

CELEBRATING
12 YEARS

Quality Thought®



Complete
.NET
Framework

Full Stack



Client Side Technologies

- ⇒ Html5
- ⇒ CSS
- ⇒ Javascript
- ⇒ Bootstrap

C# Language (*Upto C#11)

- ⇒ C#.Net
- ⇒ Language Fundamentals
- ⇒ OOPS
- ⇒ Strings
- ⇒ Exception Handling
- ⇒ Multithreading
- ⇒ Class Library
- ⇒ IO
- ⇒ Collections
- ⇒ DataAnnotations
- ⇒ Regular Expressions

RDBMS

- ⇒ Normalization
- ⇒ SQLServer

Database Library

- ⇒ ADO.Net
- ⇒ EF.Net (SQLServer, MySql and Oracle)
- ⇒ Linq
- ⇒ N-Tier Architecture - Application

Database Library

- ⇒ ADO.Net
- ⇒ EF.Net (SQLServer, MySql and Oracle)
- ⇒ Linq
- ⇒ N-Tier Architecture - Application

Web Application Framework Library .Net Framework based*

- ⇒ ASP.Net
- ⇒ ASP.Net MVC
- ⇒ ASP.Net Web API

.Net Core Framework based

- ⇒ .Net Core
- ⇒ ASP.Net Core
- ⇒ ASP.Net MVC Core
- ⇒ ASP.Net Web API Core - REST
- ⇒ EF.Net Core
- ⇒ Other Important Topics

Tools

- ⇒ Postman, Fiddler

IDE

- ⇒ Visual Studio 2019,2022 and Visual Studio Code

Server

- ⇒ IIS

Architecture & Development

- ⇒ Design Patterns

Testing

- ⇒ Nunit and Others

Client Side Javascript Frameworks

- ⇒ node introduction
- ⇒ Angular or React

Repository

- ⇒ Git

Cloud - Application Level*

- ⇒ Use of different resources of Azure
- ⇒ Deployment of .Net Web Applications on Azure
- ⇒ CI & CD Pipelines using Azure Repos
- ⇒ Deployment of .Net App using Docker
- ⇒ Deployment of .Net with Kubernetes - AKS

.Net Core - ASP.Net Core ASP.Net MVC Core with EF.Core

- ⇒ Introduction to .Net Core
- ⇒ Net Core Templates
- ⇒ Asp.Net Core Templates
- ⇒ Advantages of .Net core
- ⇒ Asp.Net MVC Core Template
- ⇒ Folders for Models, Controllers, Views, wwwroot, css, js, lib
- ⇒ Architecture, Model, View, Controller
- ⇒ ConfigureServices
- ⇒ Middleware
- ⇒ Dependency injection in .Net core
- ⇒ Hierarchy of middleware
- ⇒ UseDeveloperExceptionPage
- ⇒ UseAuthentication
- ⇒ UseMvc, UseExceptionHandler
- ⇒ Profiles
- ⇒ Environment variables in launchSettings.json
- ⇒ DbContext, AddMvc, Singleton, AddTransient, AddScoped
- ⇒ Routing - Endpoints
- ⇒ Controllers - Actions - Views
- ⇒ _Layout, ViewStart files
- ⇒ Razor engine rendering
- ⇒ use of @ symbol
- ⇒ RenderBody, RenderSection, Scripts
- ⇒ HTML helpers & Tag helpers
- ⇒ Data Annotations - Validations
- ⇒ PartialView
- ⇒ Logging
- ⇒ Filters
- ⇒ Database programming with EF.Core
- ⇒ CRUD Operations
- ⇒ Working and Customizing Scaffolding Views

Asp.Net Core Web Api - REST

- ⇒ Introduction
- ⇒ REST web service, Web API & Microservice
- ⇒ Inside HttpRequest HttpResponseMessage
- ⇒ Action Verbs - Types
- ⇒ HttpStatusCodes in WebAPI
- ⇒ Create First REST Web API
- ⇒ Configuration and Configuration files in Web API
- ⇒ Swagger - Installation - Usage
- ⇒ Postman - for all types of request types with data and authorization

- ⇒ Routes - Practices
- ⇒ Filters - Practices
- ⇒ CRUD operations using Web API with EF.Core
- ⇒ What is CORS? - CORS enablement
- ⇒ Security in WebAPI
- ⇒ JWT - Usage in Web API
- ⇒ Using REST Web API by React Application

Asp.Net AWhat is MVC? , Why MVC?with C#

- ⇒ Asp.Net MVC
- ⇒ Introduction, background and difference w.r.t. ASP.Net Webforms
- ⇒ MVC request pipeline
- ⇒ Model, View, Controller
- ⇒ Controller actions
- ⇒ MVC with Database
- ⇒ Model Binding
- ⇒ Layouts
- ⇒ Partial Views
- ⇒ Razor Engine
- ⇒ Data Annotations
- ⇒ URL Routing
- ⇒ Model Validation server and client side
- ⇒ Session Handling
- ⇒ Exception Handling
- ⇒ Security
- ⇒ Interview Questions

Asp.Net WebAPI

- ⇒ Creating/ Configuring Web API project
- ⇒ Testing Web API
- ⇒ Web API Controller
- ⇒ Parameter Binding
- ⇒ Action Return type
- ⇒ Data Formats
- ⇒ Media Type Formatters
- ⇒ Web API Filters
- ⇒ Web API for CRUD
- ⇒ GetMethod, PostMethod, PutMethod, DeleteMethod
- ⇒ Consuming Web Api
- ⇒ Http Client
- ⇒ Dependency Injection
- ⇒ WebAPI Hosting
- ⇒ Interview Questions

RDBMS

- ⇒ Normalization
- ⇒ Importance - Redundancy , Inconsistency , Performance
- ⇒ Types of Normalforms
- ⇒ Keys
- ⇒ First, 2nd , 3rd Normal
- ⇒ Practices

SQL Server

- ⇒ Management Studio
- ⇒ Tables - Cols - Datatypes - Constraints
- ⇒ DDL - Create , Drop, Alter
- ⇒ DML - Insert , Update, Delete
- ⇒ Select - Complete Structure
- ⇒ Subqueries
- ⇒ Joins
- ⇒ Functions
- ⇒ Procedures
- ⇒ Triggers
- ⇒ Indexes
- ⇒ Transactions
- ⇒ Practices

Ado.Net

- ⇒ Namespaces
- ⇒ Connected Architecture
- ⇒ Disconnected Architecture -Dataset
- ⇒ CRUD Operations using above

EF.Net

- ⇒ ORM - Framework
- ⇒ Model
- ⇒ Mapping
- ⇒ Data access layer
- ⇒ CodeFirst - with existing database
- ⇒ CodeFirst - with non existing database
- ⇒ Seeding - Database Initialization
- ⇒ Database First
- ⇒ Data Annotations in EF
- ⇒ Regular Expressions in EF
- ⇒ FluentAPI
- ⇒ Layered Architecture
- ⇒ CRUD Operations

Introduction to Frameworks

- ⇒ .Net Framework
- ⇒ .Net Standard
- ⇒ .Net Core
- ⇒ .Net 5, 6, 7

Introduction to Types of Applications
Introduction to Visual Studio 2019/2022
Net vs Other Technologies

C# Language Fundamentals - C#10.0

- ⇒ Versions and Frameworks
- ⇒ Basic Structure of Console Application
- ⇒ using statement
- ⇒ Datatypes
- ⇒ Primitive datatypes - Value Type - Predefined
- ⇒ Abstract datatypes - Reference Type - Userdefined
- ⇒ Advanced datatypes
- ⇒ Type safety
- ⇒ Variables declaration , initialization
- ⇒ Basic input and output methods
- ⇒ Type casting
- ⇒ Control Structures
- ⇒ if, if..else, if..else if...else
- ⇒ switch
- ⇒ do..while, while, for
- ⇒ foreach
- ⇒ Operators
- ⇒ Strings
- ⇒ Arrays
- ⇒ Interview Questions

Ops using C#

- ⇒ Principles of OOps
- ⇒ Encapsulation - Datahiding - Abstraction
- ⇒ Inheritance - Reusability - Extensibility
- ⇒ Polymorphism - Overloading - Overriding
- ⇒ Access Specifiers - Modifiers
- ⇒ Full Syntax to create a class
- ⇒ Full Syntax to create a variable
- ⇒ Full Syntax to create a method
- ⇒ Class vs Structure
- ⇒ Concrete Class
- ⇒ Constructor, Destructor
- ⇒ Object - Object , Collection Initializers
- ⇒ Finalizer

- ⇒ Static
- ⇒ Abstract
- ⇒ Sealed
- ⇒ Virtual
- ⇒ Partial
- ⇒ Override
- ⇒ Const , Readonly
- ⇒ ref, out parameters
- ⇒ Var variable
- ⇒ unsafe
- ⇒ class with properties
- ⇒ Indexer
- ⇒ Iterator
- ⇒ Async - await
- ⇒ Interfaces - Importance - Full Abstraction
- ⇒ Anonymous method , class
- ⇒ Interview Questions

Exception Handling - Compiler - Syntax - Runtime - Logical

- ⇒ try..catch..finally structure
- ⇒ Predefined exceptions
- ⇒ Userdefined exceptions
- ⇒ throw
- ⇒ Interview Questions

Delegates

- ⇒ What is delegate
- ⇒ Types of delegates - Single , Multicast
- ⇒ Delegates with Named vs. Anonymous Methods
- ⇒ Interview Questions

Multithreading

- ⇒ Creating and Managing Threads
- ⇒ Thread Life Cycle - Pause , Interrupt, Destroy
- ⇒ ThreadPriority - State - Storage - Data
- ⇒ Synchronization
- ⇒ Interview Questions

Class Library - Namespace - DLL - Componentization

- ⇒ and SortedDictionary
- ⇒ Concurrency
- ⇒ Interview Questions

Linq

- ⇒ Linq structure
- ⇒ Write LINQ queries in C#
- ⇒ Using Where , Group by , Order by
- ⇒ Subquery
- ⇒ joins
- ⇒ Practices
- ⇒ Interview Questions

Lambda expressions

- ⇒ anonymous function
- ⇒ Expression lambda
- ⇒ Statement lambda
- ⇒ Practices
- ⇒ Interview Questions

IO

- ⇒ Reader, Writer
- ⇒ StreamReader,StreamWriter
- ⇒ File ,Directory - Related library
- ⇒ Exceptions

Data Annotation

- ⇒ Validations
- ⇒ Mapping
- ⇒ Best practices
- ⇒ Interview Questions

Regular Expression

- ⇒ Character Escapes, Character Classes
- ⇒ Anchors
- ⇒ Grouping Constructs
- ⇒ Quantifiers, Backreference Constructs
- ⇒ Alternation Constructs
- ⇒ Best practices for regular expressions
- ⇒ Interview Questions

STUDENT TRANSFORMATION STAGES



OUR STUDENTS ARE PLACED IN
