Angular 12

- **Writing Your First Angular Web Application**
    - Introduction to Single Page Application
    - What do we need Angular for?
    - Setting up the Development Environment
    - Angular Application Architecture
    - First Angular App
    - Bootstrapping crash course

- **TypeScript**
    - Angular is built in TypeScript
    - What do we get with TypeScript?
    - Built-in types
    - Classes
        - Properties
        - Methods
        - Constructors
        - Inheritance
    - Utilities
        - Fat Arrow Functions
        - Template String

- **Angular Components**

Angular Components and the @Component Decorator
Creating an Angular Component using the Angular CLI
Component Decorator Metadata
Data Binding - String Interpolation - {{ }}
Property Binding - []
Event Binding - ()
@Input and @Output Property Decorators
Two-way Data Binding - [()]
Attribute Binding - [attr.]
Class Binding - [class.]
Style Bindings - [style.]

- **Angular Directive and Pipes**
  - Introduction to Directives
  - Built in Structural Directives
  - Built in Attribute Directives
  - Building custom Attribute Directives
  - Building Custom Structural Directives
  - Introduction to Pipes in Angular
  - Using Built In Pipes
  - Creating Custom Pipes in Angular
  - Pure and Impure Pipes in Angular

- **Angular Forms**
  - Types of Form in Angular
  - Reactive form in Angular
  - Dynamically Adding or Removing Form Control(s) or Form Group(s) using Form Array(s)
  - Validations to your Reactive Form
  - Validations to your Reactive Form
  - Resetting the value of a form

- **Angular Routing with Guards**
  - Why Do We Need Routing?
  - How client-side routing works
  - Components of Angular routing
    - Imports
    - Routes
    - Installing our Routes
    - RouterOutlet using <router-outlet>
    - RouterLink using [routerLink]
  - Putting all together
  - Implementing Child Routes
  - Introduction to Route Guards in Angular
  - CanActivate and CanActivateChild Guards in Angular
  - CanDeactivate Guard in Angular
  - Prefetching Data for a Component using Resolve

- **HTTP, Observables**
  - Dependency Injection
  - Hierarchical Dependency Injection
  - Introduction to Services in Angular
  - GET data from a Rest API using HttpClient
  - CRUD Operations using HttpClient, HttpParams and HttpHeaders
  - Observables and Operators in Observables - Map, Retry, Catch
  - Custom HTTP Headers

- **Data Architecture in Angular**

- **Introduction to Redux with TypeScript**
  - Redux
  - Storing our state
  - Setting up Redux

- **Intermediate Redux in Angular**
- **Advance Component**
  - Styling
  - Lifecycle Hooks
  - Advance Templates
  - Change Detection
- **Testing**
  - End to end testing
  - Unit testing
  - Testing Services and component