Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail

1. What is Router in component? How do you use it?
2. What is RouterOutlet?
3. What is <base> element in Angular? Where should we put it?
4. How can we link between different routes?
5. What are different ways to pass data with route? (Explain ActiveRoute)
6. Why we need to use Observable in ActiveRoute?
7. How to define child route in Angular?
8. How will you prevent user to go to certain url in Angular?
9. What is lazy loading modules in Angular?
10. What different build-in pipes have you used?

Coding Questions:

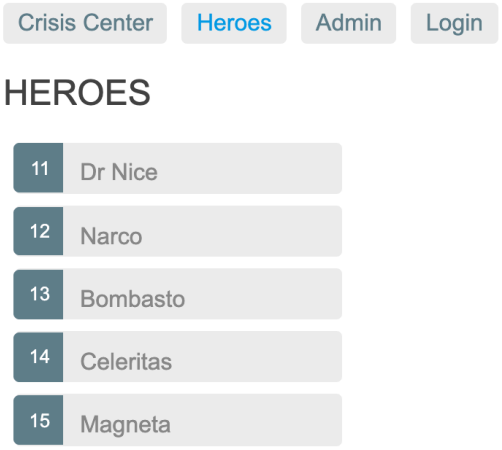
Write code in Java to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

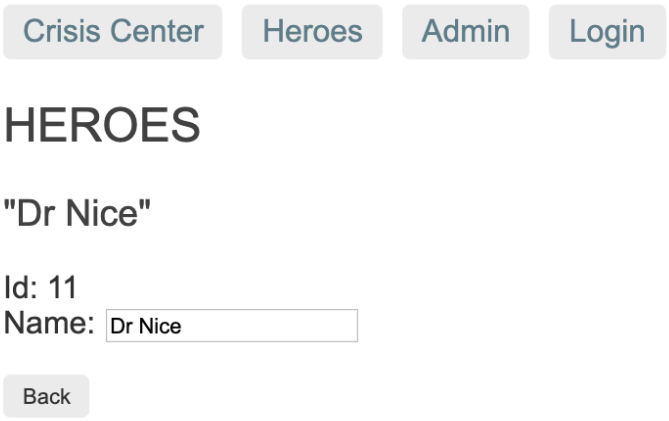
Submit screen shot in a doc or PDC for all assignment.

1. Implement an application that helps the *Hero Employment Agency* run its business. Heroes need work and the agency finds crises for them to solve.

The application has two main feature areas:
- A Crisis Center for maintaining the list of crises for assignment to heroes.
- A Heroes area for maintaining the list of heroes employed by the agency.

Requirement:
- Hero
    - All hero related component should be in a different module rather than the root
        - i.e. Create a hero.module.ts
    - Hero module should have at least two component: hero-list and hero-detail
        - whenever there is an item clicked, take it to hero detail page
        - on hero detail page, provide an input box to accept the name changes
            - This change should take effect immediately without any button click
        - pass select user by route (either route parameter or query parameter)
    - Create a hero.model.ts which represent the hero object which contain two fields:
        - Name
        - Id





- Crisis
    - Display all tasks in a list
    - Select a task and the application takes you to a crisis editing screen. The *Crisis Detail* appears in a child component on the same page, beneath the list.
    - Unlike *Hero Detail*, which updates as you type, *Crisis Detail* changes are temporary until you either save or discard them by pressing the "Save" or "Cancel" buttons. Both buttons navigate back to the *Crisis Center* and its list of crises
    - Do not click either button yet. Click the browser back button or the "Heroes" link then pops up a dialog box
    - You can say "OK" and lose your changes or click "Cancel" and continue editing
        - Behind this behavior is the router's CanDeactivate guard