VTechLabs

Angular

SYLLABUS

Course Overview

Prerequisites: Basic HTML, CSS & JavaScript knowledge

Framework Version: Angular 17+

This Angular syllabus program is designed to take beginners from zero Angular knowledge to building real-world, production-ready single-page applications (SPAs). The course follows a hands-on, project-based approach where learners build upon each concept progressively across 8 structured weeks.

By the end of this syllabus, participants will have a solid foundation in Angular's core architecture, component-driven development, data binding, routing, services, HTTP communication, and state management.

Learning Objectives

Upon completion of this course, trainees will be able to:

- Understand Angular's architecture and core building blocks
- Create and manage Angular components, modules, and templates
- Implement data binding, directives, and pipes effectively
- Build and consume RESTful APIs using Angular's HttpClient
- Design and implement client-side routing with Angular Router
- Create and validate reactive and template-driven forms
- Write services and leverage Dependency Injection
- Manage application state with RxJS and Observables
- Write unit tests using Jasmine and Karma
- Deploy a complete Angular application

Target Audience

- Frontend developers transitioning from other frameworks
- Fresh graduates and bootcamp graduates with JS knowledge
- Backend developers wanting to learn frontend SPA development
- Developers with jQuery or vanilla JS experience

Tools & Environment Setup

- Node.js (v18+) and npm
- Angular CLI (ng command-line tool)
- Visual Studio Code with Angular Language Service extension
- Chrome DevTools & Angular DevTools browser extension
- Git & GitHub for version control

TypeScript & Angular Foundations

Module 1.1: TypeScript Essentials

Why TypeScript for Angular?

- Statically typed superset of JavaScript
- Better IDE support, refactoring, and error detection
- Required for Angular development

Core TypeScript Concepts

- Types: string, number, boolean, any, void, never, unknown
- Interfaces and Type Aliases
- Classes, Inheritance, Access Modifi ers (public, private, protected)
- Generics and Type Inference
- Decorators Overview
- ES6+ features: Arrow functions, destructuring, spread, modules

Module 1.2: Angular Architecture & CLI

Angular Overview

- What is Angular and why use it?
- Angular vs React vs Vue — when to choose Angular
- Angular versioning and LTS releases

Angular CLI

- Installing Angular CLI globally
- Generating a new project: ng new my-app
- Understanding project folder structure
- Key fi les: angular.json, tsconfi g.json, package.json
- Running the dev server: ng serve
- Building for production: ng build

Hands-On Lab 1

Set up a development environment, create the fi rst Angular project, explore the folder structure, and run the app in the browser.

Components, Templates & Data Binding

Module 2.1: Angular Components

- Component anatomy: @Component decorator, selector, template, styles
- Generating components using CLI: ng generate component
- Inline vs external templates and styles
- Component lifecycle hooks overview
  - ngOnInit — initialization logic
  - ngOnChanges — input property changes
  - ngOnDestroy — cleanup subscriptions
  - ngAfterViewInit — DOM ready

Module 2.2: Data Binding

- Interpolation: {{ expression }}
- Property Binding: [property]="expression"
- Event Binding: (event)="handler($event)"
- Two-Way Data Binding: [(ngModel)] with FormsModule
- Understanding change detection

Module 2.3: Angular Modules (NgModules)

- What is NgModule and why it matters
- declarations, imports, exports, providers, bootstrap
- Root module (AppModule) vs feature modules
- Standalone components in Angular 17+

Hands-On Lab 2

Build a product card component with input properties. Implement a simple counter with event binding and two-way binding.

Directives, Pipes & Component Communication

Module 3.1: Built-in Directives

- Structural Directives
  - *ngIf / @if — conditional rendering
  - *ngFor / @for — list rendering with index and trackBy
  - *ngSwitch — multi-condition branching
- Attribute Directives
  - ngClass — dynamic CSS class binding
  - ngStyle — dynamic inline styles
- New Control Flow Syntax (@if, @for, @switch) in Angular 17+

Module 3.2: Custom Directives

- Creating structural and attribute directives
- HostListener and HostBinding
- ElementRef and Renderer2 for DOM manipulation

Module 3.3: Pipes

- Built-in pipes: DatePipe, CurrencyPipe, DecimalPipe, UpperCasePipe, LowerCasePipe, AsyncPipe
- Chaining pipes
- Creating custom pipes with PipeTransform
- Pure vs impure pipes

Module 3.4: Component Communication

- Parent -> Child: @Input() decorator
- Child -> Parent: @Output() and EventEmitter
- ViewChild and ContentChild
- Sibling communication through shared services

Hands-On Lab 3

Build a fi lterable product list using *ngFor, *ngIf, and a custom search pipe. Implement parent-child communication for a shopping cart counter.

Services, Dependency Injection & HTTP

Module 4.1: Services & Dependency Injection

- What are Angular services and why use them?
- Creating services: ng generate service
- @Injectable() decorator and providedIn: 'root'
- DI hierarchy: root, module-level, component-level
- Singleton pattern in Angular services
- InjectionToken for non-class dependencies

Module 4.2: HTTP Client

- Importing HttpClientModule / provideHttpClient()
- GET, POST, PUT, DELETE requests
- HttpHeaders, HttpParams
- Handling responses and errors with RxJS operators
  - map, catchError, tap, switchMap, mergeMap
- HTTP Interceptors — auth tokens, logging, error handling
- Environment variables for API base URLs

Module 4.3: Async Patterns

- Promises vs Observables
- subscribe() pattern
- Unsubscribing and takeUntil Destroyed
- async pipe for template subscriptions

Hands-On Lab 4

Build a User Directory app that fetches user data from JSONPlaceholder API. Display users in cards with error handling and loading state.

Angular Router & Navigation

Module 5.1: Routing Fundamentals

- Setting up Angular Router
- Defi ning routes: path, component, redirectTo, pathMatch
- <router-outlet> placement
- RouterLink and routerLinkActive directives
- Programmatic navigation with Router.navigate()

Module 5.2: Route Parameters & Guards

- Route parameters: /product/:id
- ActivatedRoute: snapshot vs paramMap observable
- Query parameters and fragments
- Route guards
  - CanActivate — protecting routes
  - CanDeactivate — preventing unsaved navigation
  - Resolve — pre-loading data
  - CanLoad — lazy loading guard

Module 5.3: Lazy Loading & Code Splitting

- Feature modules and lazy loading
- loadChildren and loadComponent
- Preloading strategies: PreloadAllModules, custom
- Analyzing bundle size with webpack-bundle-analyzer

Hands-On Lab 5

Build a multi-page blog application with a post list, post detail, about, and 404 page. Add an auth guard protecting the admin route.

Angular Forms

Module 6.1: Template-Driven Forms

- FormsModule and NgModel
- Two-way binding with [(ngModel)]
- Template reference variables (#name)
- Built-in validators: required, minlength, maxlength, email, pattern
- Showing validation errors in templates
- Form submission with ngSubmit

Module 6.2: Reactive Forms

- ReactiveFormsModule
- FormControl, FormGroup, FormArray
- FormBuilder service
- Programmatic validation with Validators class
- Custom synchronous and asynchronous validators
- Dynamic form fi elds with FormArray
- valueChanges and statusChanges observables

Module 6.3: Form UX Best Practices

- Touched, dirty, and pristine states
- Disabling submit button until form is valid
- Cross-fi eld validation
- Resetting forms programmatically

Hands-On Lab 6

Build a multi-step registration form with Reactive Forms. Include custom validators for password confi rmation and async username availability check.

RxJS, State Management & Testing

Module 7.1: RxJS Deep Dive

- Observable, Observer, Subscription
- Subject, BehaviorSubject, ReplaySubject
- Creation operators: of, from, interval, timer, fromEvent
- Transformation operators: map, switchMap, mergeMap, concatMap, exhaustMap
- Filtering operators: fi lter, debounceTime, distinctUntilChanged, takeUntil
- Combination operators: combineLatest, forkJoin, zip, merge
- Error handling: catchError, retry, retryWhen
- Practical: search-as-you-type with debounceTime + switchMap

Module 7.2: State Management Basics

- Component state vs shared application state
- BehaviorSubject-based state service pattern
- Introduction to NgRx
  - Store, Actions, Reducers, Selectors, Effects
  - When to use NgRx vs simple services
- NgRx Signals Store (Angular 17+ approach)

Module 7.3: Testing Angular Applications

- Unit testing with Jasmine & Karma
- TestBed confi guration
- Testing components: fi xture, nativeElement, DebugElement
- Testing services with dependency injection
- Mocking HTTP calls with HttpClientTestingModule
- Integration tests and async testing with fakeAsync, tick
- Introduction to E2E testing with Cypress

Hands-On Lab 7

Implement a real-time search feature using RxJS debounceTime and switchMap. Write unit tests for a service and a component.

Performance, Deployment & Capstone Project

Module 8.1: Performance Optimization

- Change detection strategies: Default vs OnPush
- TrackBy in *ngFor for list performance
- Lazy loading images and routes
- Deferrable Views (@defer) in Angular 17+
- Virtual scrolling with CdkVirtualScrollViewport
- Memoization with pure pipes
- Server-Side Rendering (SSR) with Angular Universal — overview

Module 8.2: Build & Deployment

- Production build: ng build --confi guration production
- Environment confi gurations (environment.ts)
- Deploying to Firebase Hosting with ng deploy
- Deploying to Netlify and Vercel
- Setting up CI/CD pipeline overview (GitHub Actions)

Module 8.3: Capstone Project

Project: Task Management Application

Trainees will build a complete Task Management (Kanban-style) application applying all concepts learned throughout the course.

Required Features

- User authentication (login/register UI with guards)
- CRUD operations for tasks and boards
- Drag-and-drop task movement between columns (using CDK)
- Real-time search and fi lter tasks
- HTTP integration with a mock API (json-server)
- Reactive Forms for task creation and editing
- Lazy-loaded feature modules
- OnPush change detection on list components
- Unit tests for at least 3 components/services
- Deployed to Firebase Hosting

Evaluation Criteria

- Code quality and Angular best practices
- Correct use of components, services, and routing
- Form validation completeness
- Test coverage and quality
- UI/UX and responsive design
- Successful deployment

Recommended Resources

Official Documentation

- Angular Official Docs — angular.dev
- TypeScript Handbook — typescriptlang.org/docs
- RxJS Documentation — rxjs.dev

Books

- "Angular Development with TypeScript" — Yakov Fain & Anton Moiseev
- "ng-book: The Complete Guide to Angular" — Nathan Murray et al.
- "Reactive Programming with RxJS" — Sergi Mansilla

Online Platforms

- Udemy — Maximilian Schwarzmuller's Angular Complete Guide
- Pluralsight — Angular Learning Path
- Frontend Masters — Angular courses
- YouTube — Fireship, Decoded Frontend, Kevin Powell

Practice Platforms

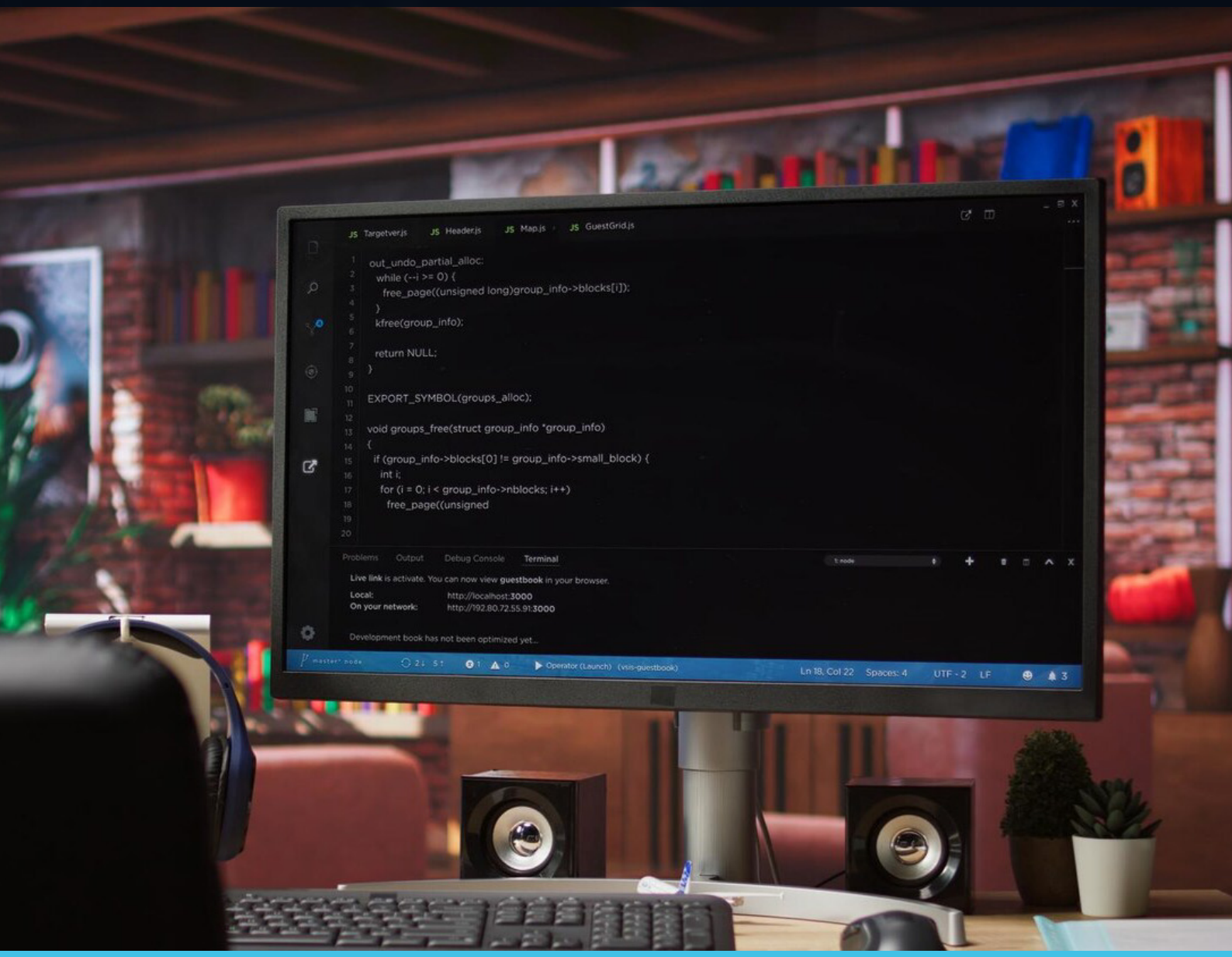
- StackBlitz — Online Angular IDE (stackblitz.com)
- CodeSandbox — Browser-based development
- GitHub — Hosting course projects

Trainer Notes & Tips

The following tips have been compiled to help trainers deliver the most effective sessions:

- Start each week with a quick recap quiz to reinforce previous content
- Use live coding extensively — trainees learn best by watching, then doing
- Encourage trainees to read Angular's official changelog to build habits
- Pair-program during labs when possible for collaborative learning
- Dedicate the last 15 minutes of each session to Q&A and code review
- Provide a starter GitHub repo for each lab to reduce setup friction
- Cover Angular DevTools early — it dramatically accelerates debugging skills
- Slack/Discord channel for async Q&A between sessions is highly recommended

VTechLabs



alok@webmyne.com

+91 94276 02525

702, Ivory Terrace , Opp. Circuit House
R.C. Dutt Road, Vadodara-07
Gujarat - India.