Dot Net Full stack Internal Training:

Index: List of Content

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR No | Topic | Days | Description | Page No |
| 1 | Index |  | List of Content | 1 |
| 2 | System Requirement |  | System Requirement details | 2 |
| 1 | HTML 5, CSS,Jquery and BootStrap | 3 | Building UI components using HTML 5, CSS and Bootstrap | 2 |
| 2 | Typescript and Angular 8 | 4 | Implementation of Angular 8 for SPA and Angular Material | 8 |
| 3 | Assessment Project 2 | 2 | Creating a Real-World Application using the Above 2 technologies. | 11 |
| 4 | Here participants would go through the real-world exposure working on the project for the first time |
| 5 | They would be divided into team and project would be given to them |
| 6 | Advance SQL Server | 2 | Advance Store Procedure, Functions, Cursor, trigger, performance tuning and ACID properties handling | 11 |
| 7 | .Net Foundation | 4 | Build Strong Object Orientation Concepts with OOPS with hands on project on Banking, understanding new C# 7 features used for modern world application | 13 |
| 8 | C# 2.0 to 7.0 |
| 9 | ADO.Net, LINQ and Entity Framework | 2 | DB Connectivity using ADO.Net and LINQ and Entity Framework | 16 |
| 10 | ASP.NET MVC 5 with Web API 2 | 4 | Building the foundation for developing RESTful services that can be used to create Services | 18 |
| 11 | Assessment Project | 0 | Participants would be creating web Project on using ASP.Net MVC, Entity Framework and SQL Server as a Backend, A Banking Project that they created in the last assessment would now be web based. | 19 |
| 12 | .Net Design Patterns, Web Security Principles, TDD and BDD and NUNIT | 2 | Understanding various patterns and practices that helps to build maintainable and structured application using .NET | 19 |
| 13 | Case Study and Final Project, Plan, Architect / Design, Develop and Test a project based on SDLC, Design Pattern, Coding Standards | 4 | Understand various SDLC stages, how to create various artefacts needed for SW development and develop applications using .NET framework | 20 |
|  | Total | 27 days |  |  |

**System and Software Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| Software : | Hardware : | | Other Facilities : |
| Windows OS 7 or above | Intel Core i5 Processor or above | | Trainer Machine (As a back Up) |
| Visual Studio 2019  https://visualstudio.microsoft.com/downloads/  SQL Server 2016 Enterprise or above  https://www.microsoft.com/en-in/evalcenter/evaluate-sql-server-2016  Visual Studio Code  https://code.visualstudio.com/ | Min 4 GB Ram, 8 GB Recommended | | Projector |
| MS Office 2010 or Above (Optional) | 80 GB HDD | | Board and Markers |
| Acrobat Reader 8 or above | | Internet Connection | |
| NodeJS for (Angular)  https://nodejs.org/en/ | | Machines should have administrator rights  Networked Machine | |

**HTML5 and CSS3 with JQuery and Bootstrap**

Duration: 3 days, Methodology: Hands-On

**HTML5**

**HTML5 Overview**

History of HTML5

The HTML5 vision

WHATWG and W3C specifications

What is part of HTML5?

**Using HTML5 Today**

When can I use these features?

Using HTML5 in browsers that do not support it

Detecting native availability of HTML5 features

Emulation

**HTML5 Markup**

HTML5 page structure

HTML5 DOCTYPE

HTML5 markup

Structural elements

Semantic elements

Deprecated elements

**HTML5 Forms**

HTML5 form elements

Building and using HTML5 forms

**HTML5 Audio and Video**

The audio and video elements

Understanding audio and video

Audio and video containers

Audio and video codecs

**HTML5 Geo-location**

Geo-location overview

User Privacy

Location information sources

**CSS3**

 New CSS3 selectors

 Borders

 Rounded corners

 Drop shadows

 Backgrounds

 Text effects

 Adding shadows to text

 Colours and Opacity

 RGBA and HSL values and alpha channel opacity

 Setting opacity

 Transitions and transforms

 Using transitions

 Using transforms

 UI Properties

 Appearance

 Box-sizing

 Resize

 Outline-offset

 Generated Content Properties

 Content

 Media Queries

**HTML5 and JQuery3.1**

**HTML5 Canvas and SVG**

Overview

Canvas vs. SVG

Canvas coordinates

Context

Pixel data

**HTML5 Storage**

Overview

Local Storage

Session Storage

**HTML5 IndexedDB**

Overview

ObjectStore

Transaction

CRUD Operations

**HTML5 Offline Web Applications**

Offline Applications Overview

Manifest File

Online and Offline Events

**JQUERY 3.1**

 What is JQuery

 Need for JQuery

 Benefits of JQuery

 Overview of major JQuery functionality

 Setting up JQuery

JQuery Core Functionality

 Understanding the JQuery function

 Using Selectors

 Accessing UI Elements using JQuery

 DOM Manipulation

DOM Manipulation

 Adding Nodes

 Removing Nodes

 Searching for Nodes

 Inserting Nodes

 Updating Nodes

 Wrapping Content

 Manipulating CSS

**Working with Forms**

 Using JQuery with HTML Form elements

 Performing operations on Form elements

 Working with Events

 Understanding how JQuery processes events

 Setting up and removing event handlers using JQuery

 Overview of the Deferred Object

**Using Effects**

 Understanding different types of Effects

 Setting up Effects to run on various events

 Using Effects like Slide, Fade, Animate etc

**Using Ajax with JQuery –\*\***

 Overview of Ajax

 Understanding the different JQuery Ajax functions

 Using Ajax functions to perform asynchronous calls

**Bootstrap Introduction**

 What, Why Where and How

 Purpose and Advantages

**Download Bootstrap**

 File structure

 PRECOMPILED BOOTSTRAP

 Basic Examples

**Bootstrap Grid System**

 What is a Grid?

 What is Bootstrap Grid System?

 MOBILE FIRST STRATEGY

 Working of Bootstrap Grid System

 Media Queries

**Grid options**

 Responsive column resets

 Offset columns

 Nesting columns

 Column ordering

**Bootstrap CSS Overview**

 HTML5 doctype

 Mobile First

 Responsive images

 Typography and links

 Normalize

 Containers

**Bootstrap Typography**

 Headings

 INLINE SUBHEADINGS

 Emphasis

 Abbreviations

 Addresses

 Blockquotes

 Lists

**Bootstrap Tables**

 Basic Table

 Optional Table Classes

 STRIPED TABLE

 BORDERED TABLE

 HOVER TABLE

 CONDENSED TABLE

 Contextual classes

 Responsive tables

**Bootstrap Forms**

 Form Layout

 VERTICAL OR BASIC FORM

 INLINE FORM

 HORIZONTAL FORM

 Supported Form Controls

 CHECKBOXES AND RADIOS

 SELECTS

 Form Control States

 Form Control Sizing

 Help Text

**Bootstrap Buttons**

 Button Size

 Button State

 DISABLED STATE

 Button Tags

**Bootstrap Images**

 .img-rounded

 .img-circle

 .img-thumbnail

**Bootstrap Helper Classes**

 Close icon

 Carets

 Quick floats

 Center content blocks

 Clearfix

 Showing and hiding content

 Screen reader content

**Bootstrap Responsive utilities**

 visible-xs

 visible-sm

 visible-md

 visible-lg

**Bootstrap Glyphicons**

 What are Glyphicons?

 Where to find Glyphicons?

**Bootstrap Dropdowns**

 Options

 ALIGNMENT

 HEADERS

**Bootstrap Button Groups**

 Basic Button Group

 Button Toolbar

 Button Size

 Nesting

 Vertical Buttongroup

**Bootstrap Button Dropdowns**

 Split Button Dropdowns

 Button Dropdown Size

 Dropup variation

**Bootstrap Input Groups**

 Basic Input Group

 Input Group Sizing

 Checkboxes and radio addons

 Button addons

 Buttons with dropdowns

 Segmented buttons

**Bootstrap Navigation Elements**

 Tabular Navigation or Tabs

 Pills Navigation

 VERTICLE PILLS

 Justified Nav

 Disabled Links

 Dropdowns

 PILLS WITH DROPDOWNS

**Bootstrap Navbar**

 Default navbar

 Responsive navbar

 Forms in navbar

 Buttons in navbar

 Text in navbar

 Non-nav links

 Component alignment

 Fixed to top

 Fixed to bottom

 Static top

**Angular 8**

**Program Duration:** 4 days.

**Prerequisite Skills:** HTML, JavaScript, AJAX Basics, jQuery Basics (Optional), Bootstrap (Optional), Node fundamentals (Optional)

 **Introduction to AngularJS and Benefits**

 **Introduction to TypeScript**

o **Why use TypeScript?**

o **TypeScript Features**

o **Typing, Variables and Functions**

o **Overview**

o **Type Annotations**

o **Type Inference**

o **Grammar**

o **Static and Dynamic Typing**

o **Compile Time or Run Time**

o **Ambient Declarations and Type Definition Files**

o **The Any Type and Primatives**

o **Applying Types**

o **Objects**

o **Functions**

o **Arrow Functions and Debugging**

o **Functions and Interfaces**

 **Classes and Interfaces**

 **Modules**

 **Introduction to Angular 5**

o Setting up Angular Development Environment

o Understanding Angular Framework

o Angular Architecture

o Angular modularity

o Component Structure

 Decorators

 Selector

 Template

 Style

 Component Class

**Contents:**

 Angular 8 Fundamentals

o Angular 8 Introduction

o Why Angular

o Why Angular 8

o Angular 8 Code Editors

o Language Choice

o Building Blocks of an Angular 8

o Angular 8 Application

o Angular 8 Modules

o Introduction to Angular Components

o Angular 8 Component

o Building Angular Components Using TypeScript

o Component Class

o Component Metadata

o Angular 8 Metadata

o Importing Modules

o Templates

o Component Styles

 Data Binding

o Databinding

o Interpolation(One-Way binding)

o Property Binding

o Event Binding

o Two-way Binding

o Nested Components

 Directives and Pipes

o Structural Directives

o Attribute Directives

o @HostBinding and @HostListener

o Pipe

o Built-in Pipes

o Custom Pipes

 Component Lifecycle

o Component Lifecycle

o Lifecycle hooks

o Accessing Component Lifecycle hooks

o Lifecycle hooks execution order

 Services and Dependency Injection

o Introduction to Dependency Injection

o Dependency Injection in Angular 8

o Services

o Working with Services in Angular 8

o Building a Service

o Registering a Service

o Injecting a Service

o Using Providers in Angular 8

o Angular 8 DI System

o Working with Angular2 HTTP Service

 Angular 8 Forms

o Template Driven Forms

o Model Driven Forms

o Form Controls and Form Groups

o Using Form Builder

o Validating Form Controls

 Internals of Angular 8

o Observables

o Introducing RxJs

o Change Detection

o How Change Detection works

o Customizing Change Detection

o Angular 8 change detection system

 Routing

o Routing Introduction

o AngularJS Routes

o Routing Setup

o Components of Angular 8 routing

o RouterConfig

o RouterOutlet

o RouterLink

o Working with Route Parameters

================================================================

**ASSESSMENT PROJECT – 2 days**

================================================================

**Microsoft SQL Server for Developers**

**Duration: 2 Days Methodology: Hand-On**

**Day 1**

**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

 ADVANCE PROGRAMMING

 INSTEAD OF triggers

 Recursive triggers

 DDL triggers

 The MERGE Statement

 compound operators

 Intellisense

**Day 2**

 SQL Server Architecture overview

 **Performance Tuning Overview**

 The Performance-Tuning Process

 Performance vs. Price

 Performance Baseline

 Where to Focus Efforts

 SQL Server Performance Killers

 Query Design Analysis

 Query Design Recommendations

 Operating on Small Result Sets

 Using Indexes Effectively

 Avoiding Optimizer Hints

 Using Domain and Referential Integrity

 Avoiding Resource-Intensive Queries

 Reducing the Number of Network Round-Trips

 Reducing the Transaction Cost

 Blocking Analysis

 Blocking Fundamentals

 Understanding Blocking

 Locks

 Isolation Levels

 Effect of Indexes on Locking

 Capturing Blocking Information

 Blocking Resolutions

 Recommendations to Reduce Blocking

 Automation to Detect and Collect Blocking Information

 Deadlock Analysis

 Deadlock Fundamentals

 Using Error Handling to Catch a Deadlock

 Deadlock Analysis

 Avoiding Deadlocks

 Cursor Cost Analysis

 Cursor Fundamentals

 Cursor Cost Comparison

 Default Result Set

 Analyzing SQL Server Overhead with Cursors

 Cursor Recommendations

 Baselines

 Review Perfmon Baselines

 Review Performance Management Data Warehouse Baselines

 Use Baselines for Troubleshooting

 Performing a Post Mortem on SQL Server

**Microsoft .Net 4.0 Framework**

**Language : C#**

**Duration: 4 Days Methodology: Hand-On**

**Day 1**

 Windows DNA

 Windows DNA and its related problems

 .NET Solution

 **.NET Framework Architecture**

 Framework Elements

 Application Domains

 Elements of CLR

 JITer

 Garbage Collector

 Class loader

 Security Manager

 Type Verifier

 Compilation Process

 Compilation process in .NET

 .NET Application Execution Process

 .NET Assemblies

 **Type System**

 Understanding .NET Type system

 Common Type System

 Value types and Reference types

 Common Language Specification

 CLS compliance across .NET languages

 .NET Namespaces

 Significance of Namespaces in .NET

 Understanding Namespaces

 Namespace extended across assemblies

 Multiple Namespaces within an assembly

 Nesting Namespaces

 .NET Framework Features

 Boxing and Unboxing

 Checked / UnChecked

 params Keyword

 out, ref keyword

 const and readonly keyword

 static members

 functions and constructor

 Object Oriented Features

 Encapsulation

 Object type

 Defining Properties

 Read write

 Read-only

 Write only

 Accessibility levels

 Defining Methods

 Using access modifiers

 Initializations and cleanup using Constructors, Destructors

 Importance of IDisposable

 Static and instance members

 Inheritance and Polymorphism

 Invoking base class functionality from derived class

 Defining virtual functions and overriding them

 Abstract classes and methods

 Sealed classes and methods

 Hiding base class members

 Partial Class

 Static classes

**Day 2**

 Delegates

 Need for callbacks in an application

 Introducing Delegates

 Multicast delegates

 Anonymous methods

 Delegate inferring

 Events

 Notifications using events

 .NET Events instances of delegates

 Events and inheritance

 Register and Un-register event

 **Exception Management**

 Understanding exceptions

 Handling exceptions

 Exception class

 User defined exceptions

 Debugging

 Visual Studio .NET debugging environment

 Using debugging tools provided by the IDE

 Local Window

 Watch window

 Breakpoints

 Call stack

**Day 3**

 Interfaces

 Defining Interfaces in .NET

 Implementing interfaces

 Providing interface specific implementation

 Querying for a supported interface

 Collection Classes

 What are collection classes

 .NET Collection Types

 Name-Value pair Collections

 Generics

 Method with Generics

 Class with Generics

 Constraints on generic methods and classes

 Delegate with Generics

 Iterators

 Creating Iterators for a class

 Iterators with Generics and Non Generics class

 Using Interfaces IEnumerable, Ienumerator

 Nullable Data Types

 Value types that hold Null values

 INullable generic type

 Using the shortcut to declare nullable types

 Assemblies

 Private and Shared Assemblies

 Creating Strong-named assemblies

 Global Assembly Cache

 Single file assemblies and Multi file assemblies

 .NET modules

 Understanding the assembly contents using ILDASM

 Exception Handling

 Handling Data Errors

 Concurrency exceptions and their management

**Day 4**

 Reflection

 Using Reflection for RTTI

 Using Reflection for RTOC

 Accessing assembly metadata

 Accessing type metadata

 File I/O

 Using the different stream classes in .NET

 BinaryReader/Writer

 StreamReader/Writer

 FileInfo/DirectoryInfo

 Serialization

 Persisting object to a stream

 Serialization and inherited classes

 Serialization modes

 Binary

 SOAP

 XML

 Customizing Serialization by

 Using attributes

 Implementing ISerializable interface

 Introduction to New Enhansment in C# 3 till C#7

 Start using New features for development and learning from hear on.

 C# new Features

 Using static keyword in using statement

 String interpolation

 Dictionary Initializers

 Auto property Initializers & Getter only auto properties

 name Of Operator

 Await in catch/finally

 Null conditional operator and Null Propogation

 Lambda functions

 Expression bodied members

 Using static with Extension Methods

 Exception filters

 File I/O and Serialization

 Persisting object state to a stream

 Various classes used for File handling

 Using StreamReader, StreamWritter

 Using BinaryReader, BinaryWriter

**ADO.Net 4.5 and Linq and Entity Framework**

**Program Duration**: 2 Days.

**Table of contents:**

 ADO.NET Architecture

o .NET Data Providers

o DB Connectivity Architectures in .NET

o Elements of .NET Data Providers

o Introduction to SQL Server

o Namespaces in ADO.NET

o Using server exlorer window

o Connection class

o Command class

o Direct Command execution against database

o Using Parameters in command

o Performing CRUD operations

o Connected Vs disconnected Architecture

o Data reader class

o The dataset and dataset Architecture

o Comparison ADO & ADO.NET on Disconnected Data architecture

o Implementing Disconnected Data Architecture

o Performing CRUD operations in disconnected architecture

o ReadXml(),ReadXmlSchema(), WriteXml(), WriteXmlSchema() usage.

o Implementing Data Access Layer

o Guidelines for Designing DAL

**Linq and Entity Framework**

**Table of contents:**

 Language Integrated Query

o Introduction , LINQ Syntax

o Query Operators

o Select, from, Where

o ofType

o OrderBy

o ThenBy

o GroupBy, into

o Select

o SelectMany

o Take, TakeWhile

o First

o FirstOrDefault

o Single

o SingleOrDefault

o Aggregate functions Sum, Min, Max, Average, Count

o Distinct

o Intersect

o Except

o Join

o LINQ projection

o Deferred execution vs immediate execution

o Let keyword

o LINQ to Object

o LINQ to DataTable

 Entity Framework

o Overview of ORM Products

o Entity Framework introduction

o Using Database first Approach

o Using Model First approach

o Using Code First approach

o Using LINQ to Entities to perform CRUD operations

**ASP.NET MVC**

**Program Duration**: 4 Days.

**Table of contents:**

 Fundamentals of ASP.NET MVC 5

o ASP.NET MVC Framework

o ASP.NET vs. ASP.NET MVC

o ASP.NET MVC Model

 Exploring Controllers

o Working with Controllers

o Routing

o Attribute Routing

o Action Methods

o Passing Data from controller to view

 Working with Views

o ASP.NET MVC Razor View Engines

o HTML Helpers

o Working with Layout

 Exploring Models & working with data

o Model Binding

o Data Annotation

 Entity Framework 6 Features with Asp. Net Scaffolding

o Code First Model

o Multiple Migrations

o Scaffolding

 Web Optimization with ASP.NET MVC

o Web Optimization

o Bundling & Minification

o Configuring bundles

 Using AJAX In ASP.NET MVC

o Introduction to Ajax

o AJAX in ASP.NET MVC

o AJAX Helpers

o Partial Page rendering

o AJAX using jQuery

 Introduction to SignalR

**ASP.NET Web API**

**Table of contents:**

 Introduction to ASP.NET Web API

o Web API features

o SOAP- based WebServices

o HTTP Web Services

o Web API Introduction

o Web API Routing

o Web API Parameter Biding

o Content Negotiation

o Working with swagger and Postman Utility

=====================================================================

**ASSESSMENT PROJECT – 2 days**

=====================================================================

**Development Design Patterns**

**Framework: .Net Duration: 2 days**

**Target Audience: Mixed Methodology: Classroom / Hands-on Training**

 Software development trends

 **Challenges in software development and Maintenance phase**

 Introduction to design pattern

 What are design patterns?

 Classification of design pattern

 Solving the problem using design pattern

 Overview of different design patterns used in different areas

**Working and Practical Implementation with demonstration of the following Design Pattern**

Creational Design Patterns

 Factory Method

 Abstract Factory

 Prototype

 Builder

 Singleton

Structural Patterns

 Adapter

 Bridge

 Decorator

 Proxy

 Composite

 Façade

Behavioral Design Patterns

 Chain of Responsibility

 Command

 Memento

 Iterator

 Visitor

 Mediator

 Observer

 State

 Strategy

Anti-Patterns

 **What are Anti-Patterns?**

 Reference Model

 **Causes of anti-patterns**

 Sequential Coupling

 Novel Notations

 Base Bean

 Metaphor mismatch

 God Object

**CASE Study & Project- 2 days**