
CELEBRATING
14 YEARS
QualityThought®
Transforming Dreams! Redefining Future!
JOB ORIENTED
JOIP
INTENSIVE PROGRAM
.NET Core
Full Stack
Developer
C#12 with .NET 8
Asp.Net MVC Core Developer
Asp.Net Web API Core Developer
ReactJS UI Developer
Database logic using EF.Core
Unit Testing by NUnit and Others
SQLServer, My SQL, Mongo DB
DevOps Tools, Agile, JIRA
Micro Services
Deployment on Azure Services
Design Patterns
qttworld
www.qualitythought.in

CELEBRATING 14 YEARS
QualityThought®

Microsoft .NET Fraework

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

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

Clinet 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

QualityThought

Microsoft .NET Fraework

.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
- PartialViews
- Logging
- Filters
- Database programming with EF.Core
- CRUP Operations
- Working and Customizing Scafolding Views

Asp.Net Core Web Api - REST

- Introduction
- REST web service, Web API & Microservice
- Inside HttpRequest HttpResponse
- 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 reuest 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

QualityThought

Microsoft .NET Fraework

RDBMS

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

SQL Server

- Management Studio
- Tables - Colums - 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
- CURD 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
- Regualar 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

Oops 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
- thow
- 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 , Interupt, 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

Web Development Fundamentals

- What is Web?
- Web Server, Web Client
- Website, Webpage
- Request and Response
- Http/Https
- Languages and Tools of Web Development
- How the Web Works
- Inspecting HTTP Requests and Responses
- HTML Basics
- CSS Basics
- Formatting Code
- Inspecting Pages Using DevTools
- Validating Web Pages

HTML & HTML 5

- Introduction
- The Head Section
- Text
- Entities
- Hyperlinks
- Images
- Video and Audio
- Lists
- Tables
- Containers
- Semantic Elements
- Structuring a Web Page

Typography

- Introduction
- Styling Fonts
- Embedding Web Fonts
- Flash of Un-styled Text
- Font Services
- System Font Stack
- Sizing Fonts
- Vertical Spacing
- Horizontal Spacing
- Formatting Text
- Practices

CSS Layout

- Introduction
- The Box Model
- Sizing Elements
- Overflowing
- Measurement Units
- Positioning
- Floating Elements
- Flexbox
- Grid
- Hiding Elements
- Media Queries
- Summary
- Practices

Bootstrap

- Introduction
- Bootstrap classes
- Responsive grid system
- Columns and rows
- Components
- Responsive navbars
- Margins
- Modals, dialogs
- Buttons
- Forms
- List groups
- Badges, pills
- Cards
- Tables
- Alerts
- Navigation options
- Links

JS
JavaScript

14 YEARS
QualityThought

JavaScript Basics

- Introduction to JavaScript
- How to Write JavaScript Program
- How to debug JavaScript code
- Where to place JavaScript code in the HTML file
- Comments and Statements
- Data Types
- Variables
- Hosting
- Operators
- Conditional Statements
- Loop
- Functions
- Function Expressions in JavaScript
- Recursive Function
- Popup Boxes
- Events
- Examples of JavaScript Events
- Exception Handling in JavaScript
- this keyword
- Call, Apply, bind
- Output
- API call using HTTP, FETCH
- HTTP Methods (GET, POST, PUT, DELETE)
- web storage
- DOM
- Add execution context and call stack
- In Javascript basics

JavaScript Promise

- Promise
- Promise Chaining in JavaScript
- Promise.race() vs. Promise.all()
- Promise Error Handling
- Async Await

OOPs

- How to Create Objects in JavaScript
- Prototypical Inheritance in JavaScript
- Classes
- Inheritance
- Abstraction
- Polymorphism
- Encapsulation
- Constructor Function
- Object Factories
- Instance Of
- Call by Value & call by reference
- Object Creation Using New Object Method
- Javascript Objects Are Mutable

JavaScript Advance Es6

- let Keyword
- const Keyword
- Iterators and Iterables
- function are First-class citizen
- Callback functions
  Synchronous and Asynchronous
- Anonymous Function
- Immediately Invoked
  Function Expressions (IIFE)
- Arrow Function
- Console Object
- Shadowing

JavaScript – New Features

- Template Literals
- Object Literals
- Default Parameters

React JS

14 YEARS
QualityThought

ReactJS18.2.0 Training Overview

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.
It is used for creating dynamic and interactive user interfaces for mobile and web applications. It is highly flexible, declarative and efficient for developing scalable, simple, and fast front-end for web & mobile applications. In simple terms, React JS effectively handles the view layer of mobile and web application.

React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, ReactJS is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality

React is flexible in such a way that, in any application, we can use as little or as much React as you need. For example, react can be used in any existing web application to develop a new feature or even the application's entire UI.

ReactJS has become highly popular across the globe because of its extra simplicity and flexibility. Many people are even referring to ReactJS as the future of web development.

Part of this huge popularity comes from the fact that top corporations such as Facebook, PayPal, Uber, Instagram, Airbnb etc use it to develop the user interfaces.

Objectives of the Course

This is to provide in-depth insights about ReactJS and keep updated yourself with latest ReactJS concepts and it's integrated other technologies according to updated industry standards.

- The main objective of React Training is to make you as a top-class React.js Developer able to design and build modern interactive user interface components to enhance application performance.

Who should do the Course

- Anyone who is trying to learn fastest growing UI framework and one of the top 3 frameworks. It is growing fast in terms of opportunities too.

Prerequisites

- HTML
- JavaScript
- CSS

ES6: JavaScript concepts related to ReacJS

- History of JavaScript
- Features
- let & const with example
- Arrow Functions
- Alternative

React JS

QualityThought
14 YEARS

ReactJS: Introduction

- What is ReactJS?
- Version history.
- Why ReactJS?
- Advantages of React JS
- Internals of ReactJS.
- Work flow of React JS
- Scope of React JS
- Software's required.
- Tips and Arrow Functions
- Exports and Imports
- Tips for exports and imports
- Classes with example
- Inheritance
- Spread and rest Parameter
- Destructing

ReactJS: Setup – First Application

- About NodeJS
- Installation
- Npm – managing the packages
- Create First ReactJS Application
- Build and Execute
- About Visual Studio Code
- Installation of Visual Studio Code
- Managing the ReactJS application using Visual Studio Code

ReactJS: JSX

- Overview of JSX
- Jsx syntax
- Difference between JS and JSX.
- Jsx Parser
- Comments in jsx
- Introduction of Virtual DOM.
- Rendering an Element into the DOM
- Naming Conventions
- JSX Transpilation to React Code Example
- Running the Transpiled Code
- What are the JavaScript expressions available in JSX?

ReactJS: All about Components

- React Components
- Creating a Functional Component Example
- Components vs Elements
- Properties
- Spread Attributes (an ES6 Feature)
- Expressions
- Fragments
- Functional Components
- Example of JSX Nesting
- Setting CSS Styles Using Classes
- Setting CSS Styles Directly
- JSX Escapes Values
- Working with Lists of Items - Iterating through Lists
- Keys in Lists
- Example List with Key
- State in React
- Setting state
- Types of State Data
- State Hierarchy
- Lifting State Up
- Props vs. State
- Immutability
- Immutability – Why?
- Virtual DOM and State
- Updating Input fields
- Passing Props to Components
- Passing Functions to Components
- Event Handling
- Event Handler Example
- Event Binding - Dos
- Event Binding – Don'ts
- Passing Parameters to Event Handlers
- Component Life-cycle
- Initial Render
- Props Change
- Stage Change
- Component willMount
- Component didMount
- Component Unmount
- Life-cycle in Functional Components
- Applying Different Lifecycles in the Application.
- When to choose Appropriate lifecycles.
- Interaction in between components
- Component Communication using Context

React JS

QualityThought — 14 YEARS

Forms and UI

- Lists of Form components.
- Setup Controlled and Uncontrolled form components.
- React JS Form validations.
- Create a React Form.
- Client-side form validation.
- Using Formik Library with React
- Using YUP Library with Formik in React

Working with Rest API

- Calling REST API
- Sending POST request
- CURD Operations example

React Router

- Routing and Navigation
- react-routet
- Creating a react-router based project
- A Basic Routed Component
- Router vs. BrowserRouter
- The Route component
- Redirect Route
- Navigating with
- Route Parameters
- Retrieving Route Parameters
- QueryString Parameters
- Create a Single Page Application.
- Applying Routing.
- Dynamically render the components based on the url

State Management for React

- React State Basics – Props and State
- Props
- State in Class Based Components
- Managing State with Hooks in Functional Components
- The Problem with Props and State
- Redux State Library
- Redux Advantages
- Redux Disadvantages
- Basic Rules for State Management
- Types of State
- Data State
- Communication State
- Control State
- Session State
- Location State
- Location State Side Effects

Event handling in JSX

- on Blur, onKeyUp, onChange and other useful primary events in React JS.
- How to Sharing events between the components?
- Communicate Data between components.
- Applying all lists of events

Styles in ReactJS

- CSS and inline styles in React JS overview.
- Introduction to styled components
- Styling the application using styled component
- How to use Animations in the Application

Building React Apps with Redux

- Redux
- Redux Terminology
- Redux Principles
- Redux: Actions
- Redux Action Types
- Action Creators
- Dispatching Actions
- Data Flow Basics
- Redux Reducers
- Pure Functions
- Reducer Example
- Returning Default State
- Creating a Development Environment with create-react-app
- Using Redux with React
- Initializing the Store
- Immutability
- Benefits of Immutable State
- Mutability of Standard types
- Copying Objects in JavaScript
- Copying Arrays in JavaScript
- One Store - Multiple Reducers
- Combining Reducers
- Components and Redux
- The React-Redux Package
- Wrapping App with Provider
- mapStateToProps
- mapDispatchToProps
- Using Mapped Properties & Methods
- Wrapping Components with Connect
- Configure Store
- Programming Advice - MultiTab Console

React JS

QualityThought 14 YEARS

Flux

- What is Flux Architecture?
- What are the Flux Components available?
- How Flux works?
- Flux and React works together.

Using React Hooks

- Functional Component Shortcomings
- Hooks Overview
- Hook Rules
- Functional Component Props
- The useState Hook
- The useEffect Hook
- The useContext Hook
- Additional Hooks
- The useReducer Hook
- The useMemo Hook
- The useRef Hook
- Creating Custom React Hooks

Unit Testing React with React Testing Library

- What are the necessary Tools required for Unit Testing?
- React Unit Testing overview
- Introduction to JEST.

Server Integration & Deployment

- https
- httpster
- npm

Code Splitting

- Code splitting & Suspense
- Route Based Code Splitting
- Lazy Loading

Isomorphic React

- Server-Side Rendering
- SSR with React - Setup & Server
- SSR with React - The Toolchain

New Features of React 16,17 & 18

Webpack Primer & Isomorphic React

Isomorphic React

- Webpack and its use
- Setting up and installing Webpack
- Working with the configuration file of Webpack
- Working with loaders
- Quick word on lazy loading, code splitting, and tree shaking
- Setting up a hot module replacement
- Server-side rendering (SSR)
- Working with renderToStatic Markup and renderToString methods

Cloud -AWS

- Types of Applications
- PC vs Server
- Need of Cloud Infrastructure
- What is Cloud Computing
- What is AWS? And Why to choose AWS
- How to create an account in AWS
- What an Operating System and its types
- Importance of Linux in Cloud Environment
- How to create a Linux server in AWS
- Putty vs Xshell
- AWS EC2 Instance
- Types of instances in AWS
- AWS Tenancy
- AWS Launch Template
- AWS EBS volumes
- Need of Elastic IP Address
- AWS RDS service for JDBC Application?
- How to deploy web application in AWS Elastic Bean Stalk
- AWS S3 service
- How to deploy spring boot application in AWS Cloud

Communication Skills

Roots of Communication	LSRW	
	7 Cs of Communication	
Roots of Grammar	Parts of Speech	Mastering Helping Verb And Main Verb
	Sentence Structure Development	
	Tense Logic	
	Worksheet Sessions	
Speech Intelligence	Vocabulary Development	
	Usage of words	
	Group Discussions	
	JAMS	
	Debates	
	Public Speaking	
Personality Development	Imagination and Innovation Training	
	Centralized Brain Storming	
	Problem Solving Skill s	
	Desiion Making	
Mangement Skills	Time Management	
	Team Building	
	Task Management	
	Leadership Skills	
Interview Skills	Employbility	
	Think like a Professional	
	Clearing HR Rounds	
	Salary Negotiation	
	Bond Negotiation	
Presentation Skills	Research Skills	Dream Company Reading Skillls Comprehension Skills
	Public Speaking	
	Visualization	
	White Board Presentation	
	Mastering Powerpoint	
	Content Creation	
Personality Development	Mind Mapping	
	Role plays	
	Mock Interview on the Hot Seat	
	Listening Skills	
	Critical Thinking	
	Thought Analysis	
	SWOT Analysis	

QualityThought

Aptitude & Reasoning

Quantitative

- Basic Maths
- Algebra
- Percentages
- Profit And Loss
- Discounts
- Averages
- Time and Work
- Chain Rule
- Pipes and Cisterns
- Ratios
- Proportions
- Partnerships
- Time and Distance
- Trains
- Boats and Streams
- Simple Interest
- Compound Interest

Data Interpretation

- Bar Charts
- Line Charts
- Pie Charts
- Table Charts

Reasoning

- Directions
- Letter Series
- Number Series
- Coding - Decoding
- Blood Relations
- Statement and Assumption
- Analogy
- Odd Man Out Series
- Venn Diagrams
- Mirror Images
- Water Images
- Arranging in Order
- Paper Folding / Cutting
- Grouping
- Counting the figures
- Clocks
- Calenders
- Seating Arrangements
- Syllogism
- Puzzles

OUR STUDENTS ARE PLACED IN

QualityThought

74162 36806

Quality Thought Infosystems India (P) Ltd.

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