) Models folder
 - b) Controllers folder
 - c) Views folder
 - d) Data folder

3. How does the Razor syntax differ from regular HTML in ASP.NET Core MVC views?
 - a) Razor syntax allows direct embedding of C# code within HTML.
 - b) Razor syntax uses JavaScript for client-side interactions.
 - c) Razor syntax replaces HTML tags with custom components.
 - d) Razor syntax cannot handle conditional statements or loops.

4. What is the primary purpose of a Model in ASP.NET Core MVC?
 - a) Displaying data to the user
 - b) Handling user interactions
 - c) Storing business logic and data
 - d) Defining the UI layout

5. Which HTTP methods are commonly used for handling GET requests in ASP.NET Core MVC?
 - a) GET and POST
 - b) PUT and DELETE

- c) GET and DELETE
- d) POST and PUT

6. What is the purpose of the Startup.cs/Program.cs file in an ASP.NET Core MVC application?

- a) It defines the routing rules.
- b) It configures routing, services and middleware.
- c) It defines the models used in the application.
- d) It handles database migrations.

7. What is the purpose of routing in ASP.NET Core MVC?

- a) Handling HTTP requests and mapping them to controller actions.
- b) Defining the appearance of the UI.
- c) Storing user data in a database.
- d) Configuring authentication and authorization.

8. What is the difference between ViewData and ViewBag in ASP.NET Core MVC?

- a) ViewData is for passing data from the controller to the view, while ViewBag is for the reverse.
- b) ViewBag is for passing data from the controller to the view, while ViewData is for the reverse.
- c) ViewData and ViewBag are the same and can be used interchangeably for passing data from the controller to the view.
- d) ViewData and ViewBag are both used for authentication purposes.

9. What is the purpose of the TempData dictionary in ASP.NET Core MVC?

- a) It's used to store sensitive user data temporarily.
- b) It's used to share data between different users.
- c) It's used to persist data across different HTTP requests.
- d) It's used to store session-specific information.

10. Which attribute can be used to specify that a parameter should be bound from the query string in an MVC controller action?

- a) [FromBody]
- b) [FromHeader]

- c) [FromQuery]
- d) [FromRoute]

11. What is the purpose of a ViewModel in ASP.NET Core MVC?

- a) It is used for displaying data to the user.
- b) It defines the structure of the database.
- c) It encapsulates the data to be used in a view.
- d) It handles user input validation.

12. How does Dependency Injection benefit an ASP.NET Core MVC application?

- a) It reduces the number of required controller actions.
- b) It improves application performance.
- c) It enhances code maintainability and testability.
- d) It automates database migrations.

13. What is Tag Helper in ASP.NET Core MVC?

- a) It's a way to include JavaScript files in the views.
- b) It's a component that simplifies generating HTML elements in Razor views.
- c) It's a built-in security feature for preventing Cross-Site Scripting (XSS) attacks.
- d) It's a type of authentication middleware.

14. How can you enable attribute routing in ASP.NET Core MVC?

- a) It's enabled by default; no action is required.
- b) By adding a "routes" section to the appsettings.json file.
- c) By decorating the controller or action with [Route] attributes.
- d) By installing a third-party NuGet package.

15. What is the purpose of an Authorize attribute in ASP.NET Core MVC?

- a) It's used to validate user inputs in forms.
- b) It's used to manage database connections.
- c) It's used to add authentication to controller actions.

d) It's used to perform cross-cutting concerns like logging and authorization.

16. What is the purpose of the [ValidateAntiForgeryToken] attribute in ASP.NET Core MVC?

- a) It prevents Cross-Site Scripting (XSS) attacks.
- b) It enforces HTTPS communication for sensitive data.
- c) It validates the anti-forgery token to prevent cross-site request forgery (CSRF) attacks.
- d) It ensures proper authorization for sensitive controller actions.

17. In ASP.NET Core MVC, what is the purpose of the IActionResult interface?

- a) It defines the structure of the application's main layout.
- b) It represents the model passed from the controller to the view.
- c) It provides a common base type for all action results returned by controller actions.
- d) It is used for handling exceptions in the application.

18. What is the role of the UseExceptionHandler middleware in ASP.NET Core MVC?

- a) It handles errors and exceptions in the application.
- b) It configures the routing for the application.
- c) It defines the layout of the application's user interface.
- d) It manages database migrations automatically.

19. Which attribute is commonly used to specify the HTTP method for an action to receive data from html form in an ASP.NET Core MVC controller?

- a) [HttpPut]
- b) [HttpPost]
- c) [HttpGet]
- d) [HttpDelete]

20. Consider the following model class:

```
public class Product
{
    public int Id { get; set; }
```

```
public string Name { get; set; }  
  
public decimal Price { get; set; }  
  
}
```

Which tag helper in a Razor view would you use to generate a form input for editing the "Name" property of a Product instance?

- a) <input asp-for="Product.Name" />
- b) <input asp-model="Product.Name" />
- c) <input asp-field="Product.Name" />
- d) <input for="Product.Name" />

21. In an ASP.NET Core MVC view, how would you render the value of a model property Title within a paragraph element?

- a) <p>@Model.Title</p>
- b) <p><%= Model.Title %></p>
- c) <p><@Model.Title></p>
- d) <p>Model.Title</p>

22. You want to pass data from an ASP.NET Core MVC controller action to a view. Which of the following approaches is suitable when using a strongly typed view?

- a) Using ViewData.
- b) Using TempData.
- c) Using ViewBag.
- d) Using the @model directive in the view.

23. In an ASP.NET Core MVC application, how can you configure dependency injection for a custom service?

- a) By directly instantiating the service within the controller.
- b) By using the the controller constructor.
- c) By adding the service to the services collection in the Startup.cs/Program.cs file.
- d) By configuring it in the appsettings.json file.

24. What is the purpose of model binding in ASP.NET Core MVC?

- a) It's used to create a connection between the model and the database.
- b) It's used to automatically generate HTML forms based on the model properties.
- c) It's used to convert HTTP request data into strongly typed model objects.
- d) It's used to define the structure of the view in the Razor template.

25. What does Inversion of Control (IoC) refer to in ASP.NET Core MVC?

- a) A design pattern that encapsulates data and methods into a single unit.
- b) The process of arranging classes and modules to form a coherent architecture.
- c) A way to achieve loose coupling by shifting the responsibility of object creation and management to a container.
- d) The use of dependency injection exclusively in the presentation layer of the application.