# Technovert - A Microsoft Development Partner Angular 7 Syllabus

## Introduction to Angular

* What is Angular?
* Angular Versions: AngularJS **(vs)** Angular
* Setup for local development environment
  + Installing NodeJS, NPM
  + Installing Visual Studio Code
  + Angular CLI
* Develop First Angular program using Angular CLI and Visual Studio Code.

## Angular Architecture

* Introduction
* Basic Building Blocks of Angular Applications
* Angular Modules and @NgModule decorator
* Angular Libraries
* Component, Templates and Metadata
* Data Binding
* Directives
* Services and Dependency Injection

## Displaying Dynamic Data

* Types of Directives
* Template Expressions
* String Interpolation
* Built-In Directives
  + ngIf
  + ngSwitch
  + ngFor
* \* vs <template>
* When to use <ng-container>

## Angular Components Deep Dive

* What are Components?
* Components Life Cycle Hooks.
* Dynamic Components.
* Working with Model Class.
* Nested Components.
* Working with Arrays / Collections.

# Technovert - A Microsoft Development Partner Angular 7 Syllabus

## Data Binding

* Binding properties and Interpolation
* One-way Binding / Property Binding
* Event Binding
* Two-way Binding
* Two-way binding with NgModel
* Attribute Binding
* Style and Class Binding

## Styles Binding In Components

* Style and Class Bindings
* Built-In Directives - NgStyle & NgClass
* [Using Component Styles](https://angular.io/docs/ts/latest/guide/component-styles.html#using-component-styles)
* [Special selectors](https://angular.io/docs/ts/latest/guide/component-styles.html#special-selectors)
* [Loading Styles into Components](https://angular.io/docs/ts/latest/guide/component-styles.html#loading-styles)
* View Encapsulation
* ViewChild & ViewChildren
* ContentChild & ContentChildren

## Template Driven Forms

* Introduction
* Create the component that controls the form
* Create a template with the initial form layout
* Bind data properties to each form input control with the ngModel two-way data binding syntax
* Add the name attribute to each form input control
* Add custom CSS to provide visual feedback
* Show and hide validation error messages
* Handle form submission with **ngSubmit**
* Disable the form’s submit button until the form is valid
* Resetting the form.

## Reactive Forms

* Reactive Forms Introduction
* More Form Controls
* Form Control Properties
* setValue and patchValue
* Validating Form Elements
* Submitting and Resetting forms
* Observing and Reacting to Form Changes

# Technovert - A Microsoft Development Partner Angular 7 Syllabus

* Using FormBuilder

## Working with Pipes

* Built-in Pipes
* Using parameters and chaining Pipes
* Custom Pipes
* Parameterized Custom Pipe
* Pipes and Change Detection
* Pure and Impure pipes
* Changes in Pipes Syntax from 4 to 5

## Custom Directives

* Custom Attribute Directive
* Using HostListener
* Using HostBinding
* Custom Validator Directive

## Dependency Injection

* Understanding Dependency Injection
* Understanding DI in Angular Framework
* ReflectiveInjector
* Exploring Provider
* Types of Tokens
* Types of Dependencies
* Configuring DI using Providers
* Implementing DI in Angular
* Optional Dependencies

## Services in Angular

* Building and Injecting Custom Services
* Service using another Service

## Reactive Extension for JavaScript

* Introduction
* Observable and Observer
* Reactive Operators

## Http Client Service

* HttpClientModule and HttpClient Classes

# Technovert - A Microsoft Development Partner Angular 7 Syllabus

* Writing Service with Get / Add / Edit / Delete
* Using Service in Component

## Angular Routing

* Introduction
* Configuring and Navigating
* Parameterized routes
* Nested (or) Child Routes
* Router Guards & Routing Strategies

## Angular Modules

* AppModule as Root Module
* Feature modules
* Lazy Loading a Module
* Shared Module

## Performing CRUD Operations as in Realtime Angular Application.

* Create and ASP.NET MVC Web API Application
* Add Angular Seed, download files and Configure Project
* Setup Project for Entity Framework Code First Model
* Add Web API Controller to perform CRUD Operations using EF.
* Create Angular Feature and Routing Modules
* Configure Angular to use HTTP and Json Services.
* Program Angular Modules to performing CRUD Operations.

## Angular CLI and MVC Application

* Create an Angular CLI Project
* Build CLI Project
* Integrate build output with MVC Application