PRSng – User

1. In **app.module.ts**
   * Add import statements
2. In **user.service.ts**
   * Import HttpClient
   * Import Observable
   * Make constructor
3. In **app.component.ts**
   * Import UserService
   * Add constructor
4. Create class (file) **user.class.ts** in VSC
   * Create and export User class
   * Add instance variables
   * Create default constructor (don’t pass anything in it), add variables that cannot be null
5. Create class (file) in new folder “util”, **JsonResponse.class.ts**
   * Create instance variables for code, message, data, and error
     + In user.service.ts, import User and JsonResponse classes
   * Create getter, pass id
6. In GitBash, inside the user folder, generate: (ng g c user/comp-name)
   * user-list component
   * user-detail component
   * user-create component
   * user-edit component
7. In **app.component.html**
   * Add routerLink for users/list (in <nav> tag)
   * In a <div> tag, add <router-outlet>
8. In **app-routing.module.ts**
   * Add to router array
   * Import: UserListComponent, UserDetailComponent, UserCreateComponent, and UserEditComponent
   * Add components to array
9. In user-list inside **user-list.component.ts**
   * Import UserService and User
   * Create an array of users of type User
   * Inject UserService
10. In user-list inside **user-list.component.html**
    * Add tables for all variables in thead
    * In tbody, do <tr \*ngFor thing> and do {{usr.ID}}
11. In user-list folder inside **user-list.component.html**
    * Add new <div> tag at the top with a routerLink
    * Add routerLinks below for Detail and Edit
12. In user-detail folder inside **user-detail.component.ts**
    * Import ActivatedRoute, UserService, and User class
    * Add ActivatedRoute to constructor
    * In UserDetailComponent, add an instance of user type User
    * In ngOnInit get :id from route
    * Get user from UserService (inject UserService into constructor
13. In user-detail folder inside **user-detail.component.html**
    * Display the page title (User Detail)
    * Add a table
    * Add \*ngIf = “user” to <table> tag (this gets rid of errors)
14. In **app.module**
    * Import FormsModule
    * Add FormsModule to import array
15. In **user-edit.component.ts**
    * Import ActivatedRoute and Router
    * Import UserService and User class
    * Inject UserService, ActivatedRouter, and Router into constructor
    * Create an instance of user of type User
    * Create save function
    * Do something in ngOnInit
16. In **user.service.ts**
    * add the add function
    * add change (update) and remove (delete)
17. in **user-edit.component.html**
    * add some styling tables!
18. In **user-create.component.html**
    * Copy and paste html table styling from **user-edit.component.html**
19. In user-create folder inside **user-create.component.ts**
    * Import Router, UserService, and User class
    * Create an instance of an empty user
    * Inject UserService and Router into constructor
    * Add the add function (copy from save in **user-edit.component.ts**)
20. In user-detail folder inside **user-detail.component.ts**
    * Add a remove (delete) function
    * Import Router
    * Inject Router into constructor
21. In **user-detail.component.html**
    * Add a delete button
22. In GitBash do >ng g c menu/menu to create a menu component inside a menu folder
23. Create **menu.class.ts** in the main menu folder (not menu component folder)
    * Export Menu class
    * Add instance variables
    * Add constructor
24. In GitBash add home and about components (ng g c name)
25. In **app-routing.module**
    * Import MenuComponent, HomeComponent, and AboutComponent
    * Add routes for home and about
26. In menu/menu open **menu.component.ts**
    * Import Menu
    * Add a property in component for all menu items we’ll create
      + Array menu of type Menu
27. In **menu.component.html**
    * You can copy a cool Bootstrap menu into here or…
    * Create an unordered list to display menu items (routerLink)
28. In **app.component.html**
    * Get rid of everything (including) <nav> tag with old menu link
    * Replace it with <app-menu></app-menu>
29. In **menu.component.css**
    * Add some pretty styling!