Experiment: Designing a Web Application Using Angular

Aim

To design and develop a simple web application using **Angular Framework**, demonstrating components, modules, data binding, services, and routing.

Requirements / Software Used

- Node.js (Latest LTS Version)
- Angular CLI
- Visual Studio Code / Any IDE
- Web Browser (Chrome Recommended)

Theory

Angular is a **TypeScript-based front-end framework** used to build dynamic single-page applications (SPAs). It uses components, templates, data binding, dependency injection, and services to structure applications efficiently.

Key Angular Concepts Used

- **Angular CLI** → used to create, build, and serve applications.
- **Components** → UI building blocks.
- **Modules** → group related components.
- **Services** → handle business logic & reusable code.
- **Two-way Data Binding** → sync UI and logic.
- **Routing** → navigate between pages.

Procedure

1. Install Angular CLI

```
npm install -g @angular/cli
```

2. Create a New Angular Project

```
ng new angular-demo
cd angular-demo
```

3. Create Components

Example components: `home`, `about`, `contact`

```
ng generate component home
ng generate component about
ng generate component contact
```

4. Configure Routing

Enable routing when prompted during project creation OR manually edit:

app-routing.module.ts

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { AboutComponent } from './about/about.component';
import { ContactComponent } from './contact/contact.component';

const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'about', component: AboutComponent },
  { path: 'contact', component: ContactComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

5. Create a Service for Data

```
ng generate service trip
```

trip.service.ts

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

@Injectable({ providedIn: 'root' })
export class TripService {
  trips = [
    { id: 1, name: 'Mountain Trip', season: 'Winter' },
    { id: 2, name: 'Beach Trip', season: 'Summer' }
  ];

  getTrips() {
    return this.trips;
  }
}
```

6. Display Data Using a Component

home.component.ts

```
import { Component, OnInit } from '@angular/core';
import { TripService } from '../trip.service';

@Component({
  selector: 'app-home',
  templateUrl: './home.component.html'
})
export class HomeComponent implements OnInit {
  trips: any = [];

  constructor(private tripService: TripService) {}

  ngOnInit(): void {
    this.trips = this.tripService.getTrips();
  }
}
```

home.component.html

```
<h2>Available Trips</h2>
<ul>
  <li *ngFor="let t of trips">
    {{ t.name }} - Season: {{ t.season }}
  </li>
</ul>
```

7. Add Navigation in App Component

app.component.html

```
<h1>Angular Web Application</h1>
<nav>
  <a routerLink="">Home</a> |
  <a routerLink="about">About</a> |
  <a routerLink="contact">Contact</a>
</nav>
<hr />
<router-outlet></router-outlet>
```

Output

- A working Angular web application
- Multiple pages with routing
- Trips displayed using a service
- Navigation between components

Result

A simple Angular web application was successfully developed using components, services, routing, and data binding.

If you want: - PDF export - Better UI using Angular Material - More modules like login, booking, dashboard - Flowchart / Diagram

Just tell me!