Angular 19

Duration: 5 Days

Course Description:

Angular 19 course equips participants with essential skills to build dynamic web applications. Participants will learn the fundamentals of components, services, and modules, enabling them to create robust user interfaces and optimize application performance. Key learning objectives include mastering the Angular CLI, handling HTTP requests, and implementing routing for seamless navigation. Through hands-on projects, students will apply these concepts in real-world scenarios, fostering practical experience and enhancing problem-solving abilities. By the end of the course, participants will be proficient in developing efficient applications that meet modern web standards, setting them up for success in the ever-evolving field of web development.

Target Audience:

Angular 19 is an advanced framework for building dynamic web applications, catering to developers seeking to enhance their skills and proficiency in modern web development techniques.

- Frontend Developers
- Web Application Developers
- Full Stack Developers
- Software Engineers
- UI/UX Designers
- Technical Project Managers
- IT Professionals transitioning into Development
- Computer Science Students
- Software Architects
- Quality Assurance Engineers
- DevOps Engineers
- Mobile App Developers (using Angular)
- Business Analysts with a tech focus

Prerequisites:

Minimum Required Prerequisites for Angular 19 Course:

To ensure a smooth and effective learning experience in the Angular 19 course, we recommend that students have the following minimum knowledge and skills:

- Basic understanding of HTML, CSS, and JavaScript: Familiarity with these core web technologies will help you grasp Angular concepts more easily.
- Experience with TypeScript: A fundamental knowledge of TypeScript is beneficial, as Angular is built using this superset of JavaScript.
- Familiarity with basic programming concepts: Understanding of variables, data types, functions, and object-oriented programming will aid in learning Angular's structure and functionality.

These prerequisites will set a solid foundation for your journey into Angular 19, allowing you to engage fully with the course content and maximize your learning experience.

Course Objectives:

- Understand the architecture of Angular and its core concepts.
- Create and manage components and modules effectively.
- Implement data binding and handle user inputs.
- Utilize services and dependency injection.
- Work with Angular routing and navigation.
- Fetch and manipulate data using HTTP requests.
- Employ reactive programming with RxJS for handling asynchronous data.
- Implement form management using reactive and template-driven approaches.
- Optimize application performance and troubleshoot common issues.
- Understand deployment best practices for Angular applications.

Course Outlines:

Day 1: Introduction to Angular & Setting Up the Environment

Introduction to Angular
- What is Angular?
- Evolution of Angular (AngularJS to Angular 19)
- Advantages of Angular
- Real-world applications

Setting Up the Development Environment
- Installing Node.js, Angular CLI
- Creating the first Angular application
- Understanding project structure
- Running an Angular application

Understanding Angular Core Concepts
- Modules (NgModule)
- Components (@Component)
- Templates and Views
- Directives (*ngIf, *ngFor, [ngClass], [ngStyle])

Hands-on Labs

Lab 1: Setting Up Angular Project
- Install Angular CLI and create a new Angular 19 project
- Explore project structure and run the app
- Modify app.component.ts and app.component.html

Lab 2: Creating Components & Using Directives
- Create new components using CLI
- Implement *ngIf and *ngFor
- Apply ngClass and ngStyle

Day 2: Components, Data Binding & Services

Component Communication
- @Input and @Output
- Parent-child communication
- Event binding and property binding

Data Binding in Angular
- Interpolation ({{ }})
- One-way and two-way binding (ngModel)
- Property binding
- Event binding

Angular Services & Dependency Injection (DI)
- Creating services with @Injectable
- Understanding Dependency Injection
- Using HttpClientModule for API calls

Hands-on Labs

Lab 3: Component Communication
- Create parent and child components
- Implement @Input and @Output

Lab 4: Services & API Calls
- Create an Angular service
- Fetch and display data using HttpClient

Day 3: Routing, Forms & State Management

Routing & Navigation
- Configuring routes with RouterModule
- Lazy loading & route guards
- Handling route parameters

Forms in Angular
- Template-driven forms
- Reactive forms (FormGroup, FormControl, Validators)

Hands-on Labs

Lab 5: Implementing Routing
- Create routes and use RouterModule
- Implement route guards

Lab 6: Forms Handling
- Build a form using Reactive Forms
- Implement form validation

Day 4: Advanced Topics

State Management with NgRx (Redux Pattern)
- Introduction to NgRx Store
- Actions, Reducers, Selectors, Effects

Directives & Pipes
- Built-in and custom pipes
- Custom directives

Hands-on Labs

Lab 7: State Management with NgRx
- Set up NgRx store
- Manage state with actions and reducers

Lab 8: Creating Custom Directives & Pipes
- Implement a custom directive
- Create a custom pipe

Day 5: Testing, Optimization & Deployment

Unit Testing in Angular
- Writing tests using Jasmine & Karma
- Testing services, components, and directives

Performance Optimization
- Lazy loading modules
- Change Detection Strategy
- Best practices for production apps

Deployment of Angular Apps
- Building & optimizing the production version
- Hosting on Firebase, Netlify, or Vercel

Lab 9: Writing Unit Tests
- Write test cases for components & services

Lab 10: Deploying Angular App
- Deploy an Angular app to Firebase or Netlify