LTI
Let's Solve

Celebrating 20 Years

# .NET Course Content

A Larsen & Toubro Group Company

**Objectives of this Course:**

- To understand the web applications that are scalable, maintainable.
- To understand the architecture and design of web applications.
- To understand how to separate the application concerns based on functionality.
- To understand effective and clean division between controllers, models and view using ASP MVC.
- To understand the Modern Development techniques using Frameworks like jQuery and Angular.

**Prerequisites:**

- Knowledge of Basic Programming Techniques, Basic Database Functionalities and Basic SDLC.

# Course Outline

**Day 1 - 3:**

**Soft Skill Training**

## Boot Camp Phase – I

**Day 4 – 7:**

**DBMS – DBMS Concepts and SQL**

- Introduction to Databases
- Database Models.
  - Relational Model
- Data Design and Normalization
- Structured Query Language and its categories
  - DDL – DML – DQL – DCL – TCL
- Selecting Data from Columns
  - All columns
  - Some columns
  - Derived columns
  - Using DISTINCT
  - Naming result columns
  - WHERE and comparison operators
  - Nulls
- Selecting using Operators
  - Arithmetic Operators
  - Relational Operators
  - Logical Operators
  - Other Operators
- Selecting using Functions
  - Number Functions
  - Character Functions
  - Date and Time Functions
  - Aggregate Functions
  - Other Functions
- ORDER BY
  - ASC
  - DESC
  - Multiple columns
  - Expressions
  - Columns not in SELECT list
- GROUP BY
  - Single column
  - Multiple columns
  - HAVING considerations

- With ORDER BY
- JOINS
  - INNER
  - OUTER (LEFT, RIGHT & FULL)
  - ON vs. WHERE
  - Cartesian product
- Implementing Data Integrity by using Constraints
  - Data Integrity Overview
  - Creating Constraints
  - Implementing Constraints
  - Not Null
  - Unique Key
  - Primary key
  - Check Constraints
  - Default
  - Foreign Key
  - Disabling Constraints
- Transaction Management
  - What is Transaction
  - Commit
  - Rollback
- Implementing Views
  - Introduction to Views
  - Creating and Managing Views
- T-SQL Programming
  - Variable Declarations
  - Programming Constructs
  - Conditional statements
  - If-else
  - Case
  - While
  - Break
  - Continue
- Implementing Stored Procedures
  - What is Stored Procedure
  - Creating Stored Procedures
  - Executing Stored Procedures
  - Creating Parameterized Stored Procedures
  - Handle errors in a stored procedure
- Implementing Functions
  - Creating Functions
  - Implement Scalar Functions
  - Create Table Valued Functions
- Implementing Triggers
  - INSERT triggers
  - DELETE triggers
  - UPDATE triggers

**Day 8 – 18:**

**C# Programming with ADO.NET**

**.Net Framework**

- Introduction t.Net Framework
- Compilation Process
- CLR
- CLS
- CTS
- Framework Library
- Different .Net Framework Version
- Assembly
- Overview of New Features of .Net 4.5

**C# Types**

- Value and Ref Types
- Struct
- String Manipulation
- Enum
- Arrays
- Boxing and Unboxing
- Type Conversion
- Scope

**C# Flow Control**

- Branching
- Switching
- Looping
- Using Foreach
- Jumping

**C# Methods**

- Method Overview
- Passing Parameter
    - IN
    - OUT
    - REF
- Param array

**Object Oriented Programming**

- Classes and Objects
- namespaces
- Constructor
- Properties and indexers
- Inheritance
- Access Modifiers
- Virtual members
- Abstract classes
- Static
- Read-only and const fields
- Interfaces

**Exception Handling**

- Built in Exceptions
- Handling Exceptions
- Custom Exception classes
- Throwing exceptions
- Properties in Exception class

**Generics**

- Need of Generics
- Generic Classes
- Generic Methods
- Generic Constraints

**Collections**

- Non-generic Collections
- Generic Collections
    - List
    - Stack
    - Queue
        - Dictionary
        - SortedList
- Benefits of Generic Collections

**ADO.NET**

- Overview of ADO.NET
- History of ADO.NET
- ADO.NET Architecture
- SqlDataReader
- Performing CRUD Operations using Connected

- DataSet
- Performing CRUD Operations using Disconnected
- Local and Distributed Transactions
- SqlTransaction Class and its methods
- TransactionScope class

**Introduction to Agile (Scrum)**

**C# Programming & ADO.Net – Mini Project**

**Interviews by internal SMEs**

## Boot Camp Phase – II

**Day 19 - 23:**

**Web Technologies - Web Concepts**

- Introduction to the Internet and the World Wide Web
- Understanding the concept of Protocols
- Why Web Standards?

**Web Technologies – HTML**

- Overview of Hypertext Mark-up Language (HTML) and Cascading Style Sheet (CSS)
- Understanding & using HTML
- HTML headings
- HTML Paragraphs
- HTML Line Breaks & Rules
- Font tags
- Hyperlinks
- The Image Tag and the Src Attribute
- LIST Tags
- Tables
- Forms

**Web Technologies - HTML 5**

- HTML5 Intro
- HTML5 New Elements
- HTML5 Semantics
- HTML5 Style Guide
- HTML5 Graphics
- HTML5 Canvas
- HTML5 Google Maps
- HTML5 Vide

- HTML5 Audi
- HTML5 Plug-ins
- HTML5 YouTube

**Web Technologies – Cascading Style Sheet (CSS)**

- Introduction to CSS
- Understanding & using CSS
- CSS Syntax
- CSS classes
- CSS IDs
- CSS Margins
- CSS Text Properties
- Font Properties
- CSS links
- CSS Backgrounds
- CSS Border
- Lists

**Web Technologies - CSS 3**

- CSS3 Introduction
- CSS3 Modules
- Selectors
- Box Model
- Backgrounds and Borders
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface
- Borders
    - border-radius
    - box-shadow
    - border-image
- CSS3 Backgrounds
    - background-size
    - background-origin
- CSS3 Text Effects
    - text-shadow
    - word-wrap

**Web Technologies – JavaScript**

- What is JavaScript?
- Setting up Variables in JavaScript
- Javascript Conditional Statements

- JavaScript Loops
- Arrays
- Javascript Events and Functions
- JavaScript Form Validation

**Web Technologies – jQuery**

- Introduction To jQuery
- Selection and DOM Traversal
- Working with JavaScript Events

**Advanced Web Technologies – Angular 7**

- Angular - Introduction
- Understanding Single Page Applications (SPA)
- AngularJS 1.x vs Angular recent versions
- Introduction to TypeScript
    - Role of typescript in Angular
- Developing a simple Angular application
- Writing custom components
- Understanding One-way data binding
- Understanding Two-way data binding
- Angular forms
- Introduction to Angular Routing and DI (Dependency Injection)

**Day 24 - 27:**

**ASP.NET**

**ASP.NET Fundamentals**

- History of ASP.NET
- Inline and Code behind Technique
- Server Controls
- Page Basics
- Application and Page Life Cycle

**Programming with Server Controls**

- Standard Control
- Validation Controls
- Data Controls
- User Controls
- Navigation Controls
- Membership Controls

**State Management**

- Need of state Management
- ViewState
- Cookie
- Session
- Application
- Session and Application Events

**Themes**

- Page Level
- Application level

**MasterPage**

- Creating Master Page
- Access master page data in Child pages

**Binding using ASP.NET**

- Working with Data Controls
- Inserting, Updating and Deleting in Gridview
- Sorting and Paging in Gridview
- DataSource Controls
- Detailsview
- Formview
- DataList
- Repeater Control

**Day 28 - 30:**

**WCF**

**WCF Essentials**

- What is Windows Communication Foundation (WCF)
- What is Address, Binding & Contract
- WCF Services & Clients
- WCF Service Libraries
- WCF Test Host and Test Client
- Self-Hosting
- WCF Clients
- Channel Factories
- Creating Proxies
- Configuration Files
- Metadata

- Standard Endpoints
- WCF Architecture

**Addresses and Bindings**

- Addresses
- Bindings
- Message Exchange Patterns
- Configuring Bindings
- Interoperating with ASMX Web Services
- Default Endpoints and Bindings
- Service Descriptions
- Multiple Endpoints
- Service Contracts
- Defining Service Contracts
- Defining Operation Contracts
- Services with Multiple Contracts
- Contract Inheritance
- Operation Overloading

**Instance Management**

- Using Per-Call Services
- Using Per-Session Services
- Using Singleton Services
- Configuring Behaviors

**Data Contracts**

- Implementing Data Contracts
- Mapping Data Contracts to XSD Schema
- Serialization
- Arrays and Generic Collections
- Enums
- Versioning Data Contracts

**More about Service Contracts**

- Versioning Service Contracts
- Implementing Message Exchange Patterns
- Oneway Contracts
- Duplex Contracts
- Asynchronous Proxies

**Handling Errors**

- Faults and Exceptions
- Generating and Handling Faults
- Fault Contracts
- Faults and Sessions

**WCF Security**

- Security Aspects of Services
- Transport Security
- Message Security
- Certificates
- Credentials

**Assessment**

## Specialization Phase

**Day 31 - 35:**

**DevOps – Before DevOps**

- Evaluation of Software Development and Methodology
- Comparison of different Methodology
- Software Development Life Cycle and Tool Sets
- Introduction of Agile Methodology (Main Principle, Ceremonies, User Story, Roles etc.)
- Challenges in Agile

**DevOps – DevOps Introduction**

- Who coined the term and how the movement started?
- Elaborate DevOps, DevTestOps, DevSecOps
- Need for DevOps: in the context of our case study and in general
- DevOps Key Players and trends
- DevOps: Toolset
- Is Agile mandatory for the culture of DevOps
- Continuous Delivery Vs Continuous Deployment

**TFS – Team Foundation Server**

- Intro to Continuous Integration
- Intro to TFS
- Understanding Team Projects
- Understanding Work Items
- TFS Version control concepts

- Understanding Source Control Explorer
- Getting Code into Team Foundation Version Control
- Linking Changesets to Work Items
- Working with Workspaces
- Check-In Policies
- Understanding Branching in TFS
- Branch Visualization and Tracking Changes
- Merging and Resolving Conflicts

**GIT Outline**

- What is GIT
- Exploring a GITHUB Repository
- Understanding the GitHub flow
- Branching with GIT
- Creating a local copy of the repository
- Git Status
- Using Branches Locally
- Switching Branches
- Pushing your changes to GitHub
- Creating a Pull Request
- Editing a file on GitHub
- Committing changes on GitHub
- Merging pull Requests
- Updating the local Repository
- Cleaning up the Unneeded Branches
- Using Git Log
- Understanding GitHub Pages
- Creating a Fork
- Resolving Merge Conflicts
- Viewing Local Changes

**Azure**

- Introduction to cloud computing
- About Azure, introduction, Managing Infrastructure on Microsoft Azure
- Azure Accounts, Subscriptions and Admin Roles
- Navigating the Azure Portal
- Getting started with Microsoft Azure Virtual Machines
- IaaS, PaaS, SaaS,
- Getting Started with Microsoft Azure IaaS Monitoring and Management
- Hosting Web Application (ASP.NET App) on Microsoft Azure

**Day 36 – 41:**

**ASP MVC**

**Introduction**

- Introduction to different Web Technology
- What is ASP.NET MVC
- Role of Model, View, and Controller
- How ASP.NET MVC Works
- Key Benefits of ASP.NET MVC
- Understanding the structure of an ASP.NET MVC project

**ASP.NET MVC Architecture**

- The MVC Pattern
- MVC Page Life Cycle
- Controllers, Models, and Views
- URL Routing
- Controller Actions
- Razor View Engine
- Extensibility

**URL Routing**

- Introducing URL Patterns
- Creating and Registering a Simple Route
- Defining Default Values
- Using Static URL Segments
- Defining Custom Segment Variables
- Constraining Routes
- Using Attribute Routing
- Generating Outgoing URLs in Views

**Razor View Engine**

- Razor Basics
- Razor design goals
- Implementation of Razor view
- Razor syntax
- Using Razor Expressions
- Accessing Model Data in Razor views

**Views**

- View Engines
- Templates and Scaffolding

- ViewData and ViewBag
- Strongly-Typed Views
- Layout Pages
- Custom Sections
- Partial Views
- Child Actions
- Using a ViewModel Object
- Bundling & Minification

**HTML Helpers**

- Basic Helpers
- Strongly-Typed Helpers
- Creating Custom Helpers
- Declarative Helper

**Controllers and Actions**

- IController, ControllerBase, and Controller
- Defining Actions
- Action Selectors, Action Filters
- HTTP Verbs
- HttpContext and RouteData
- Returning Data with ActionResult
- Parameters and the Model Binder

**Entity Framework**

- What is Entity Framework
- EF Architecture
- Creating Entity Data Model
- Model Browser
- DB Context
- Eager and Lazy Loading
- Types of Entity
- Entity Lifecycle

**Model Binding**

- Object Relational Mapping (ORM)
- Entity Framework (EF) Database-first approach
- Entity Framework (EF) Code-first approach
- Entity Framework (EF) Model-first approach
- Model Binders
- Creating Unit Testable Applications in ASP.NET MVC

**Model Validation**

- Data Annotations
- Validation HTML Helpers
- Model State
- Client-Side Validation

**Security in MVC**

- Authentication and Authorization
- ASP.NET Identity
- Configuring Forms Authentication
- MVC 5 App with Facebook, and Google OAuth2 Sign-on
- Enable role-based security
- Authorize attribute

**Filters**

- Introducing the Filter Types
- Using Authorization Filters
- Using Authentication Filters
- Using Exception Filters
- Using Action Filters
- Using Result Filters
- Using Other Filter Features

**Integration of Angular in ASP MVC**

**Assessment**

## Project Gladiator

**Day 42 - 49:**

**Project Gladiator**
**Project Evaluation by SMEs**

**Day 50:**

**Brain bench Test Preparation & Test**