LTI
Let's Solve

Celebrating 20 Years

.NET Full Stack Course Content

A Larsen & Toubro Group Company

Objectives of this Course:

- To understand role of Database in our applications
- To understand how C# plays an important role in .NET applications
- To understand the power of .NET Core over .NET framework
- To understand Web applications that are scalable, maintainable
- To understand the Architecture and Design of Web applications
- To understand modern development techniques using JS Frameworks like Angular
- To understand how to separate the application concerns based on functionality
- To understand effective and clean division between Controllers, Models and View using ASP.NET MVC Core
- To understand Data Persistence using Entity Framework
- To understand importance of DevOps & various tools like Git, Maven, Jenkins
- To understand Agile methodology & Scrum software development practices

Prerequisites:

- Basic Knowledge of Programming Techniques, Database & SDLC

Agenda:

Module	Duration (Days)
Bootcamp Phase	
Communication Skills	**3**
DBMS	**3**
C# Programming with .NET Core & C# Codility	**6**
Web Technologies	**5**
Specialization Phase	
DevOps	**3**
ASP.NET Core MVC	**7**
Project Phase	
Project Gladiator	**5**

Course Outline

Boot Camp Phase

Communication Training (3 Days)

- Spread across 3 weeks / 1 day per week

DBMS

Duration: 3 Days

DBMS Concepts and MS SQL Server

- Introduction to Databases
- Database Models
  - Relational Model
- Data Design and Normalization
- Structured Query Language and its categories
  - DDL – DML – DQL – DCL – TCL
- SELECT statement varieties with clauses
  - WHERE clause
  - GROUP BY clause
  - HAVING clause
  - ORDER BY clause
- Using SQL Server built in Functions
- Joining the tables – Join variants
  - Equi and Non-Equi Joins
  - Self-Join
  - Cartesian Product
  - Outer Join
- Subqueries
- Implementing Views
- Implementing Data Integrity by using Constraints
  - Data Integrity Overview
  - Creating Constraints
  - Implementing Constraints
  - Not Null
  - Unique Key
  - Primary key
  - Check Constraints
  - Default
  - Foreign Key
- Implementing Stored Procedures and Functions

C# 8.0 Programming with .NET Core 3.0

Duration: 5 Days

.NET Core:

- Intro to .NET Core
- Why .NET Core
- Advantages
- Features
- Creating a console application in .NET core
- Building from Command Prompt
- Executing a Core Project from Command Prompt

C# Types

- Value and Ref Types
- String Manipulation
- Arrays
- Boxing and Unboxing
- Type Conversion
- Scope
- Nullable Types
- Named and Optional Arguments

C# Flow Control

- Branching
- Switching
- Looping
- Using Foreach

Object Oriented Programming

- Characteristics of Object-Oriented Programming
- Classes and Objects
- namespaces
- Constructor
- Properties
- 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

Generics and Collections

- Need of Generics
- Generic Classes
- Generic Methods & Constraints
- Non-generic Collections
- Generic Collections
  - List
  - Stack
  - Queue
    - Dictionary
    - SortedList
- Benefits of Generic Collections

LINQ

- Introduction to Language Integrated Query
- Query a collection of objects

LAMBDA Expressions

- Introduction to Expression Language Syntax using Lambda Expression

ADO.NET

- Overview of ADO.NET
- ADO.NET APIs
- Performing CRUD Operations using Connected
- Performing CRUD Operations using Disconnected

C# Codility

Duration: 1 Day

Introduction to Data Structures & Algorithms

- Introduction to Codility Platform
- Understanding Computational Thinking
- Understanding Space and Time Complexity
- Understanding Big-O notation
- Algorithm Run Time Analysis

Coding Problems and Challenges

- Iterations & Arrays
  - Iteration techniques for arrays and collections
  - Solving coding challenges on Arrays
- Time Complexity
  - Revisiting Big-O notations
  - Writing efficient algorithms to improve performance

Web Technologies

Duration: 5 Days

Web Technologies – HTML

- Understanding & using HTML5

Web Technologies - CSS 3

- CSS3 Introduction and commonly used CSS 3 properties

Web Technologies – JavaScript

- Introduction to JavaScript
- JavaScript Events and Functions
- JavaScript Form Validation

Advanced Web Technologies – Angular 10

- 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 and it's types
- Form validation
- Angular Routing and DI (Dependency Injection)

Assessment + Mini Project + Mock Client Interview

Specialization Phase

DevOps

Duration: 3 Days

DevOps – Overview of DevOps

- What is DevOps
- Continuous Integration
- Continuous Deployment

GIT: Version Control

- Introduction to Git and Github
- About Version Control System and Types
- GIT Basics
- GIT Command Line
- Creating repository
- Cloning, check-in and committing

- Fetch pull and remote
- Branching
    - Creating, switching and merging branches

NUnit2

- Overview
- Unit Testing and NUnit Overview
- Naming Conventions and Organizing Tests
- Writing Test Methods
- Assertions

Jenkins – Continuous Integration / Continuous Delivery/Deployment (CI/CD)

- Understanding CI/CD
- Introduction about Jenkins
- Build Cycle
- Jenkins Architecture
- Installation
    - Installing and configuring Jenkins
- Exploring Jenkins Dashboard
- Jobs
    - Creating Jobs
    - Running the Jobs
    - Setting up the global environments for Jobs
- Adding and updating Plugins
- Disabling and deleting jobs

ASP .NET Core MVC

Duration: 7 Days

Overview of ASP.NET Core

- Understanding ASP.NET Core
- ASP.NET vs. MVC vs. ASP.NET Web Form vs. ASP.NET MVC vs. ASP.NET Core
- Explaining Pipelines
- Explaining Middlewares
- Demonstrate Startup.cs Codes
- Main method in ASP.NET Core
- ASP.NET Core launchSettings.json and appSettings.json files
- ASP.NET Core Dependency Injection

ASP.NET Core MVC:

- Introduction to MVC
- How MVC Works
- Understanding Model, View and Controller
- ASP.NET Core AddMvc vs. AddMvcCore

Passing data to View:

- Views Discovery
- Loosely typed: ViewData and ViewBag
- Using strongly typed model object (Strongly typed Views)
- Understanding role of ViewModel

Layout Views:

- Need of Layout View
- Creating and using Layout View
- Section in Layout View
- Rendering body and section
- Understanding _ViewStart.cshtml and _ViewImports.cshtml

Routing:

- Understanding Default Routing
- UseMvc() and UseMvcWithDefaultRoutes() middlewares
- Conventional vs. Attribute Routing

TagHelpers:

- The Form Tag Helper
- The Label Tag Helper
- The Input Tag Helper
- The Textarea Tag Helper
- The Select Tag Helper
- The Anchor Tag Helper

Models in ASP.NET Core:

- Introducing Model
- Model Binding and Validation

Entity Framework Core:

- Installing EF Core
- EF Core DB-First and Code-First approach
- DbContext in EF
- Using SQL Server with EF
- EF Core Migrations
- EF Core conventions: OneToOne and OneToMany
- Performing CRUD operations using EF Core

ASP.NET Core WEB API:

- Introducing HTTP
- Setting up environment for ASP.NET Core WEB API
- Configuring Startup class and adding middleware for WEB API
- Understanding Route Attribute
- Passing values through URL and QueryStrings
- Demonstrating HTTPResponseMessages
- Explaining IHTTPActionResult

- Controller action return types:
  - Specific types: Simple and Complex Types
  - IActionResult
  - ActionResult<T>
  - IAsyncEnumerable<T>
- HTTP Request Methods:
  - GET
  - POST
  - PUT
  - DELETE
- HTTP Response Codes
- Testing Web API using PostMan

Integration of Angular with Web API

Assessment

Project Gladiator Phase

Duration :5 days

Project Gladiator

Project Gladiator Evaluation