**Comprehensive Angular Roadmap Checklist and 15-Day Schedule**

**15-Day Schedule**

**Day 1: Introduction and Setup**

* What is Angular and why use it?
* Angular vs. other frameworks
* Installing Node.js and Angular CLI
* Creating your first Angular application
* Project structure overview

**Day 2: TypeScript Basics**

* TypeScript syntax: Variables, Data Types, and Interfaces
* Functions and Classes
* Modules and Decorators
* Arrow Functions and Promises

**Day 3: Components and Templates**

* Creating Components with CLI
* Component Lifecycle Hooks
* Binding: Property, Event, Two-Way ([(ngModel)])
* Template Syntax: Interpolation, Directives (\*ngIf, \*ngFor)

**Day 4: Services and Dependency Injection**

* Creating and using services
* Dependency Injection basics
* Understanding @Injectable
* Sharing data between components

**Day 5: Routing, Navigation, and Animations**

* Setting up routing with RouterModule
* Configuring Routes
* Navigating programmatically
* Route Guards: CanActivate, CanDeactivate
* Creating animations using Angular's Animation API (@angular/animations)
* Transitioning between states and keyframes

**Day 6: Forms in Angular**

* Template-Driven Forms
* Reactive Forms: FormGroup, FormControl
* Validation: Built-in and Custom Validators
* Error Handling in Forms

**Day 7: HTTP Client**

* Setting up HttpClientModule
* Making HTTP requests: GET, POST, PUT, DELETE
* Handling Responses and Errors
* Interceptors for HTTP requests

**Day 8: Directives, Pipes, and Error Handling**

* Structural Directives (\*ngIf, \*ngFor)
* Attribute Directives ([ngStyle], [ngClass])
* Creating Custom Directives
* Built-in Pipes and Custom Pipes
* Centralized error handling with ErrorHandler
* Logging errors to external services (e.g., Sentry, LogRocket)
* Showing user-friendly error messages

**Day 9: State Management and Progressive Web Apps (PWA)**

* What is State Management?
* Introduction to RxJS: Observables and Subjects
* Using BehaviorSubject for component communication
* Introduction to NgRx Store
* Adding PWA capabilities using Angular CLI
* Service Workers and Offline Support
* Caching strategies with @angular/service-worker

**Day 10: Advanced RxJS and Backend Integration**

* Operators: map, filter, switchMap, mergeMap
* Combining Streams with forkJoin, combineLatest
* Managing Backpressure with throttleTime and debounceTime
* Error Handling in Observables
* Async Pipe in Templates
* Best practices for integrating Angular with REST APIs and GraphQL
* Using Apollo Client for GraphQL integration
* WebSocket integration for real-time updates

**Day 11: Angular Modules and Micro-Frontends**

* Feature Modules and Lazy Loading
* Shared Modules and Core Modules
* NgModule metadata: Declarations, Imports, Exports
* Understanding the concept of Micro-Frontends
* Using tools like Module Federation in Angular

**Day 12: Testing**

* Unit Testing with Jasmine and Karma
* Writing Component Tests
* Testing Services with Mock Dependencies
* Mocking services with HttpTestingController
* Testing Observables
* Writing tests for Directives and Pipes
* End-to-End Testing with Protractor or Cypress

**Day 13: Performance Optimization**

* Change Detection Strategies (OnPush)
* Lazy Loading and Preloading Strategies
* Reducing Bundle Size: Tree Shaking, AOT Compilation
* Using Angular Universal for Server-Side Rendering

**Day 14: Angular Material, UI Libraries, and Common Tools**

* Installing Angular Material
* Using Components: Buttons, Cards, Dialogs, Tables
* Theming and Customization
* Integrating Other Libraries (e.g., PrimeNG, Bootstrap)
* Using chart libraries like Chart.js, D3.js
* File upload and drag-drop support with ngx-dropzone
* PDF generation and export using jsPDF

**Day 15: Building and Deployment**

* Building the Application with Angular CLI
* Environment Configurations
* Deploying to Firebase or AWS
* Best Practices for Production Builds
* Documentation and final testing
* Building a Full-Featured Application:
  + CRUD Operations
  + Routing and Guards
  + Form Validation
  + State Management
  + HTTP Communication
  + Integration with UI Libraries
  + Deployment

**Checklist**

**Introduction to Angular**

☑ What is Angular?  
☑ Angular CLI and Installation  
☑ Setting up a Development Environment  
☑ Creating the First Application  
☑ Understanding Project Structure

**TypeScript Basics**

☑ Variables and Data Types  
☑ Functions and Arrow Functions  
☑ Classes and Interfaces  
☑ Promises and Async/Await  
☑ Decorators in TypeScript

**Angular Fundamentals**

**Components**

☑ Creating Components with Angular CLI  
☑ Component Lifecycle Hooks  
☑ Property and Event Binding  
☑ Two-Way Binding ([(ngModel)])  
☑ Using Directives (\*ngIf, \*ngFor)

**Services and Dependency Injection**

☑ Creating and Using Services  
☑ Understanding @Injectable  
☑ Dependency Injection Hierarchy  
☑ Sharing Data Between Components

**Routing and Navigation**

☑ Setting Up Routing with RouterModule  
☑ Configuring and Using Routes  
☑ Navigating Programmatically  
☑ Route Guards: CanActivate, CanDeactivate

**Animations**

☑ Creating animations using Angular's Animation API (@angular/animations)  
☑ Transitioning between states  
☑ Keyframe animations

**Forms in Angular**

☑ Template-Driven Forms  
☑ Reactive Forms  
☑ FormGroup and FormControl  
☑ Form Validation (Built-in and Custom Validators)  
☑ Error Handling in Forms

**HTTP and Observables**

☑ Using HttpClientModule  
☑ Making HTTP Requests  
☑ Handling Responses and Errors  
☑ Using Interceptors  
☑ RxJS Basics: Observables, Subjects  
☑ Advanced RxJS Operators  
☑ Managing Backpressure (throttleTime, debounceTime)  
☑ Error Handling in Observables

**Directives and Pipes**

☑ Structural Directives (\*ngIf, \*ngFor)  
☑ Attribute Directives ([ngStyle], [ngClass])  
☑ Creating Custom Directives  
☑ Built-in Pipes  
☑ Creating Custom Pipes

**State Management and Advanced Features**

☑ Introduction to State Management  
☑ Using RxJS for State Management  
☑ BehaviorSubject for Component Communication  
☑ NgRx Store Basics  
☑ Progressive Web Apps (PWA)  
☑ Service Workers and Offline Support  
☑ Caching Strategies

**Testing and Optimization**

**Testing**

☑ Unit Testing with Jasmine and Karma  
☑ Writing Component and Service Tests  
☑ Testing Observables  
☑ Testing Directives and Pipes  
☑ End-to-End Testing with Protractor or Cypress

**Performance Optimization**

☑ Change Detection Strategies (OnPush)  
☑ Lazy Loading and Preloading  
☑ Reducing Bundle Size  
☑ Angular Universal for SSR

**UI Libraries and Deployment**

☑ Using Angular Material  
☑ Customizing Themes  
☑ Integrating Other Libraries  
☑ Chart Libraries: Chart.js, D3.js  
☑ File Upload with ngx-dropzone  
☑ PDF Generation with jsPDF  
☑ Building Applications for Production  
☑ Deploying to Firebase, AWS, or Other Platforms

**Final Enhancements**

☑ Build a Full-Featured Application:

* CRUD Operations
* Routing and Guards
* Form Validation
* State Management
* HTTP Communication
* Integration with UI Libraries
* Deployment