6 Months Training – TR-104 Full Stack Development

January 2026
Day 1 – Angular Introduction & Environment Setup

The objective of Day 1 was to gain a clear understanding of Angular as a modern front-end framework and to set up a professional development environment required for Angular application development. The session aimed to build a strong foundation by introducing Angular concepts from scratch, following the latest version and official best practices.

Introduction to Angular Framework:

The training session began with an introduction to Angular, a modern front-end framework developed and maintained by Google. Angular is used for building dynamic, single-page web applications (SPAs) that provide fast and interactive user experiences.

Angular was explained as a TypeScript-based framework, meaning it uses TypeScript instead of plain JavaScript. TypeScript provides features such as strong typing, interfaces, and decorators, which help in writing structured, maintainable, and error-free code.

The concept of Single Page Applications (SPA) was discussed in detail. In an SPA, the application loads only once in the browser, and subsequent page updates happen dynamically without reloading the entire page. Angular achieves this by updating only the required parts of the UI when data changes.

Component-Based Architecture:

Angular follows a component-based architecture, where the application is divided into small, reusable, and independent units called components. Each component is responsible for:

- Displaying a part of the user interface
- Handling its own logic
- Managing its own styles

This architecture improves:

- Code reusability
- Maintainability
- Scalability of large applications

Unlike traditional web development where HTML, CSS, and JavaScript are written separately without structure, Angular binds all three together inside components.

Difference Between Traditional Web Development and Angular:

The differences between traditional web development and Angular were explained:

Traditional Web Development	Angular
Manual DOM manipulation	Automatic UI updates
JavaScript handles logic directly	TypeScript with structure
Multiple page reloads	Single Page Application
Less scalable for large apps	Highly scalable architecture

Angular automatically synchronizes the UI with application data using data binding, eliminating the need for manual DOM manipulation.

Environment Setup for Angular Development:

The next major part of the session focused on setting up the development environment required to work with Angular.

1. Node.js and npm

- Node.js is installed and verified.
- npm (Node Package Manager) was introduced as a tool used to install and manage project dependencies.
- The role of npm in installing Angular CLI and external libraries was explained.

2. Angular CLI Installation

Angular CLI (Command Line Interface) was installed using npm. Angular CLI simplifies:

- Project creation
- Running development servers
- Building applications
- Generating components and services

3. Creating an Angular Project

Using Angular CLI, a new Angular project was created following official documentation guidelines. During project creation:

- Project structure was automatically generated
- Configuration files were created
- Development dependencies were installed

Understanding the Development Server

The command ng serve was used to run the Angular development server. The purpose of the development server was explained:

- It compiles the application
- Runs it on a local server
- Automatically reloads the browser when code changes

This feature improves development speed and efficiency.

Introduction to Modern Angular Project Structure (Angular 17+)

The project structure of modern Angular was explored in detail. Unlike older Angular versions that relied heavily on NgModules, the latest Angular versions use standalone components, which simplify application architecture.

Key files studied included:

- **app.ts** – Contains component logic and behavior
- **app.html** – Defines the user interface
- **app.css** – Handles component-specific styling
- **app.routes.ts** – Manages application routing

This structure reflects modern Angular best practices and reduces unnecessary complexity.

Development Tools Used

The development was carried out using Visual Studio Code (VS Code). The integrated terminal of VS Code was used to run Angular CLI commands. The importance of:

- Code editor extensions
- Proper file organization
- Clean project structure
  was emphasized.

Practical Tasks Performed

During the session, the following hands-on tasks were completed:

- Installed and verified Node.js and npm
- Installed Angular CLI successfully
- Created a new Angular project using CLI
- Explored project folders and configuration files
- Ran the application using ng serve
- Viewed the application in the browser

These tasks helped in gaining confidence with Angular setup and execution.

Key Learnings :

- Angular is a TypeScript-based front-end framework
- Angular applications follow a component-based architecture
- Single Page Applications improve performance and user experience
- Angular CLI simplifies development workflow
- Modern Angular uses standalone components
- Development servers support live reloading
- Proper environment setup is essential for smooth development

Conclusion:

Day 1 of the training successfully established a strong foundation for Angular development. By understanding Angular's purpose, architecture, and development environment, the session prepared participants to move forward with core Angular concepts such as components, data binding, and state management. The hands-on approach ensured practical understanding and readiness for advanced topics.