1. What is ADO.NET and advantages of ADO.NET
    - ADO.NET, or ActiveX Data Objects for .NET, is a data access technology in the Microsoft .NET Framework. It provides a set of libraries for accessing and managing data from various sources, such as databases or XML files, in a .NET application.
    - **Interoperability**: It supports multiple data sources, ensuring compatibility with various databases and data formats.
    - **Data Binding**: ADO.NET facilitates seamless data binding to UI controls, simplifying the development of data-driven applications.
    - **Security**: It offers robust security features, including parameterized queries, to prevent SQL injection attacks and enhance data protection.

2. What is DataSet in ADO.NET?
    - A DataSet in ADO.NET is an in-memory representation of data retrieved from a database. It acts as a disconnected, cache-like structure that can store tables, relationships, and constraints. It enables offline manipulation of data, allowing developers to work with data in a disconnected state from the database. The DataSet class provides methods for data manipulation, and it is a key component for building data-driven applications in .NET.

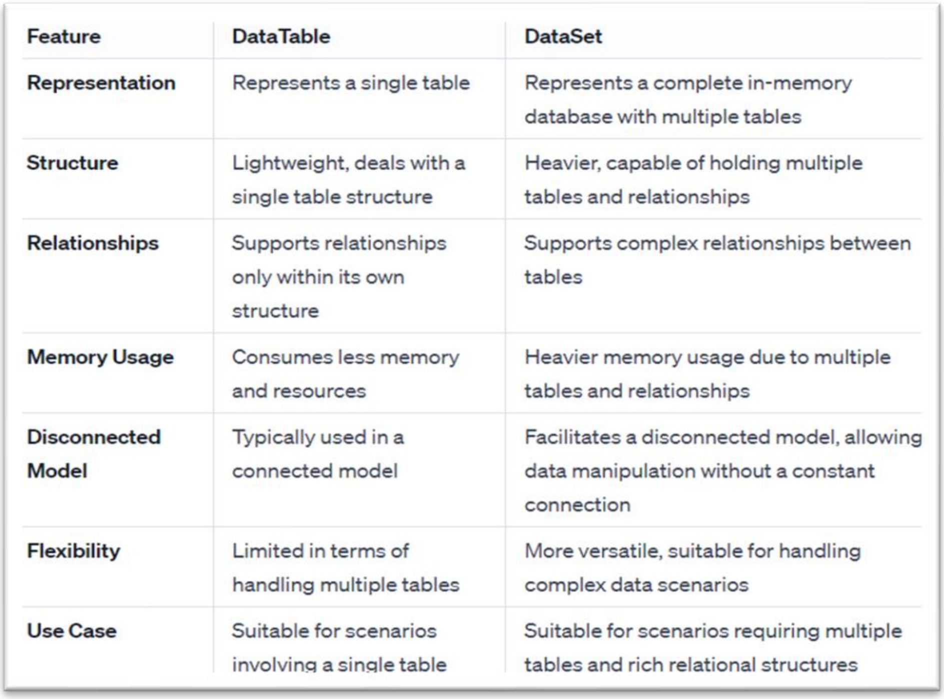
3. What is a DataAdapter in ADO.NET?
    - A DataAdapter in ADO.NET acts as a bridge between a DataSet and a data source, typically a database. It facilitates the exchange of data by filling a DataSet with data from the data source and updating the data source with changes made to the DataSet. The DataAdapter uses a set of commands (SelectCommand, InsertCommand, UpdateCommand, DeleteCommand) to interact with the data source. Helps in fetching and updating data, making it easier for developers to work with datasets in a disconnected manner.

4. Explain about DataSet types in ADO.NET.
    - In ADO.NET, there are two main types of DataSets: Typed DataSet and Untyped DataSet.

      Typed DataSet: Strongly Typed: Typed DataSet is generated at design time, providing a strongly-typed representation of the data schema. Code Generation: The structure of a Typed DataSet is defined through a .xsd (XML Schema Definition) file, and code is generated for each table and column.

      Untyped DataSet: Loosely Typed: Untyped DataSet is dynamically created at runtime and lacks the compile-time checking provided by a Typed DataSet. Flexible Structure: The structure is not predefined, making it more flexible but potentially prone to runtime errors.

5. Compare DataTable and DataSet.



Feature	DataTable	DataSet
Representation	Represents a single table	Represents a complete in-memory database with multiple tables
Structure	Lightweight, deals with a single table structure	Heavier, capable of holding multiple tables and relationships
Relationships	Supports relationships only within its own structure	Supports complex relationships between tables
Memory Usage	Consumes less memory and resources	Heavier memory usage due to multiple tables and relationships
Disconnected Model	Typically used in a connected model	Facilitates a disconnected model, allowing data manipulation without a constant connection
Flexibility	Limited in terms of handling multiple tables	More versatile, suitable for handling complex data scenarios
Use Case	Suitable for scenarios involving a single table	Suitable for scenarios requiring multiple tables and rich relational structures

6. What are the different namespaces available in ADO.NET?
   - In ADO.NET, there are several namespaces that provide classes and interfaces for working with data access. Some of the key namespaces include:
     System.Data: The core namespace for ADO.NET, containing fundamental classes like DataSet, DataTable, DataRow, DataColumn, and others.
     System.Data.SqlClient: Specific to Microsoft SQL Server, this namespace includes classes for connecting to and interacting with SQL Server databases.
     System.Data.OleDb: Provides classes for accessing data sources using OLE DB (Object Linking and Embedding, Database).
     System.Data.Odbc: Allows accessing data sources using ODBC (Open Database Connectivity).

7. Why choose Bootstrap for building websites?
   - **Responsive Design:** Bootstrap provides a responsive grid system and predefined classes for creating responsive and mobile-friendly layouts, ensuring a consistent user experience across devices.
   - **Cross-Browser Compatibility:** Bootstrap is designed to be compatible with major web browsers, ensuring consistent appearance and functionality across different platforms.
   - **Consistent Design Patterns:** Bootstrap follows a consistent design language, which helps maintain a uniform and professional appearance throughout the website.
   - **Built-in Components:** Bootstrap includes a variety of pre-built components such as navigation bars, modals, carousels, and more. This reduces the need for custom coding and accelerates development.

8. List the components of Bootstrap.
    - **Tables**: Styled tables with options for striped rows, bordered tables, and hover effects.
    - **Forms**: Pre-styled form controls, form layouts, and validation styles to enhance the look and feel of user input forms.
    - **Buttons**: Versatile button styles, sizes, and states for creating interactive user interfaces.
    - **Images**: Responsive image classes and utilities for controlling image sizes and alignments.
    - **Alerts**: Dismissible and styled alert messages for conveying information or feedback to users.
    - **Navbar**: Responsive navigation bar with various customization options for creating intuitive navigation menus.
    - **Dropdowns**: Create dropdown menus with various styles and options for navigation or user actions.

9. Explain in brief about a) View State b) Session State
    - **View State:**

      Purpose: View State is a client-side state management technique in ASP.NET that allows the preservation of the state of a page and its controls across postbacks.

      Storage: The state information is serialized and stored in a hidden field on the page, ensuring that it is sent back and forth between the server and the client with each request.

      Scope: Limited to a single page and is used to retain the values of controls and other page-specific data.
    - b) Session State:

      Purpose: Session State is a server-side state management technique in ASP.NET that allows the storage of user-specific information throughout the user's session.

      Storage: Information is stored on the server, and a session identifier is sent to the client, which is used to retrieve the session data on subsequent requests.

      Scope: Extends across multiple pages within the same application and is useful for maintaining user-specific data, such as user preferences or login information.

10. Explain in brief about a) Cookies State b) Application State
    - **Cookies State** is a client-side mechanism in web development that stores small pieces of data on a user's browser. This enables the persistence of information between the client and server across multiple requests. Cookies are often employed for user authentication, personalization, and tracking preferences. They have domain and path attributes, allowing control over their accessibility.
    - **Application State** is a server-side approach in ASP.NET for storing global data accessible to all users within an application. Information is stored in the server's memory throughout the application's lifespan, commonly used for shared configuration settings or counters. While Cookies offer individualized client storage, Application State provides a centralized server repository for data shared among all users.

11. What are web controls in ASP.NET?
    - In ASP.NET, web controls are components or objects that are used to create the user interface of a web application. These controls are part of the ASP.NET framework and provide a way to encapsulate and manage the rendering of HTML elements on a web page. Web controls offer a higher level of abstraction than traditional HTML controls, making it easier for developers to create dynamic and interactive web applications. There are two main types of web controls in ASP.NET:

      HTML Controls: These controls are an abstraction of standard HTML elements and include controls like TextBox, Button, CheckBox, and DropDownList.

      Web Server Controls: These controls are more advanced and provide a higher level of functionality. Examples include GridView, DataList, and Calendar.

12. What are the different validation controls in ASP.NET?
    - In ASP.NET, validation controls help ensure that user input is accurate and meets specified criteria before it is processed by the server. Some key validation controls include:

      RequiredFieldValidator: Ensures that a specified input field is not left blank.

      RangeValidator: Checks whether the input falls within a specified numeric or date range.

      RegularExpressionValidator: Validates input against a specified regular expression pattern, useful for complex data format validation.

      StringLengthValidator: Validates the length of a string, ensuring it falls within a specified range.

      ValidationSummary: Displays a summary of all validation errors on a page, providing a consolidated view for users.