Angular project to read C#/MVC/WebAPI/SQL Server backend

Follow the steps to build the Angular app.

1. Create the project named *angular-to-webapi* and include routing
2. In app.module.ts:
    a. Import FormsModule and add to appropriate decorator key
    b. Import HttpClientModule and add to appropriate decorator key
3. Create the following folders under the *app* folder
    a. Create a folder named *model* under *app* folder
    b. Create a folder named *service* under *app* folder
    c. Create a folder named *user* under *app* folder
4. Generate a *class* file named *user.ts* in the *user* folder
5. Create the class *User* in the *user.ts* file with the following properties and methods (make sure you export it)
    a. ID: number;
    b. UserName: string;
    c. Password: string;
    d. FirstName: string;
    e. LastName: string;
    f. Phone: string;
    g. Email: string;
    h. IsReviewer: boolean;
    i. IsAdmin: boolean;
    j. Add a constructor to the *User* class that initializes all properties
6. Create the service *user.service.ts* in the *app/service* folder
7. Add an import and appropriate decorator key for *user.service.ts* to *app.module.ts*.
8. Make the following changes in *user.service.ts*:
    a. Add an import for *HttpClient* from *@angular/common/http*
    b. Add an import for *Observable* from *rxjs/Observable*
    c. Add an import for *User* from *app/model/user* (user relative path!)
    d. Inject *UserService* into the class as *UserSvc*
    e. Create a function in the class named *list()* that returns a type of *Observable<User[]>*
        i. In the body of the *list()* function, make a call to the *get(..)* function of the *UserSrv* passing the following url 'http://prs.doudsystems.com/Users/List'. Return the result of this function call as *Observable<User[]>*
    f. Create another function in the class named *get(id)* that takes an id as a parameter and returns a type of *Observable<User[]>*
        i. In the body of the *get(id)* function, make a call to the *get(..)* function of the *UserSrv* passing the following url 'http://prs.doudsystems.com/Users/Get/+id'. Return the result of this function call as *Observable<User>*
9. Generate the *user-list* and *user-detail* components in the *user* folder
10. Make the following changes to the *user-list.component.ts* file.

    a. Import the *UserService* service
    b. Import the *User* class
    c. Create a property called *users* of type *User[]*
    d. Inject the *UserService* into the component as *UserSvc*
    e. In the ngOnInit()
        i. Call the *list()* function of the user service and subscribe to the results and place the data returned in the *users* property
        ii. Display the returned data by calling console.log(this.users)

11. Make the changes to *user-list.component.html* to render the data
    a. Add one extra column called *Action* and put a link to a route to call the *user-detail.component* and pass the Id of the user id.
12. Make the following changes to *app-routing.module.ts*:
    a. Import *user-list.component* and *user-detail.component*
    b. Add the following to the *routes* array:
        i. On startup, navigate to the */users/list* route
        ii. On route */users/list*, load the *UserListComponent*
        iii. On route */users/detail*, accept a route parameter named *id*, load the *UserDetailComponent*
        iv. On any other route, load the *UserListComponent*
13. Make the following changes to the *user-detail.component.ts*
    a. Import *Router* and *ActivatedRoute* from @angular/router
    b. Import the *UserService* service and *User* class
    c. Create a property named *user* of type *User*
    d. Inject the *UserService* as *UserSvc*, *Router* as *router*, and *ActivatedRoute* as *route*
    e. In the ngOnInit()
        i. Create a local variable named *id* as a string
        ii. Get the router parameter *id* by calling: *this.route.params.subscribe(parm => id = params['id'])*
        iii. Use the variable *id* to make a call *get(id)* call to the UserService subscribing the result and storing it in the property *user*
14. With the data being retrieved by both component
    a. Modify the *user-list.component.html* to display the list of user records
        i. Include an *Detail* like to like to the /users/detail/:id via routerLink
    b. Modify the *user-detail.component.html* to display the specific user record based on the user clicking the appropriate *Detail* link on each user.