

GAMAKA

Artificial Intelligence Solutions

Pune – Kothrud & Aundh, Mumbai

Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293
www.gamakaai.com

Angular 9, TypeScript

Duration: 60 hours with hands on tutorials

Live Projects & 60 assignments.

Course Description:

Our Angular training course gives web developers a focused hands-on experience in which they can explore and build single page HTML5 apps using Google's exciting clientside MVC framework Angular. This is a practical course aimed at developing real world skills.

Google's Angular allows you to build applications and reuse your code, as well as deploy for any platform; web, mobile web, native mobile and desktop.

Angular promises high speed and control over scalability by building data models on RxJS and Immutable.js

Angular's new tooling allows developers to build features quickly, with simple, declarative templates.

Ivy full version is due to revolutionise Angular, by making development smaller, faster and simpler - IVY is an initiative to build a next-generation rendering pipeline for Angular, and, for this, the Angular team re-wrote the code that translates the Angular template to whatever we rendered in the browser. It uses the incremental DOM

GAMAKA

Artificial Intelligence Solutions

Pune – Kothrud & Aundh, Mumbai

Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293

www.gamakaai.com

Typescript

- Typescript
- What is Typescript
- First Typescript Example
- Basic Data Types & Variables
- Using Types
- Variables in Typescript
- Type Assertion
- Operators & their types
- Decision making constructs
- Loops
- Using Arrays
- Functions
- Writing & Using Classes
- Constructor Method
- Inheritance of Classes
- Type Assertion
- Abstract Class
- Working With Interfaces
- All About Generics

Introduction to Angular

- What is Angular?
- Angular 7v/6v/5 v/s 4 v/s 2 v/s AngularJS
- Angular CLI
- NodeJS Introduction(NPM)
- Setup of NodeJs and Angular
- What is Typescript?
- How does Angular get started?
First Angular App Nested Bean
Factories

Components Overview

- Introduction to Components
- Creating components
- Role of AppModule & Component Declaration
- Registering Components
- Using Registered Components
- Creating Components with CLI
- Multiple components & passing data
- Nesting Components
- Working with Component templates
- Working with Component Styles
- Understanding Component Selector

Components & Databinding

- Introduction to Modules & Databinding
- Splitting Apps into Components
- Property & Event binding overview
- Binding to Custom Properties
- Assigning an Alias to Custom Properties
- Binding to Custom Events
- Assigning an Alias to Custom events
- Custom Property and Event Binding Summary
- Understanding View Encapsulation
- Using Local References in Templates
- Getting Access to the Template & DOM with @ViewChild
- Projecting Content into Components with ng-content
- Understanding the Component Lifecycle
- Lifecycle Hooks in Action
- Lifecycle Hooks & Template Access
- Accessing ng-content with @ContentChild

Directives

- Understanding Directives
- Using ngIf to Output Data Conditionally
- Enhancing ngIf with an Else Condition
- Using the Renderer to build Better Attribute Directive
- More about Renderer

GAMAKA

Artificial Intelligence Solutions

Pune – Kothrud & Aundh, Mumbai

Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293
www.gamakaai.com

- Output Lists with ngFor
- Styling Elements Dynamically with ngStyle
- Applying CSS Classes Dynamically with ngClass
- Creating Basic Attribute Directive
- Listen to Host Events using HostListener
- Bind to Host Properties using HostBinding
- Binding to Directive Properties
- Behind the scenes of Structural Directives
- What is ngSwitch

Services & Dependency

- Introduction to Dependency Injection
- Why do we need Services ?
- Creating a Logging Service
- Injecting the Logging Service into Components
- Creating a Data Service
- Understanding Hierarchical Injector
- How many Instances of Service?
- Injecting Services into Services
- Using Service for Cross-Component Communication

Changing Pages with Routing

- What is Routing?
- Why do we need a Router?
- Setting up and Loading Routes
- Navigating with Router Links
- Understanding Navigation Paths
- Styling Active Router Links
- Navigating Programmatically
- Using Relative Paths in Programmatic Navigation
- Passing Parameters to Routes
- Fetching Route Parameters
- Fetching Route Parameters Reactively
- Resolving Dynamic Data with the resolve Guard
- Understanding Location strategies
- Understanding Observables
- Route Observables
- Passing Query Parameters and Fragments
- Retrieving Query Parameters & Fragments
- Setting up Child(Nested) Routes
- Configuring the Handling of Query Parameters
- Redirection & Wildcard Routes
- Outsourcing the Route Configuration
- Introduction to Route Guards
- Protecting Routes with canActivate
- Controlling Navigation with canDeactivate
- Passing static data to a Route

Template Driven Forms

- Introduction to handling forms
- Why do we need Angular's help?
- Template Driven(TD) v/s Reactive Approach
- Creating Template driven Forms & Registering Controls
- Submitting & Using the Form
- Understanding Form State
- Accessing the Form with @ViewChild
- Adding Validation to check User Input
- Built-in Validators & Using HTML5 Validation
- Using the Form State
- Outputting Validation Error Messages
- Set Default Values with ngModel Property Binding
- Using ngModel with Two-Way-Binding
- Grouping Form Controls
- Handling Radio Buttons
- Setting & Patching form values
- Using Template Driven Form Data
- Resetting Template Driven Forms
- Controllers

GAMAKA

Artificial Intelligence Solutions

Pune – Kothrud & Aundh, Mumbai

Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293
www.gamakaai.com

Reactive Forms

- Introduction to Reactive Approach
- Creating a Reactive Form in Code
- Syncing HTML and Form
- Submitting Reactive Forms
- Adding Validation to Reactive Forms
- Getting Access to Controls
- Grouping Controls
- Arrays of Form Controls
- Creating Custom Validators
- Using Error Codes with Reactive Forms
- Creating Custom Async Validator
- Reacting to Status or Value Changes
- Pure & Impure Pipes
- Understanding 'async' Pipes

Making HTTP Requests

- Introduction to Http Requests
- How HttpRequests Work in SPAs
- Sending Requests
- Adjusting Request Headers
- Sending GET Requests
- Sending a PUT Request
- Observable Operators
- Using Returned Data
- Catching Http Errors
- Using 'async' Pipe with Http Requests

Authentication & Route Protection

- Introduction to Authentication
- How Authentication works in SPAs
- About JSON Web Tokens
- Creating Signup Page & Route
- Setting up Firebase SDK
- Signing Users Up
- Signing Users In
- Requiring & Sending Token
- checking & Using Authentication Status
- Adding Logout Button
- Route Protection & Redirection Example

Angular Modules & Optimizing Apps

- The idea behind Modules
- Understanding App Module
- Understanding Feature Modules
- Creating a Feature Module
- Registering Routes in Feature Modules
- Understanding Shared Modules
- Creating a Shared Module
- Creating the Auth Feature module
- Understanding Lazy Loading
- How Modules & Services Work Together
- Understanding the Core Module
- Creating a Basic Core Module
- Using Ahead-of-time compilation
- Using AoT Compilation with CLI

HttpClient

- Introduction to HttpClient
- Unlocking the HttpClient
- Request Configuration & Response
- Requesting Events
- Setting Header
- Http Parameters
- Progress
- Interceptors
- Modifying Requests in Interceptors
- Multiple interceptors

Angular Animations

- Introduction to Angular Animations
- Animation Triggers & State
- Transition Phases
- The "void" State

GAMAKA

Artificial Intelligence Solutions

Pune – Kothrud & Aundh, Mumbai

Email: enquiry@gamakaai.com Cell: 91-7378483656. WhatsApp: 91-7378493293

www.gamakaai.com

- Switching between States
- Transitions
- Advanced Transitions
- Using Keyframes for Animations
- Grouping Transitions
- Using Animation Callbacks

Unit Testing in Angular Apps

- Introduction
- Why Unit Tests?
- Analyzing the CLI Testing Setup
- Running Tests with CLI
- Adding Components and Some Tests
- Testing Dependencies: Components & Services
- Simulating Async Tasks
- Isolated v/s Non-isolated Tests

Advanced Topics – RxJS & Dynamic Form

- Promises and Observables
- What are Promises and Observables in Angular
- Comparing the two
- Handling HTTP requests with Promise
- Resolving multiple values with Observables
- RxJS library for reactive programming with Observables
- Interacting with http get() in Angular and http Post
- Dynamic Forms