Implement routing

Create application: ng new routing

Move to the folder: cd routing

Open the folder in VS Code: Code .

Run application: ng serve

Let’s create Some Components

ng g c home

ng g c about

ng g c contact

include bootstrap: npm install bootstrap

edit the styles.css file

/\* You can add global styles to this file, and also import other style files \*/

@import '../node\_modules/bootstrap/dist/css/bootstrap.min.css';

Add contents in About, Contact and Home HTML files.

Let’s Create Routes by editing app-routing.module.ts

const routes: Routes = [

  {path:'home',component:HomeComponent},

  {path:'about',component:AboutComponent},

  {path:'contact',component:ContactComponent}

];

Let’s Create Links in app.component.html file

<div class="container">

  <h1>Welcome to Simplilearn</h1>

  <ul class="nav">

    <li class="nav-item">

      <a routerLink="home" class="nav-link">Home</a>

    </li>

    <li class="nav-item">

      <a routerLink="about" class="nav-link">About Us</a>

    </li>

    <li class="nav-item">

      <a routerLink="contact" class="nav-link">Contact Us</a>

    </li>

  </ul>

  <router-outlet></router-outlet>

</div>

Save all files and check.

If you want to open a component by default then you can set the default path by using redirect.

 {path:'',redirectTo:'home',pathMatch:'full'},

Now if you try to check localhost:4200/ then also it will redirect you to localhost:4200/home

For creating child roots you can use children elements

Create 2 components Like Childa and childb

Update the about path as shown below.

{path:'about',component:AboutComponent,children:[

    {path:'',redirectTo:'childa',pathMatch:'full'},

    {path:'childa',component:ChildaComponent},

    {path:'childb',component:ChildbComponent},

  ]},

You have to add the links in About components and router-outlet too as mentioned below.

<div>

    <h1>About Us</h1>

    <nav class="nav nav-tabs">

        <li class="nav-item">

            <a class="nav-link" routerLink="childa">Overview</a>

        </li>

        <li class="nav-item">

            <a class="nav-link" routerLink="childb">Revenue Model</a>

        </li>

    </nav>

    <div class="row" style="border:2px solid red">

        <router-outlet></router-outlet>

    </div>

</div>

Let’s Say, if you want user to redirect on some component Like 404 if user trying to access any unavailable page.

Create component called pagenotfound:

Update the path as below

{path:'\*\*',component:PagenotfoundComponent}

Please make sure, this path must be at last in your routing file.

Create Dynamic Path where we can pass the data from URL

Create comp: ng g c details

Update path

{path:'details/:id',component:DetailsComponent},

Update app.component.html file

<li class="nav-item">

      <a routerLink="details/6" class="nav-link">Details Page</a>

    </li>

Update details.comp.ts file get the data from url

 id:number=0;

  constructor(private route:ActivatedRoute){}

  ngOnInit(){

      this.id=Number(this.route.snapshot.paramMap.get('id'));

  }

Update details html file to display the id

<h1>Details Page</h1>

<h3>Welcome, your Id is: {{id}}</h3>

Save a;ll and check the output.