

# **Angular App development with Module Federation**

**Duration: 5 days**

## **Prerequisites Topics (Self-study / Pre-training Session)**

- 1. HTML & CSS (2–3 Hours)**
    - Semantic HTML
    - Forms and Input elements
    - CSS basics (Selectors, Flexbox, Grid)
    - Responsive design basics
  - 2. JavaScript Essentials (4–5 Hours)**
    - Variables (let, const, var)
    - Functions and Arrow Functions
    - ES6+ features (Destructuring, Spread, Rest)
    - Promises and Async/Await
    - DOM Manipulation and Event Handling
  - 3. TypeScript Basics (4–5 Hours)**
    - Installing and setting up TypeScript
    - Variables and Data Types
    - Functions, Interfaces, Classes, Inheritance
    - Modules, Imports, Generics, Decorators
- 

## **Day 1 – Introduction to Angular & Environment Setup**

### **Morning Session**

- What is Angular and its evolution (AngularJS → Angular 19)
- Advantages and real-world applications
- Installing Node.js & Angular CLI
- Creating and running the first Angular app
- Project structure overview
- Core concepts:
  - Modules (NgModule)
  - Components (@Component)
  - Templates and Views

- Directives (\*ngIf, \*ngFor, [ngClass], [ngStyle])

### Afternoon Labs

- **Lab 1:** Set up Angular project & explore structure
  - **Lab 2:** Create components & use directives (\*ngIf, \*ngFor, ngClass, ngStyle)
- 

## Day 2 – Components, Data Binding & Services

### Morning Session

- Component Communication (@Input, @Output, Parent-Child)
- Data Binding: Interpolation, One-way & Two-way (ngModel), Property and Event Binding
- Angular Services & Dependency Injection (@Injectable)
- HttpClientModule for API calls

### Afternoon Labs

- **Lab 3:** Parent-Child Component Communication
  - **Lab 4:** Creating Services & making API calls
- 

## Day 3 – Routing, Forms & State Management

### Morning Session

- Routing & Navigation with RouterModule
- Lazy Loading and Route Guards
- Handling Route Parameters
- Forms in Angular: Template-Driven vs Reactive (Validators, FormGroup, FormControl)

### Afternoon Labs

- **Lab 5:** Implement Routing and Route Guards
  - **Lab 6:** Build Reactive Forms with Validation
- 

## Day 4 – Advanced Topics (including Module Federation)

### Morning Session

- **State Management with NgRx (Redux Pattern)**
  - Introduction to NgRx Store
  - Actions, Reducers, Selectors, Effects
- **Micro-Frontend Architecture and Module Federation (New)**

- Understanding Micro-Frontends and why they matter
- Overview of Webpack 5 Module Federation concept
- Setting up Module Federation in Angular 19 project
- Exposing and consuming modules between multiple Angular apps
- Sharing dependencies and managing version conflicts
- Dynamic Remote Loading and runtime configuration
- Communication between remote modules and host application
- Use cases and best practices for enterprise applications

#### Afternoon Labs

- **Lab 7: State Management with NgRx**
    - Set up NgRx store and manage state with actions and reducers
  - **Lab 8: Implementing Module Federation in Angular (New)**
    - Create a host and remote Angular application
    - Expose components from remote and consume them in host
    - Demonstrate dynamic loading of modules at runtime
    - Verify independent deployment of each micro-frontend
- 

### Day 5 – Testing, Optimization & Deployment

#### Morning Session

- Unit Testing with Jasmine & Karma (components, services, directives)
- Performance Optimization (Lazy Loading, Change Detection, Production Best Practices)
- **Deployment of Angular Apps**
  - Building & optimizing production version
  - Hosting on Firebase, Netlify, or Vercel
- **Deploying Micro-Frontend Applications with Module Federation (New)**
  - Build and deploy independent Angular apps on Firebase/Netlify
  - Environment-specific remote configurations

#### Afternoon Labs

- **Lab 9: Write Unit Tests for Components and Services**
- **Lab 10: Deploy Angular App with Module Federation Support**