Angular Training

Duration: 5 Days (40Hrs)

Prerequisite:

- **HTML:** Understanding of HTML structure and elements.
- **CSS:** Knowledge of CSS for styling web pages.
- **JavaScript:** Proficiency in JavaScript syntax, object-oriented programming concepts, and DOM manipulation.
- **TypeScript:** Basic understanding of TypeScript syntax, types, interfaces, and classes.
- **Node.js and npm:** Familiarity with Node.js and npm for package management.

System Requirements

- **Operating System:** Windows 7 or later, macOS, or Linux (Ubuntu 14 or later)
- **Processor:** Intel Core i3 CPU @ 2.00 GHz or higher
- **Memory:** Minimum 4 GB RAM
- **Storage:** 20 GB HDD/SSD or higher
- **Browser:** Latest version of Chrome, Firefox, or Edge

Essential Software

- **Node.js and npm (or yarn):** For package management and running Angular development environment.
- **Text Editor or IDE:** Visual Studio Code, Sublime Text, WebStorm, or any preferred code editor.
- **Angular CLI:** For creating, developing, and serving Angular applications.
- **Git:** For version control (optional but highly recommended).
- **Browser:** Chrome, Firefox, or Edge for development and testing.

Day 1: Introduction to Angular and Core Concepts

- **Angular Overview:**
  - What is Angular?
  - Key features and benefits
  - Comparison with other frameworks
  - Angular ecosystem (CLI, Material, RxJS)
- **Setting Up the Environment:**
  - Node.js and npm installation
  - Angular CLI installation
  - Creating a new Angular project
- **TypeScript Fundamentals:**
  - Introduction to TypeScript
  - Types, interfaces, and classes
  - Modules and namespaces
  - Arrow functions and decorators
- **Angular Architecture:**
  - Components, modules, services, and directives
  - Dependency injection
  - Data binding (property, event, two-way)
  - Templates and components
- **Building Your First Angular App:**
  - Creating components and modules
  - Using data binding
  - Handling user events
  - Routing basics

Day 2: Advanced Components, Forms, and HTTP

- **Deep Dive into Components:**
  - Component lifecycle hooks
  - Component interaction (parent-child, sibling)
  - Content projection
  - ViewChild and ViewChildren
- **Angular Forms:**
  - Template-driven forms
  - Reactive forms
  - Form validation
  - Custom validators
- **HTTP in Angular:**
  - Making HTTP requests (GET, POST, PUT, DELETE)
  - Observables and RxJS
  - Error handling
  - Interceptors
- **Angular Material:**
  - Introduction to Angular Material
  - Creating a Material-based theme
  - Using Material components (cards, buttons, input, etc.)
  - Layouts and grids
- **Dependency Injection:**
  - Providers and injectors
  - Dependency injection in components, services, and modules
  - Tree-shakable providers

Day 3: Routing, State Management with NgRx, and RxJS

- **Angular Router:**
  - Routing fundamentals
  - Route parameters and query parameters
  - Lazy loading modules
  - Guards and resolvers
  - Navigation extras
- **State Management with NgRx:**
  - Introduction to NgRx
  - Store, actions, reducers, and effects
  - Selectors
  - Building a simple NgRx application
- **RxJS Fundamentals:**
  - Introduction to Observables
  - Creating Observables
  - Subscribing to Observables
  - Operators (map, filter, reduce, etc.)
  - Subjects

Day 4: Advanced Topics, Testing, and Real-world Application

- **Advanced Components:**
  - Dynamic components
  - Embedded views
- **Performance Optimization:**
  - Change detection strategies (OnPush)
  - Lazy loading modules
  - Code splitting
  - AOT compilation
- **Angular Universal:**
  - Introduction to server-side rendering
  - Creating a universal Angular application
  - Benefits of server-side rendering
- **Testing in Angular:**
  - Unit testing with Jasmine and Karma
  - Component testing
  - Service testing
  - End-to-end testing with Protractor (or Cypress)
- **Building a Real-world Application:**
  - Project planning and architecture
  - Code structure and organization
  - Best practices for Angular development
  - Integrating third-party libraries

Day 5: Debugging, Troubleshooting, and Advanced Topics

- **Debugging and Troubleshooting:**
  - Angular DevTools
  - Common issues and solutions
  - Debugging techniques
- **Advanced Topics (Optional):**
  - Angular CDK
  - Web Workers
  - Custom directives and pipes
  - Internationalization
- **Hands-on Project:**
  - Building a small Angular application
  - Applying learned concepts
  - Code review and feedback
- **Q&A and Wrap-up:**
  - Addressing participant questions
  - Course summary
  - Next steps and resources