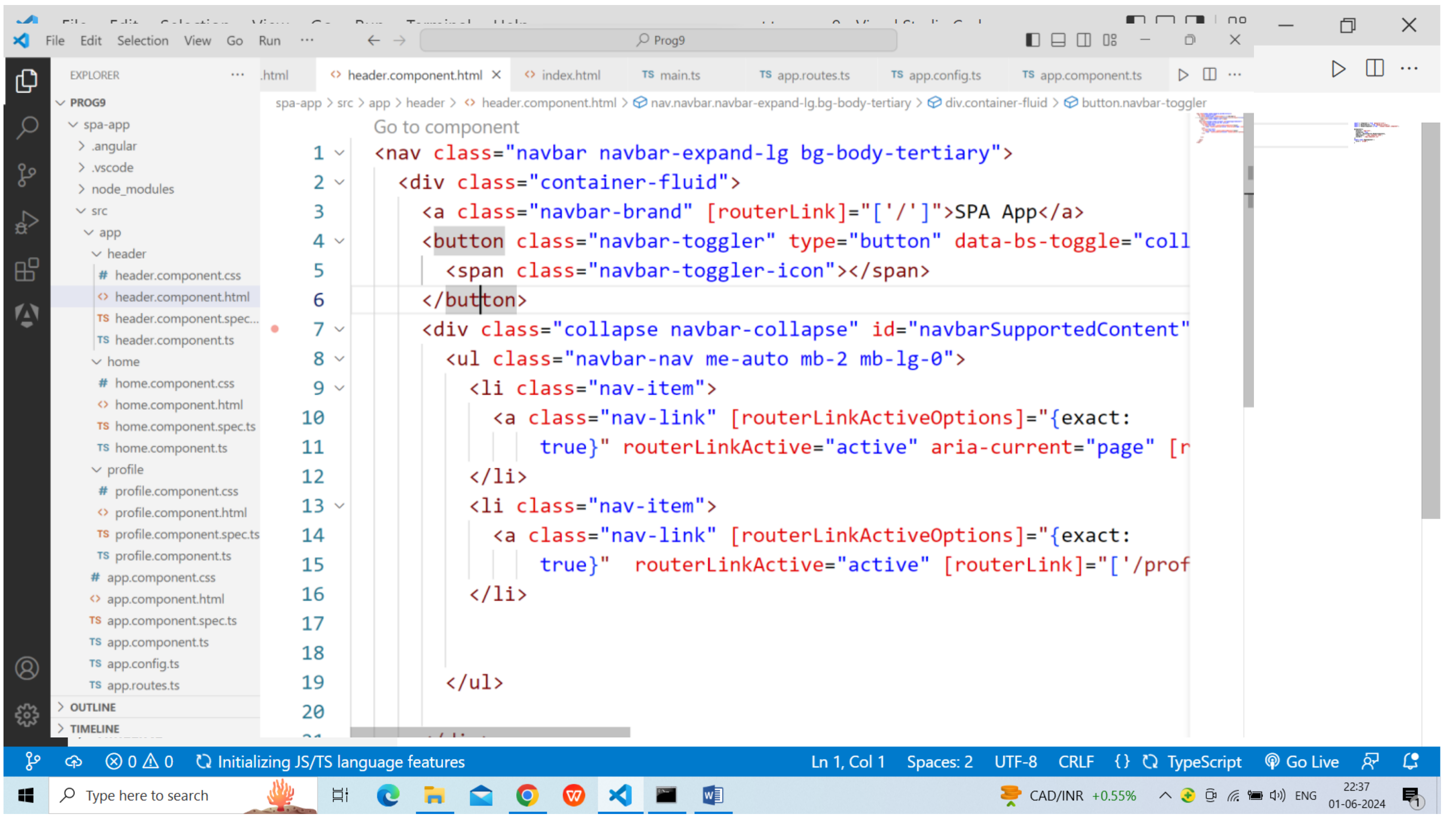
Week 9

Design a Single Page Application with different menu items using Angular.



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" [routerLink]="['/']">SPA App</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="coll
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent"
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link" [routerLinkActiveOptions]="{exact:
            true}" routerLinkActive="active" aria-current="page" [r
        </li>
        <li class="nav-item">
          <a class="nav-link" [routerLinkActiveOptions]="{exact:
            true}"  routerLinkActive="active" [routerLink]="['/prof
        </li>


      </ul>
```

Note:-Install Node Js Software and Visual Studio.

Creating Angular Project Use below Commands

- npm install –g @angular/cli //creating cli
- ng version
- ng new prog9 //creating angular project SPA(Single page application) //app component is a default component.
- cd prog9
- ng g c header //creating header component
- ng g c home //creating home component
- ng g c profile //creating profile component
- ng serve //running angular project

Step 1:-
Index.html:-
Add bootstrap CDN links in the index.html

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

Step 2:-Add RouterModule to the **header.component.ts** imports

```
import { RouterModule } from '@angular/router';

imports: [RouterModule],
```

Step 3:-Add RouterModule, HeaderComponent to the **app.component.ts** imports

```
import { RouterModule, RouterOutlet } from '@angular/router';
import { HeaderComponent } from './header/header.component';

imports: [RouterOutlet,RouterModule,HeaderComponent],
```

header.component.html:-

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
    <div class="container-fluid">
      <a class="navbar-brand" [routerLink]="['/']">SPA-Demo</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" [routerLinkActiveOptions]="{exact:true}" routerLinkActive="active"
            aria-current="page" [routerLink]="['/']">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" [routerLinkActiveOptions]="{exact:true}" routerLinkActive="active"
            [routerLink]="['/profile']">Profile</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

home.component.html

```
<section class="container mt-5 text-center">
    <h1>Welcome to SPA Demo Page</h1>
    </section>
```

profile.component.html

```
<section class="container mt-5 text-center">
    <h1>Welcome to Profile Page</h1>
</section>
```

app.component.html

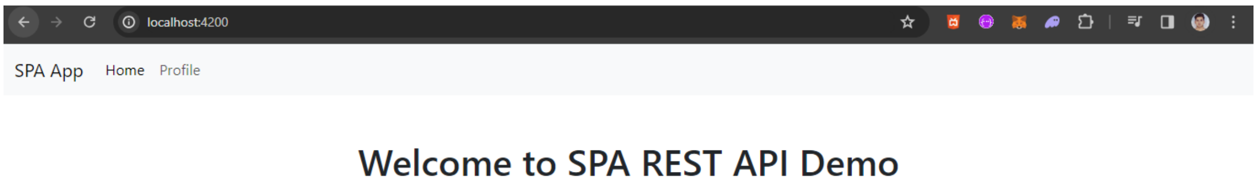
```
<app-header></app-header>
<router-outlet></router-outlet>
```

app.routes.ts

```
import { Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { ProfileComponent } from './profile/profile.component';

export const routes: Routes = [
    {
        path:'',
        component:HomeComponent,
        title:"Home"
    },
    {
        path:'profile',
        component:ProfileComponent,
        title:"Profile"
    }
];
```

Output:



Welcome to SPA REST API Demo

VIVA QUESTIONS:

1.What is Angular ?

Ans: 1.What is angular ?
Angular is a popular open-source web application framework maintained by Google and a community of individual developers and corporations. It is written in TypeScript and is designed to make the development and testing of single-page applications (SPAs) easier and more efficient. Angular provides a comprehensive solution for building client-side applications with optimized practices and tools to develop, build, and test your code.

Key Features of Angular:

1.Component-Based Architecture
2.Data Binding
3.Dependency Injection (DI)
4.Directives
5.Routing
6.Forms
7.HTTP Client
8.Angular CLI
9.RxJS

2.What are the main advantages of angular ?

Ans:Angular, a platform and framework for building single-page client applications using HTML and TypeScript, offers several advantages that make it a popular choice among developers for creating dynamic and complex web applications. Here are some of the main advantages of using Angular:

1. Component-Based Architecture
2. Two-Way Data Binding
3. Comprehensive Framework
4. Modularity and Lazy Loading
5. Dependency Injection
6. TypeScript
7. Tooling
8. Testing
9. Community and Ecosystem
10. Cross-Platform Development

3.is angular two single-page application?

Ans:
Yes, Angular 2 and later versions (commonly referred to simply as "Angular") are designed to support the development of single-page applications (SPAs). Single-page applications are web applications that load a single HTML page and dynamically update that page as the user interacts with the app, rather than loading entire new pages from the server. This approach provides a more fluid, desktop-like user experience.

1.Component-Based Architecture
2.Client-Side Routing
3.Data Binding and Reactive Programming
4.Services and Dependency Injection

4.What is type script ?

Ans: TypeScript is an open-source programming language developed and maintained by Microsoft. It is a superset of JavaScript, which means that any valid JavaScript code is also valid TypeScript code. TypeScript adds optional static typing to JavaScript, among other features, to improve the language's scalability and developer experience for building large and complex applications.

Key Features of TypeScript:
1.Static Type Checking
2.Type Inference
3.Advanced Types
4.Compatibility with JavaScript
5.Tooling Support
6.Transpilation to JavaScript

5.What is dependency injection in angular ?

Ans:Dependency Injection (DI) is a core concept in Angular that allows a class to receive its dependencies from an external source rather than creating them itself. This design pattern is integral to the way Angular applications are structured and provides a way to increase their efficiency, modularity, and testability.

How Dependency Injection Works in Angular:
Providers: In Angular, dependencies are services or objects that a class needs to perform its function. These dependencies are defined as providers in Angular's DI system. A provider tells the DI system how to obtain or create a dependency.
Injectors: The DI system has injectors that are responsible for instantiating dependencies and injecting them into classes. Angular creates and manages an injector for you automatically when the application starts.
Injection Tokens: These are used to register and look up dependencies in the injector. Typically, the token is the class type of the dependency itself.
Decorators: Angular utilizes decorators to mark a class as a dependency and to inject dependencies into components, services, or other classes. The most common decorator for injection is @Injectable(), which marks a class as available to an injector for instantiation.