Angular Full Course — Table of Contents

Beginner Level

Module 1: Introduction to Programming and Web Development

- What is Programming?
- How Computers Work (Basic Overview)
- What is Web Development?
- Introduction to Frontend vs Backend
- What are Websites, Web Servers, and Browsers?
- Introduction to HTML, CSS, JavaScript (basic mention, detailed later)

Module 2: Introduction to JavaScript (for Angular)

- What is JavaScript? Why learn it?
- Variables, Data Types, Operators
- Conditional Statements (if, else, switch)
- Loops (for, while, do-while)
- Functions and Scope
- Arrays and Objects
- Introduction to ES6+ Features (let/const, arrow functions, template literals)

Module 3: Fundamentals of TypeScript

- What is TypeScript? Why Angular uses it?
- Setting up TypeScript
- Types (number, string, boolean, any, unknown, etc.)
- Functions and Typing
- Interfaces and Types
- Classes and Objects
- Generics (basic understanding)

Module 4: Getting Started with Angular

- What is Angular? Why Angular?
- Understanding SPA (Single Page Applications)
- Installing Node.js and Angular CLI
- Creating First Angular App
- Understanding Angular Project Structure
- Running Angular Apps (ng serve)
- Introduction to Angular Modules

Module 5: Angular Basics - Components & Templates

- What is a Component?

- Creating Components (CLI)
- Component Decorators
- Templates (inline vs external)
- Data Binding Overview
  - Interpolation
  - Property Binding
  - Event Binding
  - Two-Way Binding
- Understanding Component Interaction (Input, Output basics)

Module 6: Basic Directives and Pipes

- What are Directives?
- *ngIf, *ngFor, *ngSwitch
- Attribute Directives (basic)
- Introduction to Pipes
  - Built-in Pipes (date, uppercase, lowercase, currency)

Mini-Project : Simple Product List App

(Using Components, Binding, Directives)
Build a basic product listing showing a list of products using *ngFor, with options to show/hide details using *ngIf.

Intermediate Level

Module 7: Services and Dependency Injection

- What are Services?
- Creating a Service
- Injecting Services into Components
- Introduction to Dependency Injection (DI) Pattern
- Singleton Services in Angular

Module 8: Routing and Navigation

- What is Routing?
- Configuring Routes
- RouterLink and RouterOutlet
- Route Parameters
- Nested Routes
- Lazy Loading Basics

Module 9: Forms in Angular

- Template-driven Forms
    - ngModel, Form Validation
- Reactive Forms

    - FormControl, FormGroup
- Form Validation Techniques (built-in and custom)

Module 10: HTTP and APIs

- What is HTTP?
- HttpClientModule Setup
- Making GET, POST requests
- Handling Responses
- Error Handling in HTTP
- Using environment files for API URLs

Mini-Project : User Management System

(Using Forms, Routing, HTTP Calls)
Create a simple CRUD User Management app to Add, View, Edit, and Delete users by interacting with a mock REST API.

Advanced Level

Module 11: Advanced Components Communication

- @Input() and @Output() Deep Dive
- ViewChild and ContentChild
- Services as Communication Bridges

- Shared Modules

Module 12: State Management Basics

- What is State Management?
- Introduction to Observables and RxJS
- BehaviorSubject for State Management (basic)
- Introduction to NgRx Store (very basic)

Module 13: Lazy Loading and Module Optimization

- Feature Modules
- Lazy Loading Feature Modules
- Preloading Strategies
- Route Guards (AuthGuard basics)

Module 14: Authentication and Authorization

- Introduction to JWT
- Setting up Authentication Flow
- Protecting Routes
- Token Storage (Local Storage/Session Storage)

Module 15: Angular Best Practices and Advanced Topics

- Angular Folder Structure Best Practices
- Performance Optimizations
- Error Handling and Logging
- Introduction to Unit Testing (very basic overview using Jasmine/Karma)
- Introduction to Angular Universal (Server-side rendering basics)

Mini-Project : Secure Blogging Platform

(Authentication, Lazy Loading, API Integration)
- Build a blogging platform with login, protected routes, creating and managing blog posts.

Final Projects (Capstone)

Final Project 1: E-Commerce Website

Build a full-fledged e-commerce website with product listing, cart management, user authentication, and payment mock flow.

Final Project 2: Project Management Dashboard

Create a dashboard that tracks projects, tasks, deadlines, with CRUD operations and authentication.

Final Project 3: Social Media App (Mini-Facebook)

Build a social platform where users can create profiles, post updates, like and comment on posts, with real-time updates using RxJS.