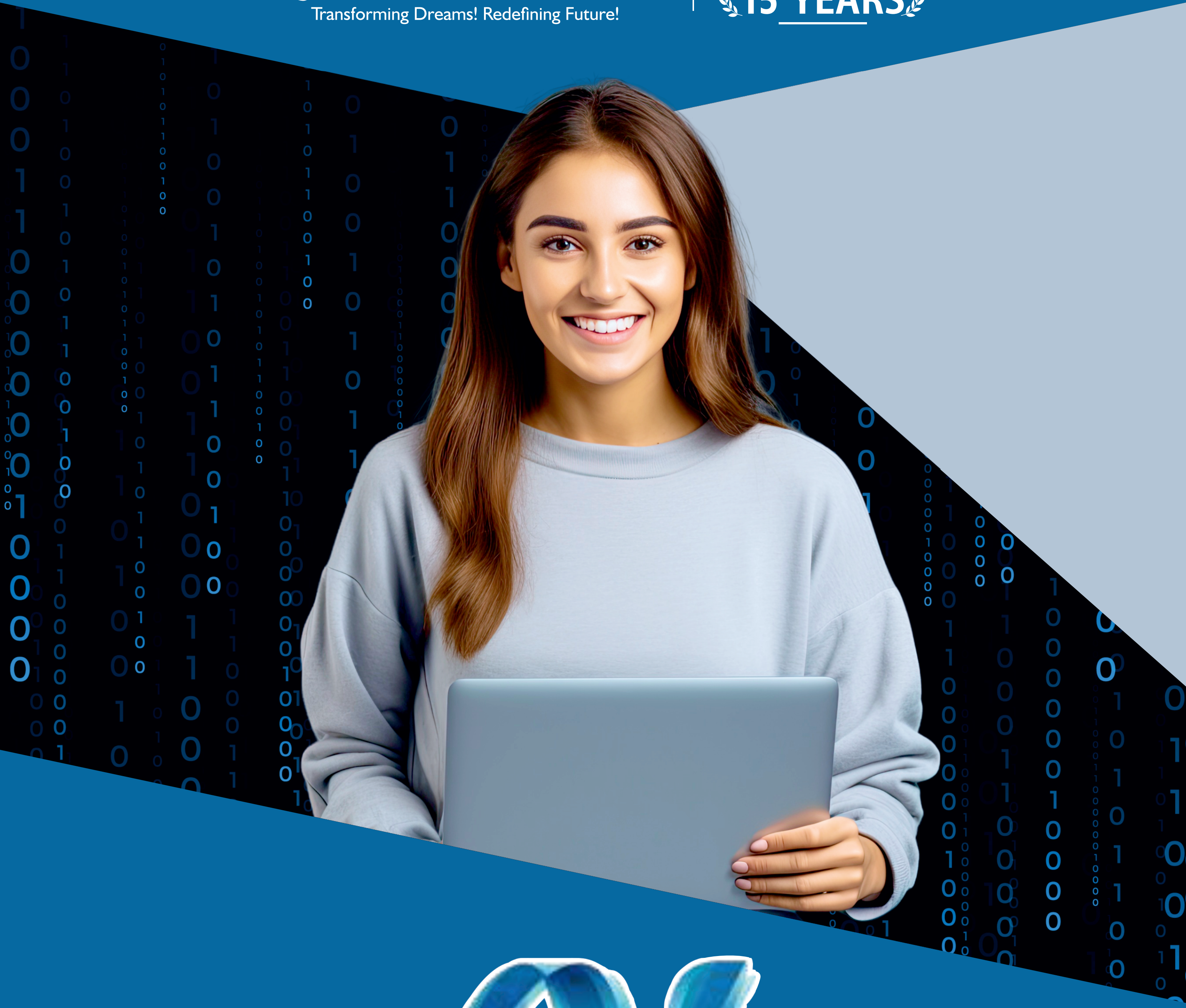
QualityThought®

Transforming Dreams! Redefining Future!

CELEBRATING 15 YEARS



.NET Core Full Stack

with Angular & Azure DevOps

Introduction to .NET Framework 4.8

- What is .NET Platform?
- What is .NET Framework
- .NET Framework, Languages, and Tools
- .NET Framework Major Components
- Common Language Runtime (CLR)
- The CLS (Common Language Specification)
- The CTS (Common Type System)
- Value Types and Reference Types
- Compilation and Execution in .NET
- Understand the .NET Framework 4.8 stack

Introduction to .NET Core – Net8

- .NET Core – Overview
- Characteristics of .NET Core
- The .NET Core Platform
- .NET CORE architecture and Advantages
- Build and run Cross platform apps
- .NET Core – Environment Setup
- .NET Core – Code Execution
- IoC Container & Middleware
- .NET Core – Modularity
- .NET Core – Project Files
- IIS Self Hosting & different cross platform deployments
- Microservices using .NET Core
- .NET Core – Windows Runtime and Extension SDKs.
- .NET Core – Create .NET Standard Library.
- Comparison between .NET Framework & .NET Core
- Introduction to Dependency Injection

Introduction to C#

- Features of C#
- C# Compilation and Execution
- General Structure of a C# Program

Data Types and Arrays in C#

- Data Types in C#
- Value Types and Reference Types
- Boxing and UnBoxing
- Single Dimensional, Multi-Dimensional & Jagged arrays
- Nullable Types
- Implicitly Typed Local variables
- Var vs dynamic
- Is and as operator
- Ref vs out keywords
- The 'object' base class in .net
- Equals() vs ==
- String vs StringBuilder
- Various String class methods
- Default parameters, named parameters
- Parse() vs TryParse() vs Convert Class methods

Debugging in C#

- Various Types of .NET Projects
- Tracing, Debugging, Build
- Compile Options
- Using break points
- Using break conditions
- Debugging Exception
- Using watch and output window
- What are Diagnostics?
- Debug and Trace Classes
- Creating multiple projects within one solution
- Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings
- Using watch and output window
- Creating multiple projects within one solution
- Customizing Visual Studio Settings – Extensions, NUGet Package, Environmental Settings

OOP with C#

- Structures and enums
- The architecture of a class in C#
- Instance, Class & Reference variables
- Access Modifier
- Abstract Classes
- Constructors, Destructors, The GC
- .NET Base class library
- Inheritance in C#
- Method Overloading
- Method Overriding
- Operator Overloading
- Method Hiding

- Access modifiers : private, pubic, protected, internal, protected internal, new
- Anonymous types
- Abstract classes
- Sealed classes
- Creating Interfaces
- Implementing Interface inheritance
- Declaring properties within Interfaces
- Namespaces
- Creating and using Generic classes
- Indexers & Properties
- Auto Implemented properties
- Static Classes
- Property Accessors
- Partial types
- Extension methods
- Object Initializer

Evaluating Regular Expressions in C#

- RegEx Class
- Forming Regular Expression
- Methods for Regular Expression
- Exception Handling
- Exceptions in C#
- Exception class hierarchy
- Try block
- Multiple catch blocks
- Finally block
- Purpose of throw keyword
- Purpose of inner exception
- Creating Custom Exception

Garbage Collection in C#

- Role of a Garbage Collector
- Garbage Collection Algorithm
- Finalize vs Dispose
- Collections & Generics
- System.Collections Namespace
- Collection Interfaces
- Collection Classes
- The collection API
- Working with Generics
- Creating Generic class, Generic Methods, Interfaces, Delegates
- Collection Initializers
- Iterators
- IEnumerable, IEnumerator, Icomparor interfaces
- Constraints

Anonymous Types, Delegates, Events & Lambda

- Extension Methods
- Anonymous Type
- Var and Dynamic
- Introduction to Delegates
- Events in C#
- Anonymous Methods
- Lambda Expression
- Expression Tree

File I/O and Serialization

- Using StreamReader, StreamWritter
- Using BinaryReader, BinaryWriter
- Using File, FileInfo, Directory, DirectoryInfo
- Serialization modes: SOAP, XML
- JSON serialization

Introduction To Reflection and Attributes

- What is Reflection?
- Attributes.
- Pre-defined Attributes
- Custom Attributes.

Threading, Parallel and Async programming with C#

- Task Parallel Library
- Threads Vs. Tasks
- Thread state
- Task Based Asynchronous Model
- Async and Await
- Using Locks

Packaging and Deployment

- File System Editor
- Registry Editor
- File Types Editor
- User Interface Editor
- Custom Actions
- Launch Condition Editor
- Creating Uninstall Shortcut

New Features in C# 10.0

- ➪ Record structs
- ➪ Improvements of structure types
- ➪ Interpolated string handlers
- ➪ global using directives
- ➪ File-scoped namespace declaration
- ➪ Extended property patterns
- ➪ Improvements on lambda expressions
- ➪ Allow const interpolated strings
- ➪ Record types can seal ToString()
- ➪ Improved definite assignment
- ➪ Allow both assignment and declaration in the same deconstruction
- ➪ Allow AsyncMethodBuilder attribute on methods
- ➪ CallerArgumentExpression attribute
- ➪ Enhanced #line pragma
- ➪ Warning wave 6

C# 11 Features

- ➪ Generic attributes
- ➪ UTF-8 string literals
- ➪ Newlines in string interpolation expressions
- ➪ List patterns
- ➪ File-local types
- ➪ Required members
- ➪ Auto-default structs
- ➪ Pattern match Span<char> on a constant string
- ➪ Extended nameof scope
- ➪ ref fields and scoped ref
- ➪ Warning wave 7

C# 12 Features

- ➪ Primary constructors
- ➪ Collection expressions
- ➪ Inline arrays
- ➪ Optional parameters in lambda expressions
- ➪ ref read only parameters
- ➪ Alias any type
- ➪ Experimental attribute
- ➪ Interceptors

Automated Testing with MSTest and Nunit

- ➪ Using Asserts to Pass or Fail Tests
- ➪ Controlling and Customizing Test Execution
- ➪ Creating Data Driven Tests
- ➪ Reducing Code Duplication and Increasing Test Readability
- ➪ Writing Your First NUnit Test
- ➪ Understanding NUnit Tests
- ➪ Asserting on Different Types of Results
- ➪ Controlling Test Execution
- ➪ Creating Data Driven Tests and Reducing Test Code Duplication

DevOps Concepts

- ➪ Introduction to DevOps :
- ➪ What is DevOps
- ➪ Evolution of DevOps
- ➪ Agile Methodology
- ➪ Why DevOps
- ➪ Agile vs DevOps
- ➪ DevOps Principles
- ➪ DevOps Lifecycle
- ➪ DevOps Tools
- ➪ Benefits of DevOps
- ➪ Continuous Integration and Delivery pipeline

Cloud Azure

1. Azure
   - ➪ Agile and Scrum
   - ➪ Cloud Computing

2. Version Control & Code Management:
   - ➪ Git
   - ➪ Azure Repos

3. Continuous Integration and Continuous Delivery (CI/CD):
   - ➪ Azure Pipelines: Designing & implementing CI/CD pipelines for automated builds, testing, and deployment.
   - ➪ Build Automation
   - ➪ Release Management: Managing deployments, releases, and environments.

4. Azure DevOps Tools:
   - ➪ Azure Boards
   - ➪ Azure Test Plans
   - ➪ Azure Artifacts

5. Infrastructure as Code & Automation
   - ➪ Infrastructure as Code (IaC)
   - ➪ Configuration Management

   - ➪ Automation Tools: Using tools like Terraform to manage infrastructure.
6. Testing & Quality:
   - ➪ Unit Testing
   - ➪ Code Coverage
   - ➪ Testing Strategies

7. Security & Compliance:
   - ➪ Security Practices: Implementing security measures within Azure DevOps.

8. Collaboration & Communication:
   - ➪ Communication: Using various communication channels within Azure DevOps.
   - ➪ Collaboration: Facilitating teamwork & knowledge sharing.

9. Advanced Topics:
   - ➪ Containerization (Docker): Understanding & using Docker for containerized applications.
   - ➪ Orchestration (Kubernetes):

Git

- ➪ Getting Started with Git
- ➪ Install the Git Tools
- ➪ Clone an Existing Repository
- ➪ Add Files to a Repository
- ➪ Edit Files in a Git Repository
- ➪ Create and Merge Branches
- ➪ Rewrite History in a Git Repository
- ➪ Resolve Merge Conflicts

RDBMS & SQL Server

- ➪ Introduction to RDBMS
- ➪ Introduction to databases
- ➪ Data Models in Database
- ➪ Properties of RDBMS
- ➪ Normalization
- ➪ CODD's Relational Database Rules
- ➪ Data Integrity
- ➪ T-SQL Language

Working with Data Types, Tables & Data Integrity covering DDL, DML, DCL statements

- Working with Data Types (Only Basics of Data Types)
- Working with Schema
- Working with Tables
- Implementing Data Integrity

Beginning with Transact-SQL

- Transact-SQL
- System Functions
- Advanced T-SQL Queries`
- Advanced T-SQL Statements
- Other T-SQL Statements
- Set Operators
- Transact-SQL
- System Functions
- Advanced T-SQL Queries
- Advanced T-SQL Statements
- Other T-SQL Statements

Working with Joins & Subqueries

- What are Joins?
- Types of joins
- Subqueries

Database Objects: Indexes & Views

- Introduction to Index in SQL Server
- Introduction to Views in SQL Server

Stored Procedures

- Stored Procedure
- Implementing Stored Procedure
- Exception handling using TRY-CATCH

ADO.NET + LINQ + EF Core

- ADO.NET Architecture
- .NET Data Providers
- DB Connectivity Architectures in .NET
- Elements of .NET Data Providers
- Introduction to SQL Server
- Namespaces in ADO.NET
- Using server explorer window
- Connection class
- Command class
- Direct Command execution against database
- Using Parameters in command
- Performing CRUD operations

LINQ

- Language Integrated Query
- Introduction , LINQ Syntax
- Introduction to System.LINQ.Queryable
- Query Operators
- Select, from, Where
- ofType
- OrderBy
- ThenBy
- GroupBy, into
- Select
- SelectMany
- Take, TakeWhile
- First
- FirstOrDefault
- Single
- SingleOrDefault
- Aggregate functions Sum, Min, Max, Average, Count
- Distinct
- Intersect
- Except
- Join
- LINQ projection
- Deferred execution vs immediate execution
- Let keyword
- LINQ to Object
- LINQ to DataTable

Entity Framework Core

- Overview of ORM Products
- Entity Framework introduction
- Using Database first Approach
- Using Code First approach
- Implementing Repository Pattern
- Introduction & Benefits
- Repository Pattern implementation

- Setting up Entities in EFCore
- Using LINQ to Entities to perform CRUD operations
- SQL Query Logging
- Migration & Database Update
- Eager Loading Vs Explicit Loading Vs Lazy Loading
- Raw SQL And Stored Procedures

ASP.NET Core Web API

- ASP .Net Core Fundamentals
- ASP.NET Core - Project.Json
- ASP.NET Core – Configuration
- Middleware Pipeline

Introduction to .Net Core WebAPI

- Introduction to Web Service
- Introduction to REST API
- Introduction to Web API
- Difference between Web Service, WCF Service and Web API
- HTTPS Verbs
- Web API Routing
- Configuring WebApi
- Testing the Web API Project with Postman and Swagger
- Building first ASP.NET Core Web API
- Fluent Validation

Working with Relational Data using Entity Framework Core

- Relationships in EF Core
- HTTP Response Status Codes
- Try-Catch-Finally block
- Throwing custom exceptions
- Global error handling
- Custom global error handling
- DML Manipulation using Repository Pattern

Controller Action Return Types

- Introduction to Controller Action Return Types
- Specific Type
- IActionResult
- ActionResult<Type>
- Custom Return Type

- Web API Versioning
- Web API Logging
- Unit Testing in Web API

Security on Web API

- Configuring Identity services
- Configuring authentication
- Preventing Cross Site Scripting
- Enabling Cross-Origin Requests (CORS)
- JWT Token Authentication

Microservices Fundamentals

- Microservices Fundamentals
- Basic
- ASP.NET Core Microservices
- Advance
- Introduction to Docker
- Choosing Between .NET 6 and .NET Framework for Docker Containers
- Architecting container and microservice-based applications
- Development environment for Docker apps
- Designing and Developing Multi-Container and Microservice-Based .NET Applications
- Implement reads/queries in a CQRS microservice
- Implementing resilient applications in .net
- Implement authentication in .NET microservices and web applications

Web Basics – HTML, CSS, JavaScript, Es6 & TypeScript

JavaScript

- Introduction to JavaScript
- Data Types, Literals, Variables & Constants
- Control Flow, Expression & Operators
- Functions & Variable Scope
- JavaScript Object & Object-Oriented Programming
- Exceptions & Error Handling
- Iterators & Generators

HTML

- ➪ HTML-Introduction
- ➪ HTML-Basic Formatting Tags
- ➪ HTML-Grouping Using Div Span
- ➪ HTML-Lists
- ➪ HTML-Images
- ➪ HTML-Hyperlink
- ➪ HTML-Table
- ➪ HTML-Form
- ➪ HTML-Headers
- ➪ New Form Elements
- ➪ Understand the new HTML form elements such as date, number, range, email, search and datalist
- ➪ Understand audio, video, article tags

CSS 3

- ➪ CSS-Introduction
- ➪ Syntax
- ➪ Selectors
- ➪ Color Background Cursor
- ➪ Text Fonts
- ➪ Box Model
- ➪ Display Positioning
- ➪ CSS Floats
- ➪ CSS Floats

Introducing TypeScript

- ➪ TypeScript Syntax
- ➪ Programming Editors
- ➪ The Type System – Defining Variables
- ➪ The Type System – Defining Arrays
- ➪ Type in Functions
- ➪ Type Inference
- ➪ Defining Classes
- ➪ Class Methods
- ➪ Visibility Control
- ➪ Class Constructors
- ➪ Class Constructors – Alternate Form
- ➪ Interfaces

Working with ES6 Modules

- ➪ var vs let
- ➪ Arrow Functions
- ➪ Arrow Function Compact Syntax
- ➪ Template Strings
- ➪ Generics in Class ,
- ➪ Generics in Function

Angular 18

Introducing Angular

- ➪ What is Angular?
- ➪ Central Features of the Angular Framework
- ➪ Appropriate Use Cases
- ➪ Building Blocks of an Angular Application
- ➪ Basic Architecture of an Angular Application
- ➪ Installing and Using Angular
- ➪ Anatomy of an Angular Application
- ➪ Running the Application
- ➪ Building and Deploying the Application

Components & Templates

- ➪ Creating a Component Using Angular CLI
- ➪ The Component Class
- ➪ The @Component Decorator
- ➪ Registering a Component to Its Module
- ➪ Component Template
- ➪ Using a Component
- ➪ Component Hierarchy
- ➪ Component Lifecycle Hooks
- ➪ Template Location
- ➪ The Mustache {{ }} Syntax
- ➪ Setting DOM Element Properties
- ➪ Setting Element Body Text
- ➪ Event Binding
- ➪ Expression Event Handler
- ➪ Attribute Directives
- ➪ Structural Directives
- ➪ Looping Using ngFor
- ➪ Grouping Elements
- ➪ Template Reference Variable
- ➪ @Output() - Child Component
- ➪ @Output() - Parent Component
- ➪ Full Two Way Binding
- ➪ Setting up Two Way Data Binding in Parent

Template Driven & Reactive Forms

- Template Driven Forms
- Importing Forms Module
- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Reactive Forms Overview
- Import ReactiveFormsModule
- Getting Input Values
- Setting Form Values
- Validation
- Using a Custom Validator
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template

Services & Dependency Injection

- The Service Class
- What is Dependency Injection?
- Injecting a Service Instance
- Injectors
- Dependency Injection in Other Artifacts
- Providing an Alternate Implementation

Pipes & Data Formatting

- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- Using a Pipe with ngFor
- A Filter Pipe

Angular Routing & Angular Modules

- The Router Component
- The Angular Router API
- Creating a Router Enabled Application
- Passing Route Parameters
- Anatomy of a Module Class
- @NgModule Properties
- Using One Module from Another

HTTP Client

- The Angular HTTP Client
- Importing HttpClientModule
- Service Using HttpClient
- Making a GET Request
- Observable Object
- Error Handling & Customizing the Error Object
- Returning an HttpResponse Object
- Creating New Observables
- Observable Operators
- The map and filter Operators

Observables & RxJS Library

- Observables Overview
- Observables in Angular
- Introduction to RxJS library
- Angular Authentication
  with JSON Web Tokens (JWT

Our Students Are Placed In

	tcs TATA CONSULTANCY SERVICES	vsoft ingenuity at work	Cognizant	Microsoft	ENVISION	Sunseaz Technologies Pvt Ltd	wipro	V-SOFT CONSULTING	amazon
PrimeSoft	statista	Practera	PIERSOFT	KELLTON TECH	Capgemini	KENSIUM	SOLIX Empowering Data Management	sumtotal A Skillsoft Company	MIRABEL TECHNOLOGIES
VALUE MOMENTUM	PERVACIO	Apollo Hospitals	LARSEN & TOUBRO	STIC Soft When you need the results	Aadvi Tech Solutions	NOOR	servicenow	ValueLabs	erwin
CyberTech	BLACK KNIGHT	Micron	highradius	HEXAWARE	NextSphere technologies	GLOBAL Management Services	splashBI	archents	techforce.ai
Nendrasys	magnasoft	Innominds Powering the Digital Next	High Noon Consulting	JRD SYSTEMS	Medtronic	MAXO	PTW	Uber	techolution
SOLUGENIX	Collabera Value. Accelerated.	iPrism Technologies	meslova	{actlogica} Actuarial | Finance | Technology	S&P Global	Fission Technosolutions	SOLIGENIX	TechnoFacts	SAHASYA GLOBAL SOLUTIONS

QualityThought®
Transforming Dreams! Redefining Future!

| 97034 61963

Quality Thought Infosystems India (P) Ltd.

#302, Nilgiri Block, Ameerpet, Hyderabad-500016 | www.qualitythought.in | info@qualitythought.in