**Angular Learning Report: Week 1 & 2 Presentation Summary**

## 📅 Week 1: 21 JUL – 27 JUL

### ✅ Key Concepts Covered:

* **Angular Architecture**: Understanding how Angular is built using components, modules, and services.
* **Components**: Created multiple components to build a structured UI.
* **Modules**: Used NgModule to organize and register components.
* **Services & Dependency Injection**: Created reusable services and injected them into components.
* **Angular CLI**: Generated components and services, served the application using ng serve, and built the app with ng build.

### 📊 Accomplishments:

* Set up Angular 17 project with Tailwind CSS and global fonts.
* Created a Todo App skeleton using multiple components.
* Implemented routing and used FormsModule.

### 🎯 Learning Outcome:

Gained a clear understanding of Angular’s foundational structure and how to scaffold and configure projects using Angular CLI.

## 📅 Week 2: 28 JUL – 3 AUG

### ✅ Key Concepts Covered:

* **Structural Directives**: Used \*ngIf, \*ngFor, and \*ngSwitch to control DOM layout.
* **Attribute Directives**: Used ngClass, ngStyle, and created a custom directive using @Directive.
* **DOM Manipulation**: Manipulated styles using Renderer2 for safer DOM changes.
* **Data Binding**:
  + Interpolation ({{ title }})
  + Property Binding ([value])
  + Event Binding ((click))
  + Two-Way Binding ([(ngModel)])

### 📊 Accomplishments:

* Built a live character counter component.
* Learned the difference between one-way and two-way data binding.
* Answered tricky interview questions related to directives and binding.

### 🎯 Learning Outcome:

Developed deep knowledge of how Angular templates interact with component logic and how DOM behavior can be dynamically controlled.

## 🌟 Summary:

Over the first two weeks, I successfully:

* Understood Angular’s architecture and project structure.
* Mastered component-based UI creation.
* Explored powerful directive systems to control DOM layout and behavior.
* Connected component logic to templates using binding techniques.

These fundamentals have prepared me to move forward into complex topics like **Forms, Validation, HTTP Requests, and RxJS** in the coming weeks.

**Next Step:** ➡ Begin **Week 3: Forms (Template-driven & Reactive)** with hands-on form validation and form-building exercises.