**Section - A**

Q1)  
  
a) What do you mean by Active Server Pages?  
b) What is a Multi tiered Design?  
  
0) What do you mean Distributed Application?  
(1) What do you mean by Cookies?  
  
e) What is a Web Service?  
  
f) What is SOAP?  
  
g) What do you mean by DataGrid Control?  
h) What is a Demo Application?  
  
i) What is a Dataset Object?  
  
j) What do you mean by Validation Controls? Give one example each.  
k) What is a Form?  
  
1) What is DataAdapter?  
  
m) What is .NET?  
  
11) What is Web Form?  
  
0) What do you mean by Synchronization?

**Section - B**

(9 X 5 = 45)  
  
Q2) Discuss at least five HTML Controls.  
  
Q3) Write a short note on State Management.  
  
Q4) Discuss the advantages of using .NET over other parallel technologies.  
Q5) Write a program in ASP .NET to find server’s Date and Time.  
  
Q6) What is purpose of using Session Objects?  
  
Q7) Discuss the design of a Credit Card Validator Service?  
  
Q8) Discuss at least three Data Bound controls.  
  
Q9) Design a web form for a Guessing Game of your choice.  
  
Q1 0) Discuss in detail Distributed Application Architecture.  
  
Q11) Discuss at least three request objects in ASP.NET.  
  
Q12) Write a program in ASP.NET to demonstrate use of CheckBox in a form.  
  
Q13) Write a program in ASP.NET to demonstrate use of Buttons in a Webform.

**Important Questions about ASP.NET with Answers-**

**Ques 1 What is ASP.NET?**  
**Ans-** ASP.NET is the new offering for web developer from the Microsoft. It is not simply the next generation of ASP (Active Server Page).It is completely re-engineered and enhanced technology. It provides large set of user controls, XML-based components and integrated user authentication.

**Ques 2 What are advanced features of ASP.NET ?**  
**Ans-** Listed below are the some of the advanced features of ASP.NET-  
1) Support for compiled language.  
2) Services provided by .NET framework  
3) Graphical development environment  
4) State management  
5) Update files while server is running  
6) XML based configuration files

ASP.NET is most demanding

**Ques 3 What is “Atlas”?**  
**Ans-** ASP.NET AJAX was previously known by Atlas which is an implementation of Microsoft based on AJAX framework.

**Ques 4 What are web controls in ASP.NET?**  
**Ans-** Web controls are special ASP.NET tags understood by the server. Web server controls are created on the server and they require a run at=”server” attribute to work.

**Ques 5 Write the syntax for creating web controls?**  
**Ans-** The syntax for creating a web control is written as follows-  
<asp:control\_name id=”some\_id” run at=”server” />

**Ques 6 Give some examples of web controls ?**  
**Ans-** Following are the some of the examples of the Web Controls-  
1) AdRotator  
2) Button  
3) Calender  
4) CheckBoxList  
5) DropDownList  
6) HyperLink  
7) Image  
8) Label  
9) PlaceHolder  
10) TableRow  
11)RadioButtonList  
12) TableCell  
13)TextBox  
14) XML

**Ques 7 What are three category of Web Controls?**  
**Ans-** Three categorization of web controls are-  
a) Text and graphics controls  
b) AdRotator controls  
c) Validation controls

**Ques 8 What is session tracking?**  
**Ans-** Session Tracking is a concept which allows you to maintain a relation between two successive requests made to a server on the internet .As we know HTTP is stateless protocol ,so never keep record of request made to it.Session tracking allow you to store the information would be checked every time you do any thing within your inbox. Thus you would not be asked to enter your password with every click.

**Ques 9 What are the solution for the arise by fact that HTTP being stateless?**  
**Ans-** There are three typical solution to this problem –  
a) cookies  
b) URL rewriting  
c) Hidden form fields

**Ques 10 Is ASP.NET support any type of databases?**  
**Ans-** ASP.NET support for web services, oracle databases, SQL server databases with custom schema.

1. You are developing an ASP.NET web application that needs to store sensitive information securely, such as connection strings and API keys. What is the best way to achieve this?
2. Encrypt the sensitive information and store it in web.config.
3. Use the ViewState to store sensitive information.
4. Store sensitive information in a separate XML file outside the application root.
5. Utilize the SecureString class to store sensitive information.

1. You are developing an application, with an SLA of 99% availability. What will be your approach on the various aspects of the application development and configuration? Select all correct answers.
2. Deploy the application on high availability cloud servers
3. Apply CI/CD offered by Azure, or use similar technologies for seamless deployment, without downtime.
4. Make distributed processing decisions wisely using maximum CDNs possible
5. Development architecture should be based on Dependency Injection framework to accommodate changes with minimum efforts

1. You are tasked with optimizing the performance of an ASP.NET web application. Which technique would be most effective in reducing the number of round-trips between the client and the server?
2. Minification of JavaScript and CSS files.
3. Implementing AJAX for partial page updates.
4. Enabling output caching.
5. Bundling multiple script and style resources.
6. You need to implement authentication in an ASP.NET MVC application. Which authentication method provides the highest level of security?
7. Forms Authentication
8. Windows Authentication
9. OAuth Authentication
10. Token-based Authentication
11. In an ASP.NET Core application, you want to implement dependency injection. Which built-in container is used by default?
12. Unity
13. Ninject
14. Autofac
15. IServiceCollection
16. You are developing a RESTful API using ASP.NET Web API. How can you enable Cross-Origin Resource Sharing (CORS) to allow requests from a different domain?
17. Use the [EnableCors] attribute on the controller.
18. Configure CORS in the web.config file.
19. Set the Access-Control-Allow-Origin header in the response.
20. Install the CORS NuGet package.
21. You want to implement caching in an ASP.NET application to improve performance. Which caching option allows you to cache data based on a dependency, such as a file or database change?
22. Output Caching
23. Data Caching
24. Object Caching
25. Cache Dependency
26. You are developing a web application that requires real-time communication between the server and the client. Which technology is best suited for achieving real-time communication in an ASP.NET application?
27. SignalR
28. WebSockets
29. AJAX
30. Web API
31. You want to implement role-based authorization in an ASP.NET MVC application. Which attribute should you use to restrict access to a controller action based on the user's role?
32. [Authorize(Roles = "Admin")]
33. [AuthorizeIfRole("Admin")]
34. [AuthorizeAccess(Role = "Admin")]
35. [RoleAuthorize("Admin")]

1. You are developing a multi-tier ASP.NET application with a separate business logic layer. How can you ensure that your business logic layer is loosely coupled with the presentation layer?
2. Use Dependency Injection
3. Utilize Inversion of Control (IoC)
4. Implement the Singleton pattern
5. Use sealed classes for business logic
6. You are working on an ASP.NET MVC project, and you need to implement client-side validation using unobtrusive JavaScript. Which library is commonly used for unobtrusive validation in ASP.NET MVC?
7. jQuery Validate
8. AngularJS
9. React
10. KnockoutJS
11. You are developing an ASP.NET application that needs to support multiple languages. What is the correct way to implement localization and resource files for different languages?
12. Use satellite assemblies
13. Utilize the <globalization> element in web.config
14. Create separate folders for each language
15. Implement language-specific controllers
16. You are implementing a background task in an ASP.NET Core application that needs to run periodically. Which service should you use for this purpose?
17. IHostedService
18. IMiddleware
19. IActionFilter
20. IResultFilter
21. You want to implement URL routing in an ASP.NET MVC application. Which file is responsible for defining the routes in the application?
22. RouteConfig.cs
23. Global.asax
24. web.config
25. Routes.json
26. You are developing an ASP.NET Core MVC application, and you need to configure authentication using an external identity provider, such as Google or Facebook. Which authentication middleware should you use?
27. JwtBearerAuthentication
28. OpenIdConnectAuthentication
29. CookieAuthentication
30. OAuthBearerAuthentication
31. You are developing a RESTful API in ASP.NET Core. How can you version the API to ensure backward compatibility?
32. Use query parameters for versioning.
33. Include the version in the request headers.
34. Use versioning in the URL route.
35. Use content negotiation for versioning.
36. You are implementing a file upload feature in an ASP.NET MVC application. What is the recommended way to handle large file uploads efficiently?
37. Use AJAX for asynchronous file uploads.
38. Increase the maximum request length in web.config.
39. Use a third-party library, such as FineUploader.
40. Implement chunked file uploads.
41. You are working on an ASP.NET Core project, and you need to configure dependency injection for a service that should have a scoped lifetime. Which method should you use when registering the service in the Startup.cs file?
42. AddTransient
43. AddScoped
44. AddSingleton
45. AddScopedService
46. You are implementing authentication in an ASP.NET Core Web API. What is the purpose of the [AllowAnonymous] attribute on a controller action?
47. It allows anonymous access to the action.
48. It denies access to the action for all users.
49. It restricts access to authenticated users only.
50. It ignores authentication for the action.

|  |
| --- |
| 1. You are working on an ASP.NET MVC application. Explain how you would implement a global exception handler to log all unhandled exceptions and display a custom error page to the user.      1. Describe a situation where you would use asynchronous programming in an ASP.NET Core application, and provide an example of how you would implement it using C#. 2. Explain the concept of dependency injection in ASP.NET Core. Provide an example of how you would register and resolve a scoped service in the application. 3. Describe a scenario where you need to implement token-based authentication for a Web API in ASP.NET Core. Provide steps to validate and authorize requests using JWT (JSON Web Tokens). 4. Discuss different caching strategies in ASP.NET. Provide an example of how you would implement caching for a frequently accessed database query to improve performance. 5. How would you implement a secure file upload feature in an ASP.NET MVC application, ensuring that uploaded files are scanned for malware and stored in a secure manner? 6. Explain a scenario where you would use SignalR in an ASP.NET application to enable real-time communication between clients and the server. Provide a brief example of how you would implement a real-time feature using SignalR. 7. Describe common performance optimization techniques for Entity Framework in an ASP.NET application. Provide an example of how you would optimize a database query to minimize the number of database round-trips.   **Section – C (10X4=40 Marks)**  **(Attempt Any 10)**   1. You are tasked with implementing a secure user authentication and authorization system in an ASP.NET MVC application. Explain the steps you would take to ensure that only authenticated users can access certain parts of the application, and different roles have different levels of access.      1. Describe the best practices for handling exceptions in an ASP.NET application. How would you log and handle exceptions to ensure a smooth user experience and aid in debugging? 2. How would you prevent Cross-Site Scripting (XSS) attacks in an ASP.NET web application? Provide examples of measures you would take to ensure the security of user input and output. 3. Explain the benefits of asynchronous programming in ASP.NET. Provide examples of situations where asynchronous programming can significantly improve performance. 4. Discuss the importance of dependency injection in ASP.NET and how you would implement it. Provide an example of a scenario where dependency injection is beneficial. 5. You are tasked with building a RESTful Web API using ASP.NET Web API. Explain the key components and steps involved in creating a secure and scalable Web API.      1. You need to implement real-time communication in an ASP.NET application. Explain the options available for achieving real-time updates, and provide an example of when SignalR might be the preferred solution. 2. Your ASP.NET application is experiencing performance issues. Describe the steps you would take to identify and resolve bottlenecks, optimizing the application for better performance. 3. Your ASP.NET application is expected to experience a significant increase in traffic. Explain how you would ensure the scalability of the application, both in terms of vertical and horizontal scaling. 4. Discuss the use of WebSockets in ASP.NET. Provide an example of a scenario where WebSockets are preferable over traditional HTTP communication. 5. Explain the importance of unit testing in ASP.NET development. Describe how you would approach writing unit tests for a controller in an ASP.NET MVC application. |
|  |