**Experiment No.: 06**

**Title:** Demonstrate services in angular

**Objectives:**

1. To demonstrate reusing the code in different components using services.

**Theory:**

In Angular, a service is a class that encapsulates a specific functionality or feature that can be shared across multiple components. Services are typically used to manage data, perform HTTP requests, handle business logic, or interact with external resources.

Dependency injection (DI) is a design pattern used in Angular to manage the dependencies of a class. With DI, Angular's injector subsystem automatically provides dependencies to a class when it is instantiated, rather than requiring the class to create or find its dependencies.

**Working of services and DI in angular:**

1. Creating a Service: To create a service in Angular, you create a class with the @Injectable() decorator. This decorator marks the class as a service and allows Angular to inject dependencies into it.

import { Injectable } from '@angular/core';

@Injectable({

providedIn: 'root'

})

export class DataService {

constructor() { }

getData() {

// Code to fetch data

}

}

**2. Injecting a Service:** To use a service in a component or another service, you simply declare a constructor parameter of the corresponding service type. Angular's DI system automatically injects an instance of the service when the component or service is instantiated.

import { Component } from '@angular/core';

import { DataService } from './data.service';

@Component({

standalone: true,

selector: 'app-my-component',

templateUrl: './my-component.component.html',

styleUrls: ['./my-component.component.css']

imports: [ NgFor, NgIf ],

providers: [ DataService]

})

export class MyComponent {

constructor(private dataService: DataService) { }

fetchData() {

this.dataService.getData();

}

}

Services and dependency injection in Angular promote modularity, reusability, and maintainability by enabling components to interact with shared functionality without directly managing dependencies.

**Key Concept:** Dependency Injection, Service

**Steps:**

1. Create Angular App with name my-first-app.

2. Create a new service using ng generate service name\_component

3. Inject this service in different components.

4. Check the result.