**Important Viva Questions and Answers**

**Module 1: Introduction to ASP.NET**

1. **What is ASP.NET?**
   * ASP.NET is a server-side web application framework developed by Microsoft to build dynamic web pages, services, and applications.
2. **What is PostBack in ASP.NET?**
   * PostBack is the process where a webpage posts data back to the same page on the server.
3. **What are Server-side controls?**
   * Controls like TextBox, Button, etc., processed on the server. Their state is preserved between requests.
4. **What is AutoPostBack?**
   * When set to true, it automatically sends the page back to the server on control value change.

**Module 2: UI Design with Controls and Master Page**

1. **What is a Master Page?**
   * A Master Page allows creating a consistent layout for multiple pages in a web application.
2. **What are validation controls in ASP.NET?**
   * Controls like RequiredFieldValidator, RangeValidator, etc., used to validate user input.
3. **What is the use of AdRotator control?**
   * It displays a sequence of ads/images based on an XML file or database.

**Module 3: Database Programming with ADO.NET**

1. **What is ADO.NET?**
   * It is a set of classes that expose data access services to .NET programmers.
2. **Difference between connected and disconnected architecture?**
   * Connected: Uses SqlDataReader; Disconnected: Uses DataSet, DataAdapter.
3. **Name any two data-bound controls.**
   * GridView, Repeater, DetailsView.
4. **How to call a stored procedure in ASP.NET?**
   * Use SqlCommand.CommandType = CommandType.StoredProcedure and execute it.

**Module 4: State Management and AJAX**

1. **What is State Management?**
   * Technique to maintain state (data) across multiple HTTP requests.
2. **Types of State Management?**
   * Client-side: ViewState, Cookies; Server-side: Session, Application, Cache.
3. **What is AJAX in ASP.NET?**
   * AJAX allows partial page updates without full page reload using asynchronous requests.
4. **What is ViewState?**
   * Mechanism to preserve page and control values between postbacks.

**Module 5: Web Services and WCF**

1. **What is a Web Service?**
   * A web-based application that provides data or services to other applications over the internet.
2. **Difference between Web Service and WCF?**
   * WCF is a unified framework to build service-oriented applications, supporting more protocols than basic Web Services (like TCP, Named Pipes).
3. **What is SOAP?**
   * SOAP (Simple Object Access Protocol) is an XML-based protocol for exchanging structured data.

**Module 6: ASP.NET Core MVC**

1. **What is MVC architecture?**
   * Model–View–Controller is a design pattern that separates the application logic into three interconnected components.
2. **What are Razor Views?**
   * Views in ASP.NET Core that use Razor syntax for mixing C# code with HTML.
3. **What is Entity Framework?**
   * An ORM framework for .NET that allows developers to work with databases using .NET objects.
4. **Explain CRUD in MVC.**
   * CRUD stands for Create, Read, Update, Delete — the four basic operations on persistent storage using MVC and EF.

**✅ Experiment Explanations**

**1. Registration Form with Advanced Controls**

**Objective**: Create a web application for an organization with forms using advanced controls.  
**Includes**:

* Controls like TextBox, DropDownList, Calendar, FileUpload, etc.
* Server-side validation
* User-friendly layout with CSS or themes

**2. Website Using Master Page**

**Objective**: Demonstrate the use of **Master Pages** for consistent layout.  
**Includes**:

* One master page with header, footer, and menu
* Multiple content pages that change dynamically while keeping layout fixed

**3. Connection-Oriented Architecture**

**Objective**: Show how to use **connected mode** with ADO.NET (e.g., SqlConnection, SqlCommand, SqlDataReader).  
**Includes**:

* Display data directly from the database
* Keep the connection open during data processing

**4. Disconnected Architecture**

**Objective**: Use **disconnected mode** (e.g., DataSet, DataAdapter).  
**Includes**:

* Retrieve data, disconnect from DB
* Modify/display data without reconnecting to the database

**5. Data-Bound Controls**

**Objective**: Bind server-side data to UI controls.  
**Includes**:

* Use controls like GridView, ListView, Repeater
* Display database table contents dynamically

**6. Simple Stored Procedure**

**Objective**: Create and use a stored procedure that returns data.  
**Includes**:

* Write a SQL stored procedure
* Call it from ASP.NET using ADO.NET
* Show results in GridView or DetailsView

**7. Parameterized Stored Procedure**

**Objective**: Use a stored procedure with input parameters.  
**Includes**:

* Procedure with WHERE clause
* Take user input (like EmployeeID)
* Display corresponding records

**8. Using LINQ**

**Objective**: Demonstrate **LINQ to SQL or Objects** in ASP.NET.  
**Includes**:

* Write queries in C# using LINQ syntax
* Fetch and bind results to UI controls

**9. Entity Framework in .NET**

**Objective**: Use **Entity Framework** (EF) to access the database.  
**Includes**:

* Code-First or Database-First model
* Perform operations like read, insert, update

**10. Client-Side Session Management**

**Objective**: Demonstrate state management using **client-side techniques**.  
**Includes**:

* Use Cookies, QueryString, HiddenField, or ViewState
* Show how data persists across pages without server memory

**11. Server-Side Session Management**

**Objective**: Use server memory for state management.  
**Includes**:

* Use Session, Application, or Cache
* Store and retrieve user-specific information across pages

**12. AJAX Controls**

**Objective**: Use AJAX for asynchronous, partial-page updates.  
**Includes**:

* Use controls like UpdatePanel, ScriptManager
* No full-page postback on control update (e.g., auto-search)

**13. Web Service Consumption**

**Objective**: Create and consume a **Web Service** in ASP.NET.  
**Includes**:

* Define a WebMethod
* Consume it in a web app to fetch/display data

**14. MVC Framework Application**

**Objective**: Build a basic app using **ASP.NET Core MVC**.  
**Includes**:

* Use Models, Views (Razor), and Controllers
* Implement navigation using Routing

**15. MVC Application with Entity Framework**

**Objective**: Integrate **EF** into an MVC app.  
**Includes**:

* Use DbContext
* Query data using LINQ
* Bind data to views

**16. MVC Application with CRUD**

**Objective**: Perform full **Create, Read, Update, Delete** operations.  
**Includes**:

* HTML forms with Razor
* Use HttpGet and HttpPost
* Use Entity Framework for database interaction