LessonPlanningNo	PlanningTitle	PlanningDescription
1	Perform TypeScript Programs	1. Print Hello world using TypeScript (A) 2. WAP in TypeScript to perform basic arithmatic operations (A) 3. WAP to perform basic operations on Array in TypeScript. (A) 4. WAP to define and call user defined functions in TypeScript. (A)
2	Perform TypeScript Programs	1. WAP to demonstrate the use of interface in TypeScript. (A) 2. WAP to demonstrate the use of classes in TypeScript. (A) 3. WAP to perform String manipulation in TypeScript. (A) 4. WAP to perform Type Assertion in TypeScript. (B)
3	Installation and project setup	1. Setup Environment for Angular. (A) 2. Create first project in Angular. (A) 3. Hello world webapp using Angular (A)
4	Create component	1. Demonstrate creating and using components in Angular (A) 2. Practice AngularCLI (B)
5	Demonstrate Interpolation, pipes and propert	1. Perform basic interpolation (A) 2. Perform interpolation with pipes (A) 3. Demonstrate property binding in Angular (A)
6	Demonstrate Event binding and Twoway data	1. WAP to handle key events in Angular. (A) 2. WAP to handle mouse events in Angular (A) 3. WAP to handle form events in Angular (A) 4. WAP to demonstrate twoway databinding (A) 5. WAP to create GUI calculator in Angular (B) 6. WAP to create static Resume Builder webapp (C)
7	Demonstrate Routing in Angular	1. WAP to create three components and perform basic routing (A) 2. WAP to demonstrate the use of route parameters (A) 3. WAP to create ten different components with proper layouting and create a static webapp. (C)
8	Practice Directives in Angualr - 01	1. Practice ngIf directive. (A) 2. Practice ngSwitch directive (A) 3. Practice ngClass directive (A) 4. Practive ngStyle direcive (A) 5. WAP in angular to calculate electricity bill, if the amount is below 1000 Rs it should display amount in green color and in red color otherwise. (B) 6. WAP in Angular to display grade of the student based on marks entered, if average marks of a student is less than 60 it should display pass class, it average marks is grater than 60 and less than 80 it should display first class and if marks are grater than 80 it should display topper class (B) 7. WAP to apply a specific class based on some condition in Angular (B)
9	Practice Directives in Angualr - 02	1. Practice ngFor directive. (A) 2. WAP to render array as ordered list in Angular. (A) 3. WAP to render array as table in Angular (B) 4. WAP to render array as card view from bootstrap in Angular (C) 5. WAP to render object containing array in Angular. (B)
10	use template driven forms	1. Create basic GUI calculator using Template driven forms. (A) 2. Perform CRUD operation of a static array containing students data using Template driven forms. (A). 3. Perform CRUD operation of a static array containing faculties data using Template driven forms. (B). 4. Perform CRUD operation of a static array containing books data using Template driven forms. (B).
11	perform validation in template driven forms	1. Validate data from basic GUI calculator created in the last lab. (A) 2. Validate data from all the forms that we have created in the last lab. (B)
12	use reactive forms	1. Create basic GUI calculator using reactive forms. (A) 2. Perform CRUD operation of a static array containing students data using reactive forms. (A). 3. Perform CRUD operation of a static array containing faculties data using reactive forms. (B). 4. Perform CRUD operation of a static array containing books data using reactive forms. (B).
13	perform validation in reactive forms	1. Validate data from basic GUI calculator created in the last lab. (A) 2. Validate data from all the forms that we have created in the last lab. (B)
14	Practice some more template driven forms in	1. Practice forms provided using Template Driven forms.
15	Practice some more reactive forms in Angular	1. Practice forms provided using Reactive forms.
16	Implement Deferrable view and SSR	1. Implement Deferrable view in Angular. (A) 2. Create an Optimization Report after Applying Deferrable view in Angular. (B) 3. Write a short note on how Deferrable view is effecting Core Web Vitals (CWV) results. © 4. Implement Server Side Rendering in Angular. (A)
17	Configure Static Site Genration (SSG)	1. Configure Static Site Genration (SSG) in Angular. (A) 2. Build and prerender site. (A) 3. Serve the static site. (A) 4. Check Performance analysis after applying SSG. (B) 5. Deploy the static site. (C)
18	Consuming REST API - 01	1. Consume GetAll api from mockapi. (A) 2. Consume GetAll api to display all students stored in MongoDB using api created with ExpressJS. (B) 3. Consume GetAll api for any three collection stored in MongoDB using api created with ExpressJS. (C)

19	Consuming REST API - 02	1. Consume GetByID and delete api from mockapi. (A) 2. Consume GetByID and delete api to display and delete student using id stored in MongoDB using api created with ExpressJS. (B) 3. Consume GetByID and delete api for any three collection stored in MongoDB using api created with ExpressJS. (C)
20	Consuming REST API - 03	1. Perform insert and update operation using mockapi. (A) 2. Perform insert and upadte operation on a faculties collection stored in MongoDB using ExpressJS api. (B) 3. Perform insert and upadte operation on any three collection stored in MongoDB using ExpressJS api. (C)
21	Authentication and route protection	1. Implement Authentication using JWT token. (A) 2. Implement Route Protection in Angular. (A)
22	Capstone Project - 01	Capstone Project
23	Capstone Project - 02	Capstone Project
24	Capstone Project - 03	Capstone Project
25	Capstone Project - 04	Capstone Project
26	Testing and deploying Angular Application	1. Perform Unit teseting on Capstone Project created with Angular (A) 2. Deploy Capstone Project to live server (A)