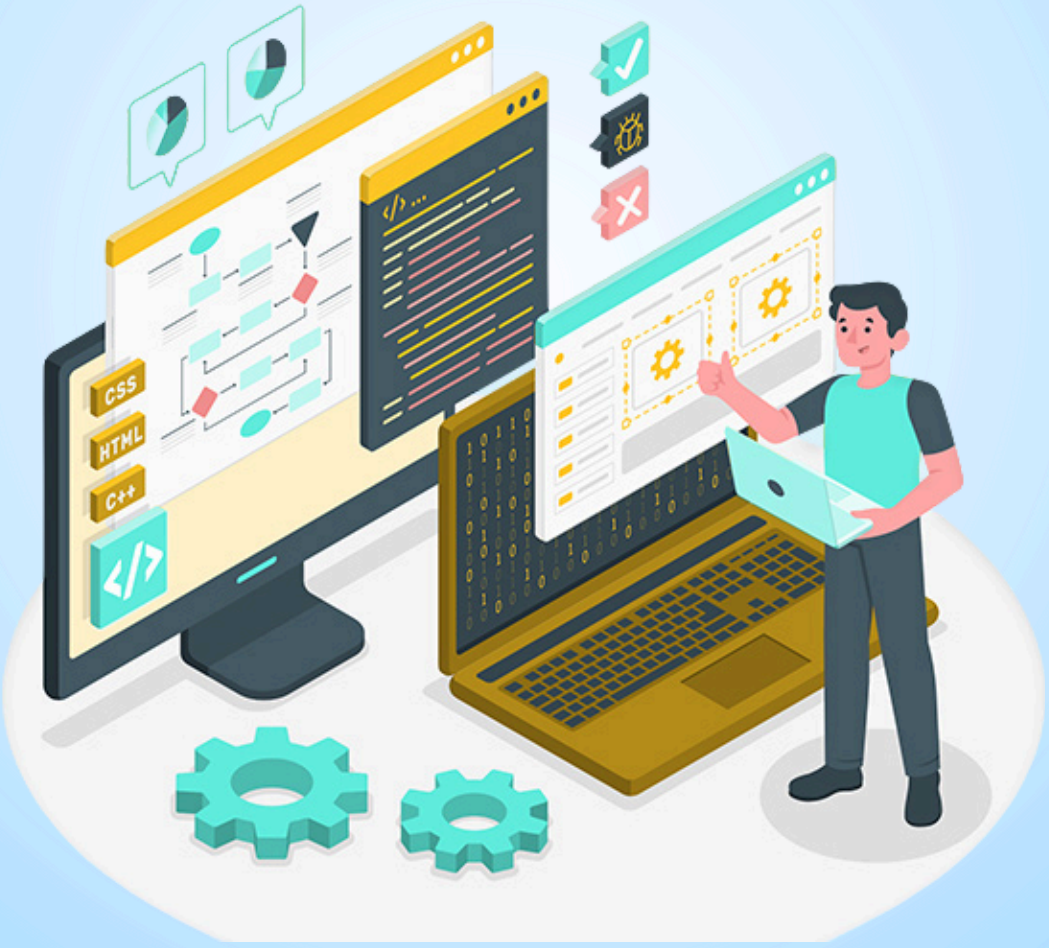
.NET FULL STACK

COURSE CURRICULUM



ENROLL NOW

CALL US
+91 630 171 9270

1.Introduction to C#
• What is C#?

2. Basics of C#
• Data Types and Variables
• Constants and Literals
• Keywords and Identifiers
• Type Conversion and Casting
• Nullable Types

3. Control Flow
• If-Else Statements
• Switch Statements
• Loops: for, while, do-while, foreach
• Break, Continue, and Goto Statements

4. Object-Oriented Programming
• Classes and Objects
• Encapsulation
• Inheritance
• Polymorphism
• Abstraction
• Constructors and Destructors
• Static Classes and Members

5. Advanced OOP Concepts
• Abstract Classes
• Interfaces
• Sealed Classes
• Partial Classes
• Indexers and Properties

6. Exception Handling
• Try, Catch, Finally
• Throwing Exceptions
• Custom Exceptions
• Exception Filters

7. Delegates and Events
• What is a Delegate?
• Single-Cast and Multi-Cast Delegates
• Anonymous Methods
• Lambda Expressions
• Events and Event Handlers

8. Collections
- Arrays
- List, Dictionary, HashSet, Queue, Stack
- Generic Collections (List<T>, Dictionary<TKey, TValue>)
- Non-Generic Collections
- Collection Initializers

9. Multithreading and Asynchronous Programming
- Introduction to Threads
- Thread Pooling
- Tasks and Task<T>
- async and await
- Parallel Programming

10. Memory Management
- Garbage Collection
- Dispose Pattern and Finalizers
- IDisposable and Using Statement
- Weak References

11. Reflection
- What is Reflection?
- Metadata and Assemblies
- Dynamic Type Creation
- Attributes and Reflection

12. Dependency Injection
- What is Dependency Injection?
- Constructor Injection
- Method Injection
- Property Injection

13. Assemblies and Namespaces
- Assemblies in .NET
- Private and Shared Assemblies
- Namespaces
- using Directive

JavaScript Topics

1. Introduction to JavaScript
- What is JavaScript?

2. Basics of JavaScript
- Syntax and Structure
- Variables (var, let, const)
- Data Types (String, Number, Boolean, Object, Array)
- Operators (Arithmetic, Logical, Comparison)

3. Control Flow
- Conditional Statements (if-else, switch)
- Loops (for, while, do-while, for-in, for-of)
- Error Handling (try-catch, finally)

4. Functions in JavaScript
- Function Declaration and Expression
- Arrow Functions
- Callback Functions

5. Object-Oriented JavaScript
- Objects and Properties
- Prototypes and Prototype Inheritance
- Classes and Constructors
- Encapsulation
- Polymorphism

6. DOM (Document Object Model) Manipulation
- Selecting Elements (getElementById, querySelector)
- Modifying Elements (innerHTML, styles, attributes)
- Event Handling (addEventListener)
- Creating and Removing Elements

7. JavaScript Events
- Event Types (Click, Submit, Mouse Events)
- Event Bubbling and Capturing
- Event Delegation

8. JavaScript and AJAX
- Introduction to AJAX
- Fetch API
- XMLHttpRequest
- Promises and Async/Await

MVC (Model-View-Controller)

1. Introduction to MVC
- What is MVC?

2. MVC Architecture Components
- Model: Represents Data and Business Logic
- View: Represents UI (User Interface)
- Controller: Manages User Input and Interaction

3. MVC in .NET
- Overview of MVC in ASP.NET Core
- How ASP.NET Core Implements MVC
- Folder Structure in MVC Projects (Models, Views, Controllers)

4. Routing in MVC
- URL Routing in MVC
- Attribute Routing
- Route Parameters and Constraints
- Custom Route Configurations

5. Controllers in MVC
- Creating Controllers
- Action Methods
- Returning Views and Data
- Controller Lifetimes and Dependency Injection

6. Views in MVC
- Razor View Engine
- Creating Views (HTML, Razor Syntax)
- ViewData and ViewBag
- Strongly Typed Views (Model Binding in Views)
- Partial Views

7. Models in MVC
- What is a Model?
- Creating Models (Plain C# Classes)
- Model Binding
- Data Annotations and Validation
- Using ViewModels

8. Data Binding in MVC
- Model Binding in Views
- Passing Data to Views (ViewData, ViewBag, TempData)
- Bindable Collections (List, Dictionary, etc.)

9. MVC Forms and Validation
• Creating Forms in Views
• Handling Form Submissions
• Validating Forms (Data Annotations, Custom Validation)
• Client-Side and Server-Side Validation

10. Action Filters and Middleware
• What are Action Filters?
• Built-in Action Filters (e.g., Authorize, ValidateAntiForgeryToken, etc.)
• Custom Action Filters
• Middleware in ASP.NET MVC

11. Authentication and Authorization in MVC
• Forms Authentication
• Cookie-based Authentication
• Role-based Authorization
• Using Identity Framework for Authentication

12. Dependency Injection in MVC
• Introduction to Dependency Injection in MVC
• Using DI in Controllers
• Constructor Injection vs. Property Injection
• Registering Services in Startup Configuration

13. MVC Scaffolding
• What is Scaffolding in MVC?
• Creating Models, Views, and Controllers with Scaffolding
• Customizing Scaffolded Views

14. Razor Pages in MVC
• What are Razor Pages?
• Difference between Razor Pages and MVC Views
• Using Razor Pages in ASP.NET Core MVC

15. AJAX in MVC
• Introduction to AJAX in MVC
• Partial Views with AJAX
• Using jQuery and AJAX for Asynchronous Updates
• Handling AJAX Requests in Controllers

Web API Topics

1. Introduction to Web API
- What is a Web API?
- Difference between Web API and Web Services
- HTTP Methods (GET, POST, PUT, DELETE)

2. HTTP Protocol in Web API
- HTTP Request and Response Structure
- HTTP Methods and Status Codes
- Headers and Body in HTTP Requests
- Request and Response Formatting (JSON, XML)

3. Creating a Web API in .NET
- Setting Up a Web API Project in ASP.NET Core
- Understanding Startup.cs and Program.cs
- Creating Controllers and Action Methods
- Routing and URL Mapping in Web API
- Action Filters and Middleware

4. Routing in Web API
- Attribute Routing vs Conventional Routing
- Route Parameters and Query Strings
- Custom Route Templates
- Route Constraints

5. RESTful Principles in Web API
- Stateless Communication
- Resource Identification through URIs
- Standard HTTP Methods
- Representations of Resources (JSON/XML)

6. Authentication and Authorization in Web API
Authentication Methods (Basic, Token-Based, OAuth)
- Implementing JWT (JSON Web Tokens) Authentication
- Role-based Authorization in Web API
- Token Validation and Expiration

7. Error Handling in Web API
- Standardizing Error Responses
- Exception Handling in Web API
- Using ProblemDetails for Error Responses
- Custom Error Responses and Status Codes
- Global Error Handling with Middleware

8. Web API Security
• Cross-Origin Resource Sharing (CORS)
• Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) Protection
• Rate Limiting and Throttling APIs

9. API Documentation
• Generating API Documentation with Swagger/OpenAPI
• Documenting Models and Parameters
• Using Annotations for Swagger UI Documentation

10. Async and Performance in Web API
• Asynchronous Programming in Web API (async/await)
• Performance Optimization Techniques (Caching, Compression, Pagination)
• Managing Large Responses with Streaming

MSSQL (Microsoft SQL Server) Topics

1. Introduction to MSSQL
• What is SQL Server?
• SQL Server Architecture (SQL Server Instance, Databases, Tables)

2. Database Design
• Normalization (1NF, 2NF, 3NF)
• Denormalization
• Data Types in SQL Server
• Designing Tables (Primary Keys, Foreign Keys, Constraints)
• Indexing (Clustered and Non-Clustered)

3. SQL Queries
• SELECT Statement (Basic and Advanced Queries)
• Filtering Data (WHERE, BETWEEN, IN, LIKE, etc.)
• Sorting and Grouping (ORDER BY, GROUP BY)
• JOIN Operations (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN)
• Subqueries and Correlated Subqueries

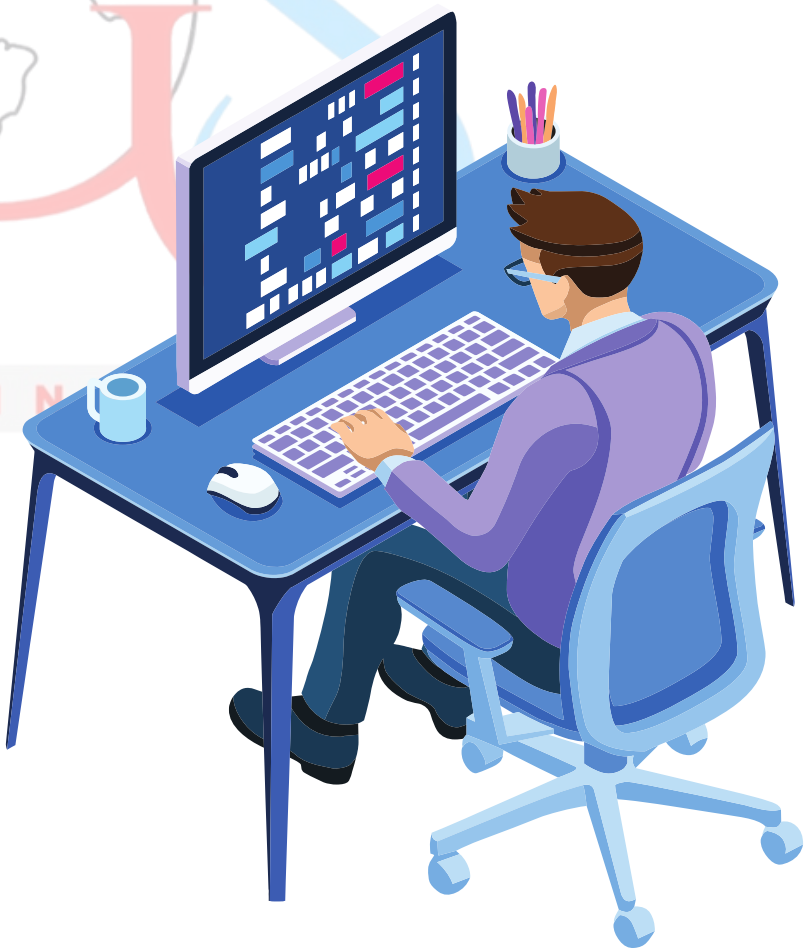
4. Advanced SQL Queries
• Using Views
• Common Table Expressions (CTE)
• Window Functions (ROW_NUMBER, RANK, etc.)
• Aggregate Functions (COUNT, SUM, AVG, MAX, MIN)
• String and Date Functions (CONCAT, DATEPART, GETDATE, etc.)

5. Stored Procedures and Functions

- Creating Stored Procedures and Functions
- Input and Output Parameters
- Executing and Calling Stored Procedures • Error Handling in Stored Procedures
- Using Scalar and Table-Valued Functions

6. SQL Server Data Types

SQL Server Basic Data Types (INT, VARCHAR, DATETIME, etc.)
Working with Date and Time Data Types
Advanced Data Types (XML, JSON, Spatial Data, HierarchyID)
Handling Large Data Types (TEXT, IMAGE, XML, FILESTREAM



Thank you

INFOTEKH

Thank You for Going Through Dot Net Full Stack Curriculum
We hope this guide has provided a clear and structured learning path to strengthen your skills in Dot Net Full Stack .

📌 **NEXT STEPS**

- Start practicing with real-world use cases and hands-on exercises
- Build personal or client-based projects for your portfolio
- Keep exploring updates and best practices in the industry
- Join discussions and stay connected with the community

📞 Need Help or Guidance?

Feel free to contact our course support team:

Course Coordinator

GS Infotekh

📧 **contact@gsinfotekh.com**

🌐 **www.gsinfotekh.com**

📞 **+91 630 171 9270**