Name of Student: Pushkar Sane	
Roll Number: 45	Lab Assignment Number: 11
Title of Lab Assignment: Create an application to demonstrate SPA.	
DOP: 24-10-2023	DOS: 24-10-2023

CO Mapped: CO6	PO Mapped: PO3, PO5, PSO1, PSO2	Faculty Signature:	Marks:

Practical No. 11

Aim: Create an application to demonstrate SPA.

Theory:

1. **Single page application (SPA):**

   SPA is a web application that fits on a single page. All your code (JavaScript, HTML, and CSS) is recovered with a single page stack. Furthermore, route between pages performed without invigorating the entire page.

2. **Why is Angular called a single-page application?**

   AngularJS is a full featured SPA framework, with the help of which a single page application is created. In the SPA, the whole page is not reloaded every time, only every time the view will be changed.

   So when you load the application for the first time, not all the pages from the server will be rendered... It's only index.html that loads when you load the application. Since only a single page is loaded it is called SPA.

   The name "single-page application" comes from the fact that, in an SPA, the initial HTML page is loaded once, and after that, only the data is fetched and updated dynamically, without reloading the entire page. This approach provides a more fluid and responsive user experience, similar to that of a desktop application.

3. **Even if the URL changes in an angular site will it be called SPA?**

   In Angular applications, the URL can change, and the application is still considered a single- page application (SPA). The key characteristic of an SPA is that, even though the URL changes, the application dynamically updates and navigates within a single HTML page, without performing a full page reload.

   Angular's routing system allows you to change the URL, and when a new route is activated, it loads the appropriate component and updates the content on the page without requesting an entirely new HTML page from the server. This gives users the experience of navigating through different views or pages within the application, even though it's technically a single HTML page.
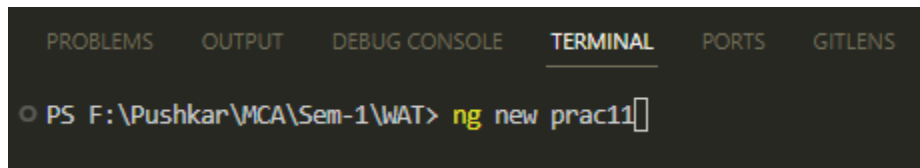
So, when you navigate to different routes in an Angular application, it's still considered a single-page application because the core HTML page remains the same, and the content changes dynamically, providing a seamless and responsive user experience.

The URL changes, because the router sets a virtual URL on your defined routes. This is just to fool the browser and make the go back and forth functionality work.
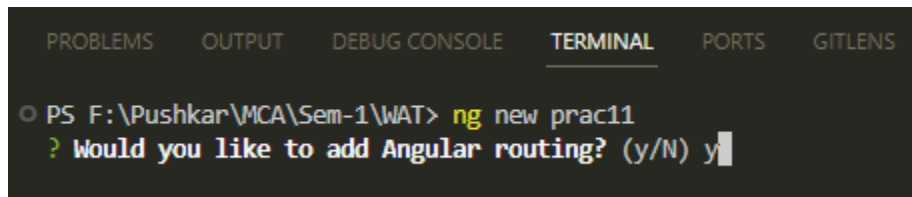
1. **Create an application to demonstrate SPA.**

   Steps to execute the practical:

   1) Create a folder "practical 11"

   2) Open the terminal (please use cmd, avoid using powershell) and navigate to the "practical 11" folder by using the cd command

   3) Create a new angular project by running the following command
      ng new prac11

   4) When prompted for Angular Routing, Enter "y"

5) When prompted for CSS, use arrow keys to select "CSS" (it is the first option and is selected by default) and then press Enter

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS

PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac11
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS   [ https://sass-lang.com/documentation/syntax#scss            ]
  Sass   [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less   [ http://lesscss.org                                         ]
```
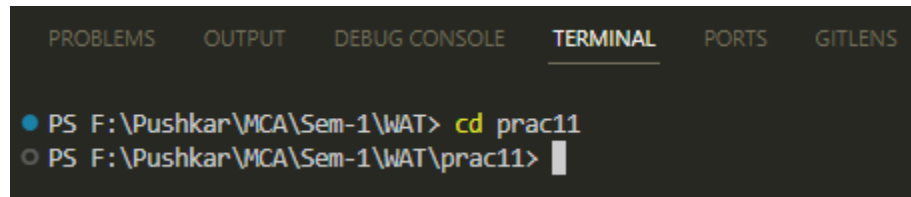
6) The new Angular project is created

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS

PS F:\Pushkar\MCA\Sem-1\WAT> ng new prac11
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE prac11/angular.json (2700 bytes)
CREATE prac11/package.json (1037 bytes)
CREATE prac11/README.md (1060 bytes)
CREATE prac11/tsconfig.json (901 bytes)
CREATE prac11/.editorconfig (274 bytes)
CREATE prac11/.gitignore (548 bytes)
CREATE prac11/tsconfig.app.json (263 bytes)
CREATE prac11/tsconfig.spec.json (273 bytes)
CREATE prac11/.vscode/extensions.json (130 bytes)
CREATE prac11/.vscode/launch.json (470 bytes)
CREATE prac11/.vscode/tasks.json (938 bytes)
CREATE prac11/src/main.ts (214 bytes)
CREATE prac11/src/favicon.ico (948 bytes)
CREATE prac11/src/index.html (292 bytes)
CREATE prac11/src/styles.css (80 bytes)
CREATE prac11/src/app/app-routing.module.ts (245 bytes)
CREATE prac11/src/app/app.module.ts (393 bytes)
CREATE prac11/src/app/app.component.html (22709 bytes)
CREATE prac11/src/app/app.component.spec.ts (991 bytes)
CREATE prac11/src/app/app.component.ts (210 bytes)
CREATE prac11/src/app/app.component.css (0 bytes)
CREATE prac11/src/assets/.gitkeep (0 bytes)
√ Packages installed successfully.
    Directory is already under version control. Skipping initialization of git.
PS F:\Pushkar\MCA\Sem-1\WAT> █
```

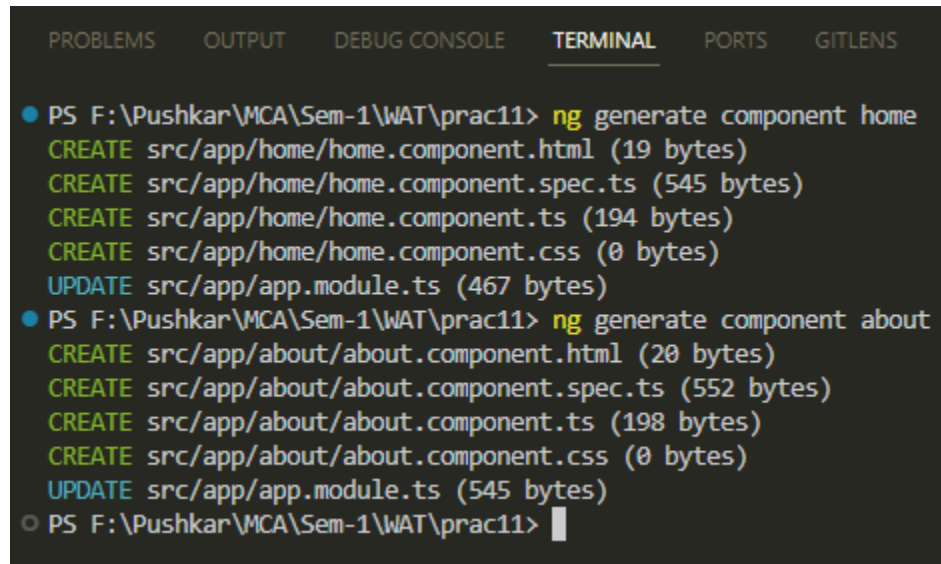7) Navigate to the prac11 folder by using the cd command

cd prac11

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS

PS F:\Pushkar\MCA\Sem-1\WAT> cd prac11
PS F:\Pushkar\MCA\Sem-1\WAT\prac11>
```

8) Create a new component "home" and "about" by running the following commands.
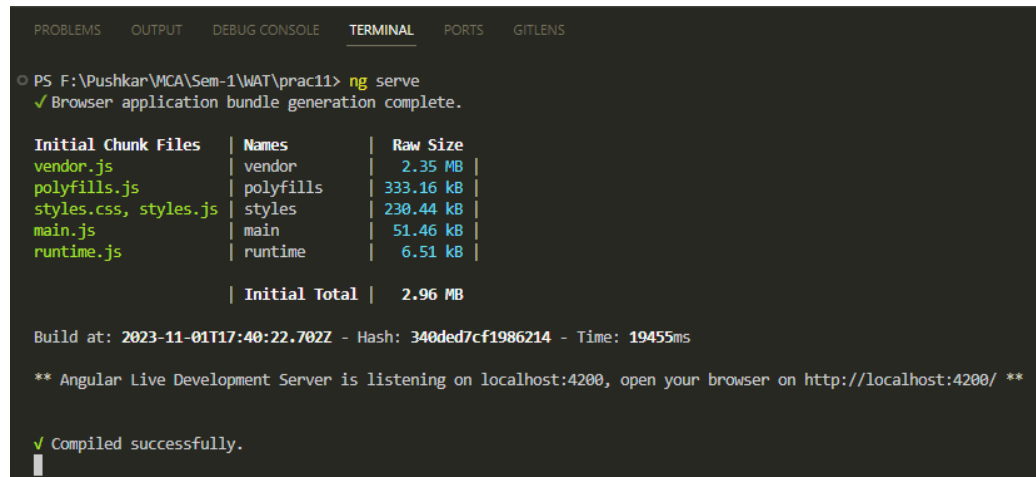
ng generate component home

ng generate component about

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    GITLENS

PS F:\Pushkar\MCA\Sem-1\WAT\prac11> ng generate component home
CREATE src/app/home/home.component.html (19 bytes)
CREATE src/app/home/home.component.spec.ts (545 bytes)
CREATE src/app/home/home.component.ts (194 bytes)
CREATE src/app/home/home.component.css (0 bytes)
UPDATE src/app/app.module.ts (467 bytes)
PS F:\Pushkar\MCA\Sem-1\WAT\prac11> ng generate component about
CREATE src/app/about/about.component.html (20 bytes)
CREATE src/app/about/about.component.spec.ts (552 bytes)
CREATE src/app/about/about.component.ts (198 bytes)
CREATE src/app/about/about.component.css (0 bytes)
UPDATE src/app/app.module.ts (545 bytes)
PS F:\Pushkar\MCA\Sem-1\WAT\prac11>
```

9) Run the following command to serve the angular project on a server

ng serve

10) The server is running and will automatically re-compile the project when we make any changes

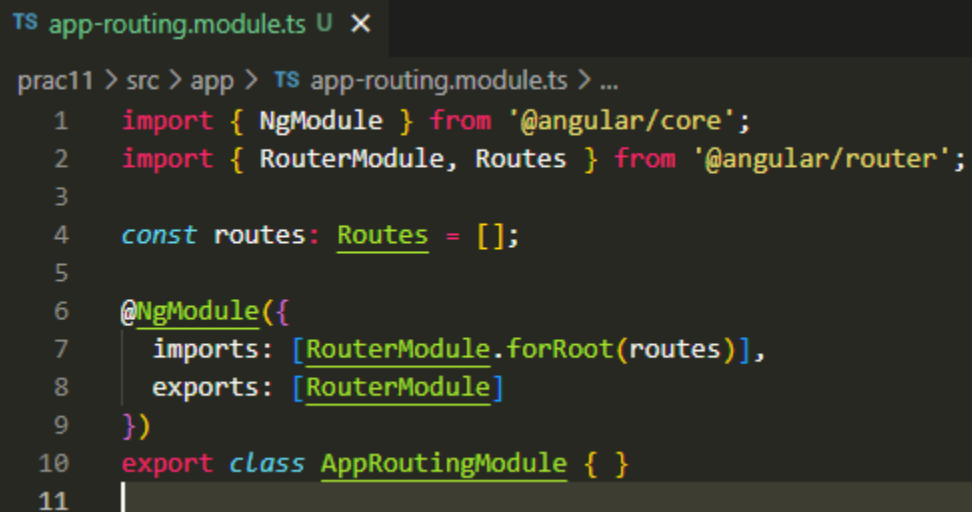11) Open any browser and go to the following link to see the output (this is the default output whenever any new Angular project is made) http://localhost:4200/
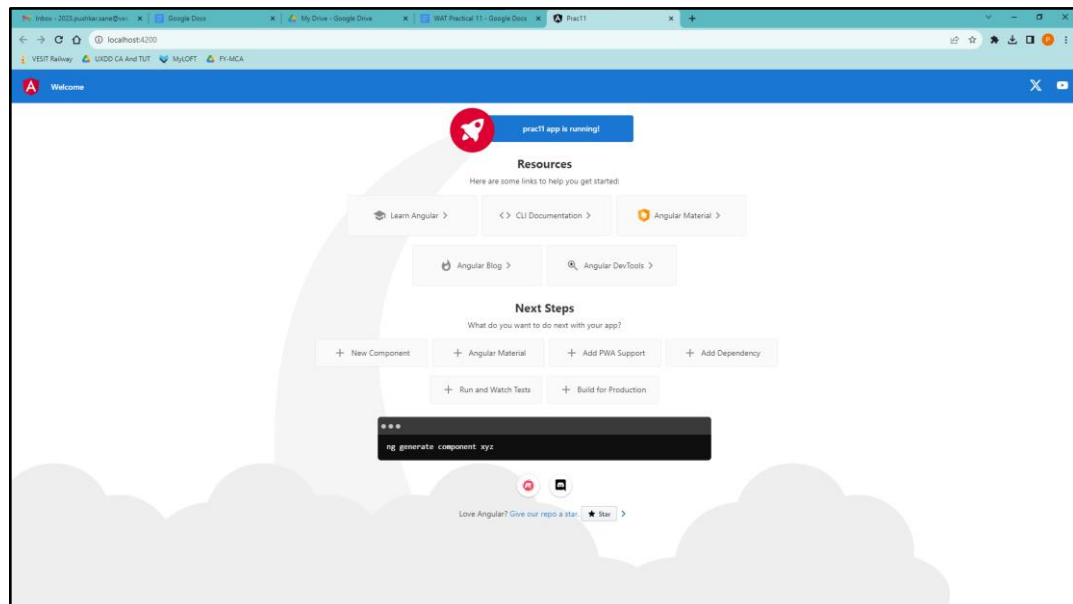
12) Add the "home" and "about" component to the **app.component.html** file

Delete all the contents of app.component.html file (located in the prac11/src/app/ folder) Add the following code to the file and save it:

```
<nav>
  <a routerLink="/">Home Link</a>
  <br>
  <a routerLink="/about">About Link</a>
</nav>
<router-outlet></router-outlet>
```
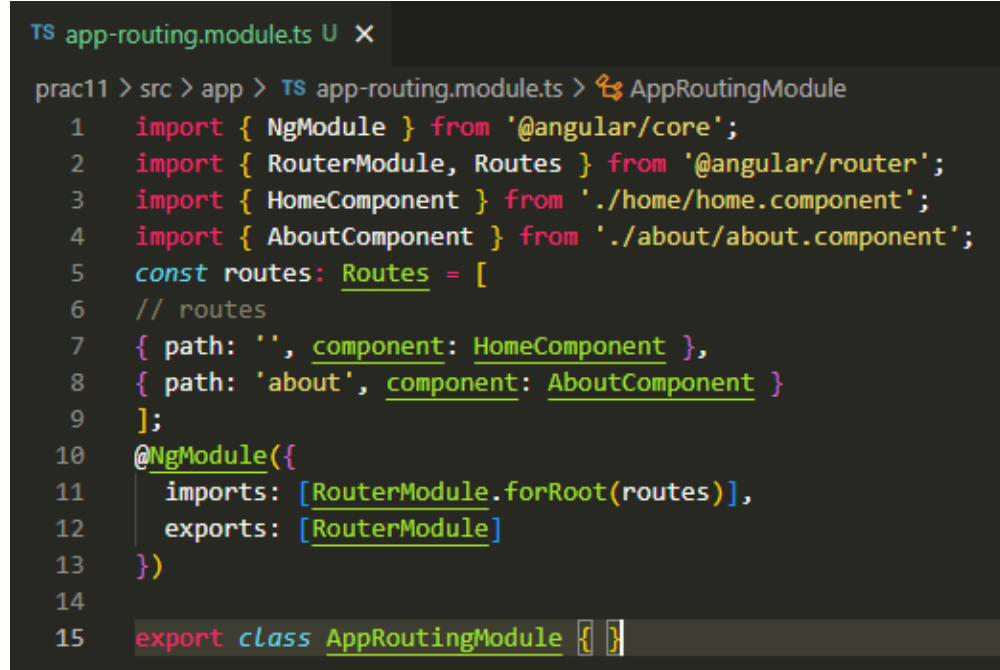
13) We will add routing to our Angular project

To do so, we have to modify the **app-routing.module.ts** in the "app" folder (prac11/src/app/)

```
TS app-routing.module.ts U ✕

prac11 > src > app > TS app-routing.module.ts > ...
    1    import { NgModule } from '@angular/core';
    2    import { RouterModule, Routes } from '@angular/router';
    3
    4    const routes: Routes = [];
    5
    6    @NgModule({
    7      imports: [RouterModule.forRoot(routes)],
    8      exports: [RouterModule]
    9    })
   10    export class AppRoutingModule { }
   11    |
```

```ts
TS app-routing.module.ts U  ✕

prac11 > src > app > TS app-routing.module.ts > ⚡ AppRoutingModule
  1    import { NgModule } from '@angular/core';
  2    import { RouterModule, Routes } from '@angular/router';
  3    import { HomeComponent } from './home/home.component';
  4    import { AboutComponent } from './about/about.component';
  5    const routes: Routes = [
  6    // routes
  7    { path: '', component: HomeComponent },
  8    { path: 'about', component: AboutComponent }
  9    ];
 10    @NgModule({
 11      imports: [RouterModule.forRoot(routes)],
 12      exports: [RouterModule]
 13    })
 14
 15    export class AppRoutingModule { }
```

After making the changes, the **app-routing.module.ts** file should look like this

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

import { RouterModule, Routes } from '@angular/router';

import { HomeComponent } from './home/home.component';

import { AboutComponent } from './about/about.component';

const routes: Routes = [

// routes

{ path: '', component: HomeComponent },

{ path: 'about', component: AboutComponent }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

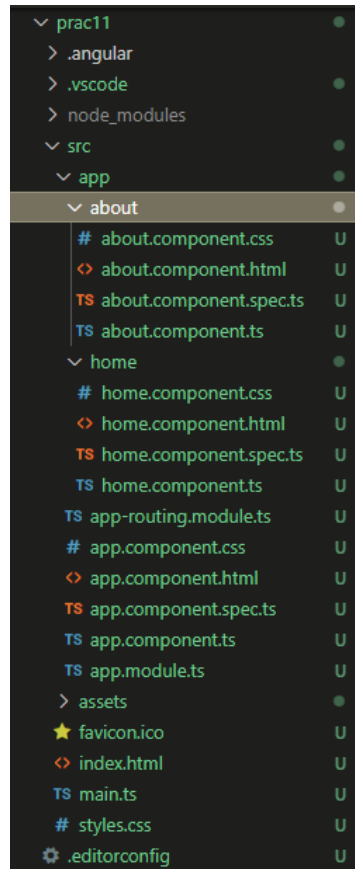14) We will remove the default code that's generated and add our practical code.
    We will make changes to the following 2 files in the "home" and "about" folders
    (prac11/src/app/home/

and

prac11/src/app/about/)

home.component.html

about.component.html

```
∨ prac11                          ●
  > .angular
  > .vscode                       ●
  > node_modules
  ∨ src                           ●
    ∨ app                         ●
      ∨ about                     ●
        #  about.component.css    U
        <> about.component.html   U
        TS about.component.spec.ts U
        TS about.component.ts     U
      ∨ home                      ●
        #  home.component.css     U
        <> home.component.html    U
        TS home.component.spec.ts U
        TS home.component.ts      U
      TS app-routing.module.ts    U
      #  app.component.css        U
      <> app.component.html       U
      TS app.component.spec.ts    U
      TS app.component.ts         U
      TS app.module.ts            U
    > assets                      ●
    ★ favicon.ico                 U
    <> index.html                 U
    TS main.ts                    U
    #  styles.css                 U
  ⚙ .editorconfig                 U
```

15) Open the **home.component.html** file and delete all the code

16) Write the following code in **home.component.html**

    <p>home page</p>

17) Save the **home.component.html** file.

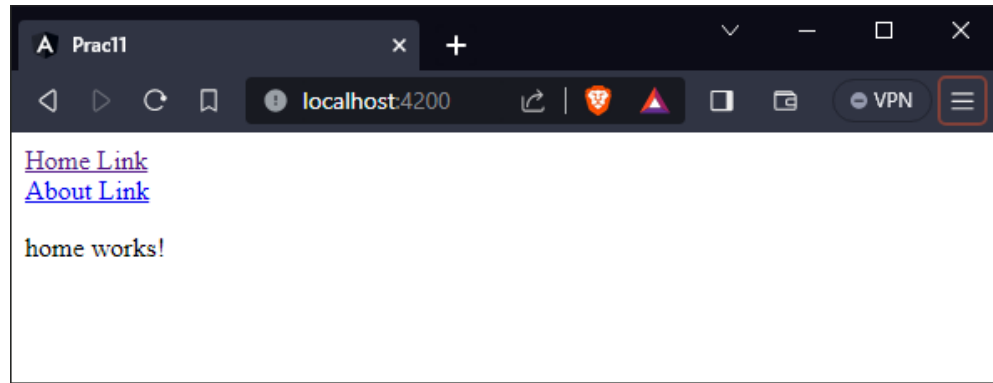18) Open the **about.component.html** file and delete all the code

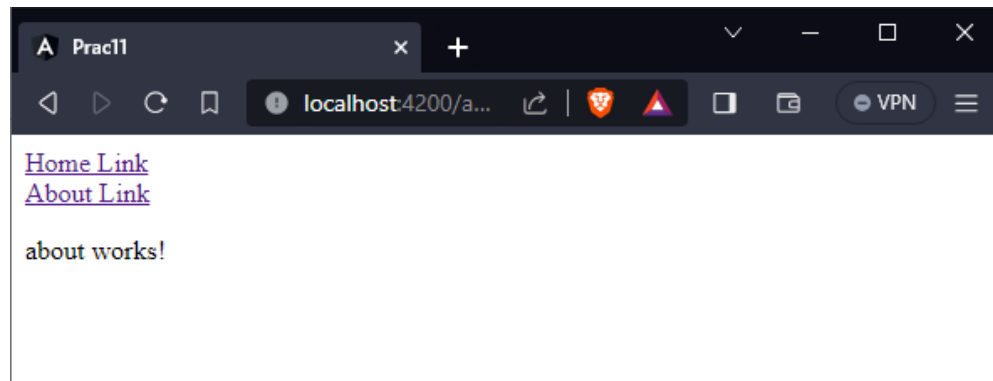19) Write the following code in about.component.html

    <p>about page</p>

20) Save the **about.component.html** file

21) Go to the browser to see the output at http://localhost:4200/
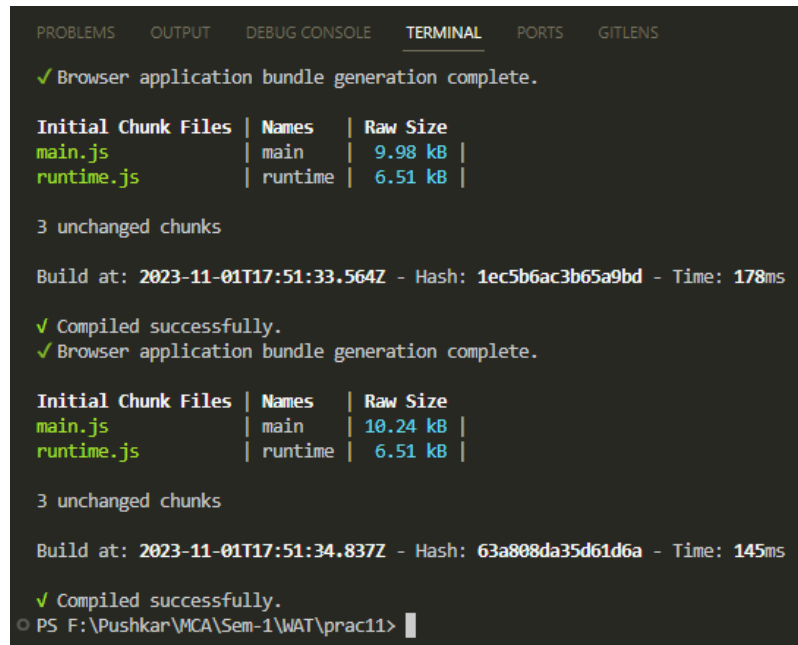
Output for the Home Route

Output for the About Route

22) Go to the terminal and stop the server by pressing Ctrl + C

When we see the prompt as "....\prac11>" then it means that the server has stopped

Code:

app.component.html

```
<nav>
  <a routerLink="/">Home Link</a>
  <br>
  <a routerLink="/about">About Link</a>
</nav>
<router-outlet></router-outlet>
```

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';
const routes: Routes = [
// routes
{ path: '', component: HomeComponent },
{ path: 'about', component: AboutComponent }
];
@NgModule({
```

   imports: [RouterModule.forRoot(routes)],

   exports: [RouterModule]

})

export class AppRoutingModule { }

home.component.html
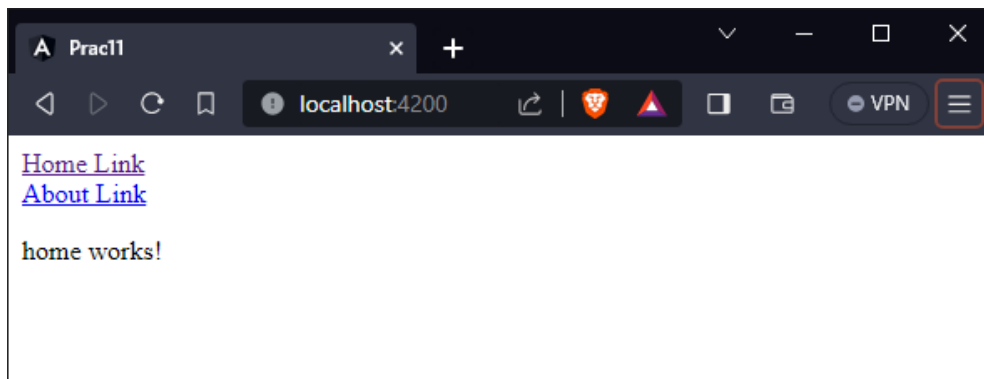
<p>home works!</p>

about.component.html

<p>about works!</p>

Output:

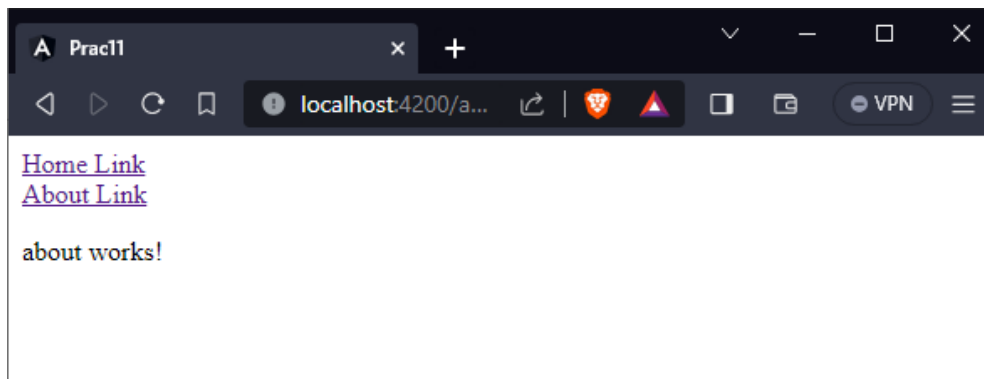**Home route**

About route

Conclusion: We learnt about SPA in Angular js.