Dot Net
Fresher's Induction Training
Course Curriculum

Curriculum Structure
(Course Duration: 35 Days)

Level	Targeted Participants	Topics to be covered	Day	Date	Duration in hours
Basic + Intermediate	20	CSS and HTML	Day 1	23-Aug-21	8
Basic + Intermediate	20	JavaScript	Day 2	24-Aug-21	8
Basic + Intermediate	20	JavaScript	Day 3	25-Aug-21	8
Basic + Intermediate	20	C#.NET Programming	Day 4	26-Aug-21	8
Basic + Intermediate	20	C#.NET Programming	Day 5	27-Aug-21	8
Basic + Intermediate	20	C#.NET Programming	Day 6	31-Aug-21	8
Basic + Intermediate	20	C#.NET Programming	Day 7	02-Sep-21	8
Basic + Intermediate	20	Windows Application and Multithreading	Day 8	03-Sep-21	8
Basic + Intermediate	20	SQL Server 2019	Day 9	04-Sep-21	8
Basic + Intermediate	20	SQL Server 2019	Day 10	06-Sep-21	8
Basic + Intermediate	20	SQL Server 2019	Day 11	07-Sep-21	8
Basic + Intermediate	20	ADO.NET	Day 12	08-Sep-21	8
Basic + Intermediate	20	ASP.NET	Day 13	09-Sep-21	8
Basic + Intermediate	20	Project Day – DB, HTML	Day 14	10-Sep-21	8
Basic + Intermediate	20	ASP.NET	Day 15	13-Sep-21	8
Basic + Intermediate	20	ASP.NET	Day 16	14-Sep-21	8
Basic + Intermediate	20	Entity Framework	Day 17	15-Sep-21	8
Basic + Intermediate	20	Entity Framework	Day 18	16-Sep-21	8
Basic + Intermediate	20	C#.NET Programming	Day 19	17-Sep-21	8
Basic + Intermediate	20	Windows Application and Multithreading	Day 20	20-Sep-21	8
Basic + Intermediate	20	Windows Communication Foundation	Day 21	21-Sep-21	8
Basic + Intermediate	20	ASP.NET WEB API	Day 22	22-Sep-21	8
Basic + Intermediate	20	ASP.NET WEB API	Day 23	23-Sep-21	8
Basic + Intermediate	20	Project Day	Day 24	24-Sep-21	8
Basic + Intermediate	20	Project Day	Day 25	27-Sep-21	8
Basic + Intermediate	20	Project Day	Day 26	28-Sep-21	8
Advance	5	ASP.NET MVC 5	Day 27	29-Sep-21	8
Advance	5	ASP.NET MVC 5	Day 28	30-Sep-21	8
Advance	5	ASP.NET MVC 5	Day 29	01-Oct-21	8

Advance	5	TypeScript and Angular, React	Day 30	04-Oct-21	8
Advance	5	TypeScript and Angular, React	Day 31	05-Oct-21	8
Advance	5	TypeScript and Angular, React	Day 32	06-Oct-21	8
Advance	5	TypeScript and Angular, React	Day 33	07-Oct-21	8
Advance	5	Upgrading to DotNet Core	Day 34	08-Oct-21	8
Advance	5	Project Day	Day 35	11-Oct-21	8

CSS and HTML

CSS

- Inline Styles
- Internal Styles
- Id and Class Id
- Selectors
- External Styles
- Backgrounds and Borders
- Text Effects
- Bootstrap
- Multiple Column Layout
- User Interface

HTML Overview

- History of HTML
- The HTML vision
- WHATWG and W3C specifications
- What is part of HTML5?
- HTML5 roadmap

HTML5 Markup

- HTML5 page structure
- HTML5 DOCTYPE
- HTML5 markup
- Structural elements
- Semantic elements
- Deprecated elements
- Images
- Lists
- Links
- Tables

HTML5 Forms

- HTML5 form elements
- Building HTML5 forms
- Using HTML5 forms

JavaScript

Day 1:

- **Introduction**
  - Introduction to JavaScript
  - Basic JavaScript constructs
    - Simple Functions
    - Functions with Parameters and Return values

- **Programming in JavaScript**
  - Variables
  - Data Types
  - Operators
  - Conditions
  - Looping
  - Working with HTML DOM
  - Alerts
  - Prompts

- **In-Built Objects in JavaScript**
  - In-Built objects
    - String object
    - Date object
    - Math object
    - Array object
    - Navigator object
    - History object
    - Window object
    - Screen object
    - Location object
  - Custom Object Creation in JavaScript

C#.NET Programming

Day 1:
MS.NET Fundamentals
- .NET Initiative
- .NET Frameworks
- Design Goals
- Language Support
- .NET Tools
- Common Language Runtime
- Common Language Specification
- MSIL (Micro Soft Intermediate Language)
- JIT (Just in time compilation)
- Application Execution/Managed Code Execution
- Namespaces
- Standard I/O

Introduction to C#.NET

Type Hierarchy
- Data Type Support
- Objects and basic Types
- Operators
- Reference and Value Types
- User-defined Types
- Boxing and UnBoxing

Iteration and Flow Of Controls
- Conditional Control Statements
- Relational and Logical Operators
- Loops
- Foreach Statement

Day 2:
Arrays
- Single-dimensional array
- 2-dimensional array
- Multi-dimensional array
- Jagged array

Classes and Objects
- Object-oriented design principles
- The relationship between classes and objects

Implementing and Using Classes
- Instantiating classes
- Building custom classes
- Adding properties, methods and Events
- Property Accessor Visibility
- Types Of Classes

Inheritance and Polymorphism
- Types of inheritance
- Abstract Methods
- Override Methods

- Shadowing Concept

Interfaces
- Defining Interfaces
- Inheritance of Interfaces
- Implementing interfaces
- Benefits of interfaces
- Interface inheritance
- Delegates and Events
- Indexers

Day 3:
Structured Exception Handling
- Exceptions
- Types Of Exceptions
- .Net Exceptions
- User-defined Exceptions
- Try, Catch, Finally
- Throwing Exceptions
- Creating and using User-defined Exceptions

Streams
- File Stream
- File and Directory Classes
- Stream Reader
- Stream Writer
- Serialization Concepts
- Binary Serialization
- SOAP Serialization

Assemblies
- Private, Shared and Satellite Assemblies
- Strong Names
- Versioning
- Assigning Public Key to Assemblies
- Configuring Assemblies
- Placing User-defined Assemblies into GAC [Global Assembly Cache]
- Benefits of Assemblies over DLL

Day 4:
C# 2005 New Features
- Partial Classes
- Anonymous Methods
- Iterators
- Nullable

Generics
- Generic Classes
- Generic Methods
- Generic Collection

XML
- XML Overview
- XML in Visual Studio .NET
- Well-formed XML
- XML Schemas
- Valid XML
- XML in .NET Framework

C# 2008 New Features

- Implicitly Typed Local Variables
- Automatic Properties
- Extension Methods.
- Partial Methods
- Object Initializer
- Collection Initializer
- Anonymous Types
- Lambda Expressions

Day 5:
LINQ

- Introduction to LINQ
- What is LINQ
- How LINQ Works
- Relational vs. Hierarchical/Graph Models
- Using Lambda and Extension Methods
- Language Integration
- Declarative Programming
- Type Checking
- LINQ Flavors
    - LINQ to Objects
    - LINQ to Database
    - LINQ to XML

C#.NET 2010 New Features

- Optional Parameters
- Named Parameters
- Covariance and Contravariance
- Dynamic Support
- Office Programmability

Testing

- What is Software Testing
- Different Types of Testing
- Unit Testing
- Test Driven Development (TDD)
- Manual Testing
- Automated Testing
- Nunit Testing

SQL Server 2019

Day 1:
Introduction to SQL Server 2019 and its Toolset
- Introduction to the SQL Server Platform
- Working with SQL Server Tools
- Configuring SQL Server Services

Working with Data Types
- Using Data Types
- Working with Character Data
- Converting Data Types
- Specialized Data Types

Designing and Implementing Tables
- Designing Tables
- Working with Schemas
- Creating and Altering Tables

Ensuring Data Integrity through Constraints
- Enforcing Data Integrity
- Implementing Domain Integrity
- Implementing Entity and Referential Integrity

Basic SELECT statements
- Write simple SELECT statements
- Eliminate Duplicates using DISTINCT
- Use table and column aliases
- Use a simple CASE expression

Day 2:
Querying Multiple Tables
- Understanding Joins
- Querying With Inner Joins
- Querying With Outer Joins
- Querying Using Self and Cross Joins

Sorting and Filtering Data
- Sorting Data
- Filtering Data
- Filtering with the TOP and OFFSET-FETCH Options

Using Built-In Functions
- Writing Queries with Built-In Functions
- Using Conversion Functions
- Using Logical Functions

Grouping and Aggregating Data
- Using Aggregate Functions
- Using the GROUP BY Clause
- Filtering Groups with the HAVING Clause

Using Subqueries
- Writing Self-Contained Subqueries
- Writing Correlated Subqueries

- Using the EXISTS Predicate with Subqueries

Planning for SQL Server Indexing
- Core Indexing Concepts
- Data Types and Indexes
- Single Column and Composite Indexes

Implementing Table Structures in SQL Server
- SQL Server Table Structures
- Working with Clustered Indexes
- Designing Effective Clustered Indexes

Reading SQL Server Execution Plans
- Execution Plan Core Concepts
- Common Execution Plan Elements
- Working with Execution Plans

Improving Performance through Nonclustered Indexes
- Designing Effective Nonclustered Indexes
- Implementing Nonclustered Indexes
- Using the Database Engine Tuning Advisor

Day 3:
Designing and Implementing Views
- Introduction to Views
- Creating and Managing Views
- Performance Considerations for Views

Designing and Implementing Stored Procedures
- Introduction to Stored Procedures
- Working With Stored Procedures
- Implementing Parameterized Stored Procedures
- Controlling Execution Context

Merging Data and Passing Tables
- Using the MERGE Statement
- Implementing Table Types
- Using TABLE Types As Parameters

Designing and Implementing User-Defined Functions
- Overview of Functions
- Designing and Implementing Scalar Functions
- Designing and Implementing Table-Valued Functions
- Implementation Considerations for Functions
- Alternatives to Functions

ADO.NET and Entity Framework & ASP.Net

Accessing Data with ADO.NET

- Introduction to ADO.NET
- Using ADO.NET Managed Providers
- Retrieving Data
- Executing DML Commands
- DataReader
- Asynchronous Data Access
- DataSet
- Creating DataSet
- DataTables and DataRelations
- Updating DataSet
- XML and ADO.NET

ADO.NET Entity Framework

- ORM Frameworks
- History of Entity Framework
- EF vs ADO.NET
- Adding Entity Data Model
- Database First Approach
- Code First Approach
- Model First Approach
- Working with edmx File
- Entity Classes
- Navigation Properties
- Entities Context Class
- Writing LINQ Queries with EF
- Manipulating Data with EF

Code First Approach

- Coding Model Classes
- Applying Attributes for Key Columns
- Adding Navigation Properties
- Coding DataContext Class
- Setting Up Connection String
- Creating Migration
- Updating the Database

ASP.NET

- **Introduction to ASP.NET**
    - A Review of Classic ASP
    - ASP.NET Web Applications
    - Rendering HTML with Server Controls
    - Data Binding in ASP.NET
- **Working with Controls**
    - Introduction to Web Controls
    - Simple Input Controls
    - HyperLink and Button Controls
    - List Controls
    - Controlling Focus

- **Configuration**
    - Configuration Overview
    - Using the Web Site Administration Tool
    - Programming Configuration Files
    - Encrypting Configuration Sections
- **Data Binding**
    - Introducing Data Source Controls
    - Reading and Write Data Using the SqlDataSource Control
    - Displaying and Editing Middle-Tier Data using the ObjectDataSource Control
    - Displaying XML Data Using the XmlDataSource Control
- **Validating User Input**
    - Overview of ASP.NET Validation Controls
    - Using the Simple Validators
    - Using the Complex Validators
    - Summarizing Results with the ValidationSummary Control
    - Separating Validation into Validation Groups
- **Themes and Master Pages**
    - Creating a Consistent Web Site
    - ASP.NET 2.0 Themes
    - Master Pages
- **Managing State**
    - Preserving State in Web Applications
    - Page-Level State
    - Using Cookies to Preserve State
    - ASP.NET Session State
    - Storing Objects in Session State
    - Configuring Session State
    - Setting Up an Out-of-Process State Server
    - Storing Session State in SQL Server
    - Application State

Windows Application and Multithreading

Windows Forms Applications

- Console vs Windows Applications
- Creating Windows Forms Applications
- ToolBox and Controls
- Properties Window
- Designing a Windows Form
- Coding Event Handlers
- Creating MDI Applications

Windows Services

- Application vs Service
- Creating a Windows Service
- Coding a Windows Service
- ToolBox and Controls
- Installing a Windows Service
- Starting and Stopping a Windows Service
- Uninstalling a Windows Service

Concurrency programming in .Net (Threads)

- Managed Thread v/s Process Thread
- Creating Thread
- Application Thread v/s Background Thread
- Thread Lifecycle
- Thread Synchronization
  - i. Auto
  - ii. Manual
- Lock Managers
  - i. Mutex
  - ii. Monitors
  - iii. Events
  - iv. Interlocked
- Thread Debugging
- Asynchronous Programming Model

Windows Communication Foundation

- **Introduction to SOA**
  - Why SOA?
  - Role of WCF in SOA
  - First Generation Web Services
  - Second Generation Web Services (WS- *)

- **WCF Basics**
  - Address, Binding and Contract in WCF
  - Service Contract
  - Operation Contract
  - Data Contract
  - Message Contract

- **Message Exchange Pattern**
  - Request Reply
  - One–Way
  - Asynchronous
  - Duplex

- **Service and Operation Contract**
  - Service Instance management
  - Service Behavior
  - Operation Behavior
  - Sessions

- **Addressing and Bindings**
  - Addressing Schemes
  - Standard Bindings
  - NetTcpBinding
  - NetPeerTcpBinding
  - BasicHttpBinding
  - WSHttpBinding
  - WSDualHttpBinding
  - NetMsmqBinding

ASP.NET MVC 5

❖ **ASP.NET MVC Overview**
- Introduction
  - ♦ Advantages of an MVC-Based Web Application
  - ♦ Advantages of Web Forms-Based Web Application
  - ♦ Features of the ASP.NET MVC Framework
- Execution Process
  - ♦ UrlRoutingModule
  - ♦ MvcHandler
- Understanding Model, View and Controller
  - ♦ Understanding Controllers
  - ♦ Understanding Views
  - ♦ Understanding Models
- Creating an ASP.NET MVC Application
  - ♦ Create the Project
  - ♦ Create the Database
  - ♦ Create the Database Model
  - ♦ Create the Controller
  - ♦ Create the Views

❖ **Routing**
- Overview
  - ♦ Using the Default Route Table
- Creating Custom Routes
  - ♦ Adding a Custom Route to the Route Table
  - ♦ Mapping URLs to Controller Actions
- Creating a Route Constraint
- Creating a Custom Route Constraint
  - ♦ IRouteConstraint Interface

❖ **Controllers**
- ASP.NET MVC Controller Overview
  - ♦ Understanding Controllers
  - ♦ Understanding Controller Actions
  - ♦ Understanding Action Results
  - ♦ Types of Action Results
- Creating a Controller
  - ♦ Add Controller Menu Option
  - ♦ Scaffolding Action Methods
  - ♦ Creating a Controller Class
- Creating an Action
  - ♦ Adding an Action to a Controller
  - ♦ Preventing a Public Method from Being Invoked

❖ **Views**
- ASP.NET MVC Views Overview
  - ♦ Understanding Views
  - ♦ Adding Content to a View
  - ♦ Using HTML Helpers to Generate View Content
  - ♦ Using View Data to Pass Data to a View
- Creating Custom HTML Helpers
  - ♦ Understanding HTML Helpers
  - ♦ Creating HMTL Helpers with Static Methods

    - Creating HTML Helpers with Extension Methods
- Displaying Views
- Displaying a Table of Database Data
    - Creating Model Classes
    - Formatting Within a View
    - Formatting Within a Partial
- Using the TagBuilder Class to Build HTML Helpers
    - Overview of the TagBuilder Class
    - Methods and Properties of the TagBuilder Class
    - Creating an Image HTML Helper

❖ **Models**
- Creating Model Classes with the Entity Framework
    - Creating the ADO.NET Entity Data Model
    - Modifying the ADO.NET Entity Data Model
    - Selecting Database Records with the Entity Framework
    - Inserting, Updating and Deleting Records with the Entity Framework
- Creating Model Classes with LINQ to SQL
    - Create LINQ to SQL Classes
    - Using LINQ to SQL in a Controller Action
    - Using the Repository Pattern

❖ **ASP.NET MVC Validation**
- Performing Simple Validation
    - Understanding Model State
    - Using the Validation Helpers
    - CSS Classes
    - Prebinding Validation and Postbinding Validation
- Validating with the IDataErrorInfo Interface
    - OnChanging and OnChanged Partial Methods
    - Implementing the IDataErrorInfo Interface
- Validating with a Service Layer
    - Separating Concerns
    - Creating a Service Layer
    - Decoupling the Service Layer
- Data Annotations
    - Using Buddy Class

❖ **Master Pages**
- Creating Page Layouts with View Master Pages
    - Creating a View Master Page
    - Creating a View Content Page
    - Modifying View Master Page Content
- Passing Data to View Master Pages
    - Adding View Data in Each and Every Controller Action
    - Adding View Data in Application Controller

❖ **Action Filters and Model Binders**
- Built-in Action Filters
- Using an Action Filter
- Types of Filters
- The Base ActionFilterAttribute Class
- Creating a Simple Action Filter

❖ **Security**
- Authenticating Users with Forms Authentication
    - Using the Web Site Administration Tool

        - Requiring Authorization
        - Authorizing by User Name or Role
        - Configuring Authentication
        - Configuring Database Permissions
    - Authenticating Users with Windows Authentication
        - Enabling Windows Authentication
        - Authorizing Windows Users and Groups

- **Navigation**
    - Website Navigation with SiteMaps
        - Creating a SiteMap
        - Understanding the SiteMap API
        - Creating a Menu HTML Helper
        - The Importance of Canonical URLs

- **AJAX Integration**
    - Add AJAX Functionality in ASP.NET MVC Applications
        - Why use AJAX?
        - Adding the Required JavaScript Files
        - Refactoring the Index View to AJAX
        - Adding jQuery Animation Effects
        - Adding Browser History Support
        - Performing AJAX Deletes

- **Web API**
        - Building an HTTP Service using ASP.NET Web API
        - Creating Web APIs
        - Web API Clients
        - Web API Routing and Actions
        - Working with HTTP
        - Formats and Model Binding
        - oData Support in ASP.NET Web API
        - Hosting ASP.NET Web API
        - Testing and Debugging
        - Extensibility

- **Testing**
    - Creating Unit Tests for ASP.NET MVC Applications
        - Creating the Controller Under Test
        - Testing the View Returned by a Controller
        - Testing the View Data Returned by a Controller
        - Testing the Action Result Returned by a Controller

TypeScript, Angular and React.js

Introduction to Angular 9

- What is Angular?
- Central Features of the Angular Framework
- Why Angular?
- Scope and Goal of Angular
- AngularJS vs Angular 2 vs. Angular 9

- Installing and Using Angular 9
- Adding Angular 9 and Dependencies to Your App
- Building Blocks of and Angular 9 Application
- A Basic Angular 9 Application
- New Features in Angular 9

Introduction to TypeScript

- Programming Languages for Use with Angular
- TypeScript Syntax
- The Type System – Defining Variables
- The Type System – Defining Arrays
- The Type System – Classes & Objects
- Class Constructors
- Interfaces
- Parameter and Return Value Types
- Working with Modules
- TypeScript Transpilation
- Arrow Functions
- Template Strings
- Generics

Components in Angular 9

- What is a Component?
- An Example Component
- Component Starter
- Developing a Simple Login Component
- Login Component: Add HTML
- The HTML Component Template
- Login Component
- Component Decorator Properties
- Component Lifecycle Hooks
- Using a Lifecycle Hook: OnInit

Data and Event Binding

- Binding Syntax
- One-Way Output Binding
- Binding Displayed Output Values
- Two-Way Binding of Input Fields
- Binding Events
- Setting Element Properties

Attribute Directives and Property Bindings

- What are Directives
- Directive Types
- Apply Styles by Changing Classes
- Applying Styles Directly
- Obsolete Directives and Property Binding
- Controlling Element Visibility

Structural Directives

- Structural Directives
- Adding and Removing Elements Dynamically
- Looping Using ngFor
- Swapping Elements with ngSwitch

Template Driven Forms

- Template Driven Forms
- Note on Deprecated Forms APIs
- A Basic Angular Form
- Binding Input Fields
- Accessing the Form Object
- Binding the Form Submit Event
- The Submit Function
- Basic HTML5 Validation - "required" Attribute
- HTML5 vs. Angular Validation
- Angular Validation
- Displaying Form Validation State
- Displaying Field Validation State
- Displaying Validation State Using Classes
- Disabling Submit when Form is Invalid
- Submitting the Form
- Binding to Object Variables
- Additional Input Types

Service and Dependency Injection

- What is a Service?
- Creating a Basic Service
- What is Dependency Injection?
- What Dependency Injection Looks Like
- Injecting Services
- Using a Service in a Component: Dedicated Instance
- Using onInit to Initialize Component Data
- Using a Shared Service Instance
- Dependency Injection

HTTP Client

- The Angular HTTP Client
- Setting up the Root Component
- Service Using Http Client
- Importing Individual HTTP Providers into Services
- Service Imports
- The Observable object type
- Making a Basic HTTP GET Call
- Using the Service in a Component
- Importing Observable Methods
- Enhancing the Service with .map() and .catch()
- Using .map()
- Using .catch()
- Using toPromise()
- GET Request
- GET Request with Options
- POST Request
- Reading HTTP Response Headers

Pipes and Data Formatting

- What are Pipes?
- Formatting Changes in Angular 6
- Built-In Pipes
- Using Pipes in HTML
- Chaining Pipes

- Using Pipes in JavaScript
- Using Custom Pipes
- A Filter Pipe
- A Sort Pipe

React.js
- Overview, Features, Advantage
- Installation and Setting up Environment
- JSX, Component, State
- Props, Component API, Forms , Events
- React Routing, Communication between component
- React AJAX Call

Upgrading to DotNet Core

What is .net Core
- .Net Core vs. .Net Framework
- Architecture of .Net Core
- What's New in .NET 5
- Introduction to Asp.Net Core MVC and Web API
- Middleware, Routing, URL Rewriting, Dependency Injection

Building web applications
- Your first ASP.NET Core application using Visual Studio Code
- Building your first ASP.NET Core MVC app with Visual Studio
- Getting started with ASP.NET Core and Entity Framework Core using Visual Studio
- Building projects with Yeoman
- Authoring Tag Helpers
- Creating a simple view component
- Developing ASP.NET Core applications using dotnet watch

Building web APIs
- Building your first web API with ASP.NET Core MVC and Visual Studio
- ASP.NET Web API Help Pages using Swagger

- Creating backend web services for native mobile applications

Working with data

- Getting started with ASP.NET Core and Entity Framework Core using Visual Studio
- ASP.NET Core with EF Core - new database
- ASP.NET Core with EF Core - existing database