**Day 4 Lab Assignments**

**Complete on the previous lab, and do the following assignments:**

1. Add a custom service ‘Products Service’:
   1. In the service add these functions (Depending on the classes you created before):
      1. getProductsByCatID(catID): Products []
      2. getProductByID(prodID): Product
   2. In Shopping Cart Component, and ProductDetails Component use the previous service instead of using the classes directly.
      1. What’s the meaning of ProvidedIn: ‘root’, in the service @injectable function?
         1. Explain the following statement: *“A service becomes singleton when we apply providedIn property of @Injectable decorator or Declare service in the providers’ array of @ngModule”*
         2. Can you instead add it in the Providers of specific Module or component? How?(Bouns)
            1. You can check the documentation here: <https://angular.io/guide/providers#providedin-and-ngmodules>
      2. Services in Angular are using Dependency Injection; you’ll need to inject the service in your component constructor, how?
2. Add routing to your Project:
   1. add the following links and components:
      1. Home (static page with any data)
      2. About us (static page with any data)
      3. Contact us (static page with any data)
      4. Products (that opens Shopping cart component).
      5. Default route, when user writes no route, redirects to /home route
      6. A Component to handle any wrong routes that when user writes a route that isn’t exist (Wildcard route).
   2. Remember to:
      1. Add your routes to routes variable in app-routing.module.
         1. Is the order of the routes in the configuration matters? Why?
         2. The router uses **first-match wins strategy** when matching routes, explain that?
      2. Add the <router-outlet> in the component which you’ll display the components you navigate to it.
      3. Change the links and add routerLink attribute.
      4. In all links add RouterLinkActive
         1. Why??
3. Add a route ‘products/:id’ that takes and id and display the given product details:
   1. Add the route in the router module routes.
   2. In your Shopping Cart component, add a button ‘details’ in front of each product, that navigates to the route: ‘product/:id’.
      1. You may need to use Router service, how?
         1. navigate() function in router service, return a promise, try it to print any message in the console after the navigation is completed.
      2. Can you do it by a link instead?
   3. Create ProductDetails Component, that receives an id and display the given product details.
   4. In the ProductDetails Component, get the sent ID, and display the given product details.
      1. You’ll need to use ActivatedRoute service, how?
   5. Add (Go back) button, that uses browser’s back to go back to the previous page .

**Bonus:**

* 1. Add new 2 links in nave bar (login, logout) , and handle the required login, logout components that use the login service to simulate the login and logout logic.
  2. Create a router guard (CanActivate guard), and use the UserAuthServie on it to check:
     1. If the iser is logged in allow him to open admin components (Add, delete and edit product).
     2. If not logged in don’t allow him to open admin components and redirect him to the login page.
  3. You may refer to the following **route guards** tutorial: https://[angular.io/guide/router#milestone-5-route-guards](https://angular.io/guide/router)
  4. Use BehaviorSubject, to show the login status in the navbar (to display: ‘Welcome + user name’ and Logout link when user is logged in, and login link only when is logged out **[Big Bonus]**.
  5. You can also refer to the following tutorial for login and registration using angular: <https://jasonwatmore.com/post/2018/10/29/angular-7-user-registration-and-login-example-tutorial>

1. **Try to display the product details in Angular material Dialog instead of new Component.**

**Enrich your knowledge [Super Bonus]:**

1. Search for the following topics in routing, and explain using a demo:
   1. Route guards: [https://angular.io/guide/router#milestone-5-route-guards](https://angular.io/guide/router)
   2. Asynchronous routing (Lazy Loading route): [https://angular.io/guide/router#milestone-6-asynchronous-routing](https://angular.io/guide/router)
   3. Child Routes: [https://angular.io/guide/router#a-crisis-center-with-child-routes](https://angular.io/guide/router)
   4. Make a route that takes some data through Breadcrumbs: <http://brianflove.com/2016/10/23/angular2-breadcrumb-using-router/> <https://medium.com/@bo.vandersteene/angular-5-breadcrumb-c225fd9df5cf>
   5. What are router events? How to handle them? Explain with Demo?   
      <https://toddmotto.com/dynamic-page-titles-angular-2-router-events>   
      <https://angular.io/api/router/Event>
   6. Displaying multiple routes in named outlets (Having more than outlet): [https://angular.io/guide/router#displaying-multiple-routes-in-named-outlets](https://angular.io/guide/router)
2. What are Observables in Angular? Explain using a demo?
   1. You can refer to: <https://angular.io/guide/observables>
3. What is Angular RxJS library? Explain using a demo?
   1. You can refer to: <https://angular.io/guide/rx-library>
4. What are: Subject, BehaviorSubject and ReplaySubject in Angular? Explain using a demo?
   1. You can refer to: <https://stackoverflow.com/questions/34376854/delegation-eventemitter-or-observable-in-angular/35568924#35568924>
5. What is Angular localization? Use it to make an Arabic version of your Angular project.
   1. You can refer to: <https://angular.io/guide/i18n>
6. What is Angular Material? Explain using a demo?
   1. You can refer to: <https://material.angular.io>
7. How to use unit testing in Angular? Explain using a demo?
   1. You can refer to: <https://blog.logrocket.com/angular-unit-testing-tutorial-examples>
8. What is Angular universal? Explain using a demo?
   1. You can refer to: <https://angular.io/guide/universal>