ASP.NET Viva Questions - Short Answers

Practical 1: Registration forms with advanced controls

Q: What are the different types of form controls in ASP.NET? A: TextBox, DropDownList, CheckBox, RadioButton, RadioButtonList, CheckBoxList, Button, LinkButton, ImageButton, Label, Literal, FileUpload, Calendar, AdRotator.

Q: How do you validate user inputs on the client and server side? A: Client-side: JavaScript validation, ASP.NET validation controls with EnableClientScript=true. Server-side: ASP.NET validation controls, Page.IsValid property, custom validation in code-behind.

Q: What is the difference between TextBox, DropDownList, and RadioButtonList? A:

- TextBox: Single/multi-line text input
- DropDownList: Single selection from dropdown menu
- RadioButtonList: Single selection from multiple radio buttons

Practical 2: Master Page Concept

Q: What is a master page in ASP.NET? A: Template page defining common layout and structure for multiple pages, with .master extension containing ContentPlaceHolder controls.

Q: How does a content page relate to a master page? A: Content page inherits from master page using MasterPageFile attribute and provides content through Content controls that map to ContentPlaceHolders.

Q: Can you nest master pages? If yes, how? A: Yes, a master page can inherit from another master page using MasterPageFile attribute, creating nested master page hierarchy.

Practical 3: Connection-Oriented Architecture

Q: What is connection-oriented communication in ASP.NET? A: Database connection remains open throughout the operation, providing real-time data access with continuous connection to database.

Q: Explain how ADO.NET handles connection-oriented architecture. A: Uses SqlConnection to establish connection, SqlCommand to execute queries, SqlDataReader for forward-only data reading while connection stays open.

Q: What is the role of SqlConnection, SqlCommand? A: SqlConnection: Establishes database connection. SqlCommand: Executes SQL statements/stored procedures against the database.

Practical 4: Disconnected Architecture

Q: What is a disconnected architecture in ADO.NET? A: Data is retrieved, connection is closed, operations performed on cached data, then reconnected only when updating database.

Q: What are DataSet and DataAdapter? How do they work? A: DataSet: In-memory cache of data tables. DataAdapter: Bridge between DataSet and database, fills DataSet and updates database.

Q: How do you update a database using a DataSet? A: Modify DataSet data, then use DataAdapter.Update() method to push changes back to database.

Practical 5: Data-Bound Controls of ASP.NET

Q: What are data-bound controls in ASP.NET? A: Controls that automatically display data from data sources like GridView, Repeater, DataList, DropDownList, ListBox.

Q: Explain how GridView or Repeater works. A: GridView: Displays data in tabular format with built-in features. Repeater: Template-based control for custom data display layout.

Q: How do you bind data from a database to a control? A: Set DataSource property to data source, call DataBind() method, or use declarative data source controls like SqlDataSource.

Practical 6: Simple Stored Procedure

Q: What is a stored procedure? A: Pre-compiled SQL code block stored in database that can be executed with parameters.

Q: How do you execute a stored procedure in ASP.NET? A: Create SqlCommand with CommandType.StoredProcedure, set CommandText to procedure name, execute using ExecuteNonQuery/ExecuteReader.

Q: What are the advantages of using stored procedures? A: Better performance, security, code reusability, centralized business logic, reduced network traffic.

Practical 7: Parameterized Stored Procedure

Q: What is the need for parameterized stored procedures? A: Prevents SQL injection attacks, improves performance through query plan reuse, enables dynamic queries safely.

Q: How do you prevent SQL injection using parameters? A: Use SqlParameter objects instead of string concatenation, parameterized queries treat input as data, not executable code.

Q: Demonstrate how you pass parameters to a stored procedure from ASP.NET. A:

```
cmd.Parameters.Add("@ParamName", SqlDbType.VarChar).Value = "value";
```

Practical 8: Use of LINQ

Q: What is LINQ in .NET? A: Language Integrated Query - allows querying data using C# syntax instead of SQL.

Q: What are the types of LINQ? A: LINQ to Objects, LINQ to SQL, LINQ to XML, LINQ to Entities, LINQ to DataSet.

Q: Write a LINQ query to fetch all records from a list where age > 25. A: `var result = list.Where(x => x.Age > 25);`

Practical 9: Entity Framework

Q: What is Entity Framework? A: Object-Relational Mapping (ORM) framework that maps database tables to .NET objects.

Q: What are the different approaches in EF (Code First, Database First)? A: Code First: Create classes first, generate database. Database First: Generate classes from existing database. Model First: Design model, generate both.

Q: How is EF different from ADO.NET? A: EF is ORM with object-based queries, ADO.NET is low-level data access with SQL queries.

Practical 10: Client-Side Session Management

Q: What is client-side session management? A: Storing session data on client browser using cookies, hidden fields, or query strings.

Q: How is data stored in cookies? A: Small text files stored on client machine with key-value pairs, sent with each HTTP request.

Q: What are the risks of client-side session storage? A: Security vulnerabilities, data tampering, size limitations, dependency on client settings.

Practical 11: Server-Side Session Management

Q: What is server-side session management? A: Storing session data on web server, identified by session ID sent to client via cookie.

Q: How does ASP.NET manage sessions on the server? A: Uses Session object, stores data in server memory/database, tracks using session ID.

Q: What is the difference between Session, Application, and ViewState? A: Session: Per user data. Application: Global data for all users. ViewState: Page-level data between postbacks.

Practical 12: AJAX Controls

Q: What is AJAX? A: Asynchronous JavaScript and XML - enables partial page updates without full page refresh.

Q: How does AJAX improve user experience? A: Faster response, reduced bandwidth, better interactivity, no page flickering.

Q: What are some built-in AJAX controls in ASP.NET? A: UpdatePanel, ScriptManager, UpdateProgress, Timer, AsyncFileUpload.

Practical 13: Produce and Consume Web Service

Q: What is a web service? A: Platform-independent service accessible over HTTP using standard protocols like SOAP/REST.

Q: What are the protocols used in web services (SOAP/REST)? A: SOAP: XML-based messaging protocol. REST: HTTP-based architectural style using standard HTTP methods.

Q: How do you consume a web service in ASP.NET? A: Add web reference/service reference, create proxy class, call web service methods.

Practical 14: MVC Framework

Q: What is the MVC design pattern? A: Separates application into Model (data), View (UI), Controller (logic) for better maintainability.

Q: Explain the role of Model, View, and Controller. A: Model: Data and business logic. View: User interface. Controller: Handles user input and coordinates Model-View.

Q: How is routing handled in MVC? A: URL routing maps URLs to controller actions using route table configuration.

Practical 15: MVC + Entity Framework

Q: How is Entity Framework used in an MVC application? A: Models represent EF entities, Controllers use DbContext for data operations, Views display data.

Q: What is a DbContext? A: Primary class for interacting with database, manages entity objects and handles database connections.

Q: Explain Code First migration in EF. A: Automatically updates database schema based on model changes using migration files.

Practical 16: MVC + CRUD Operations

Q: What are CRUD operations? A: Create, Read, Update, Delete - basic database operations.

Q: How do you implement CRUD in an MVC project? A: Create controller actions for each operation, corresponding views, and use Entity Framework for data access.

Q: What HTTP methods are used for each CRUD operation? A: Create: POST, Read: GET, Update: PUT/POST, Delete: DELETE/POST.