

Angular

Capabilities covered

KO observation 1:

Ability to create an Angular application using Typescript & Angular CLI

KO observation 2:

Ability to use and create angular modules, components, controllers

KO observation 3:

Ability to use Routing (module, component), lazy loading of components

Problem Statement

- ➔ Design a Single Page Application (SPA) named as “**Kalinga Care**” which contains fixed header, fixed side navbar and content page by using Angular Routing concepts to achieve the desired SPA.

Assignment

→ Create an Angular App named as “Kalinga-Care” using Angular CLI in the Visual Studio Code IDE.

→ Create a fixed header comprising of various links like “Logo”, “Home” and “Care”.

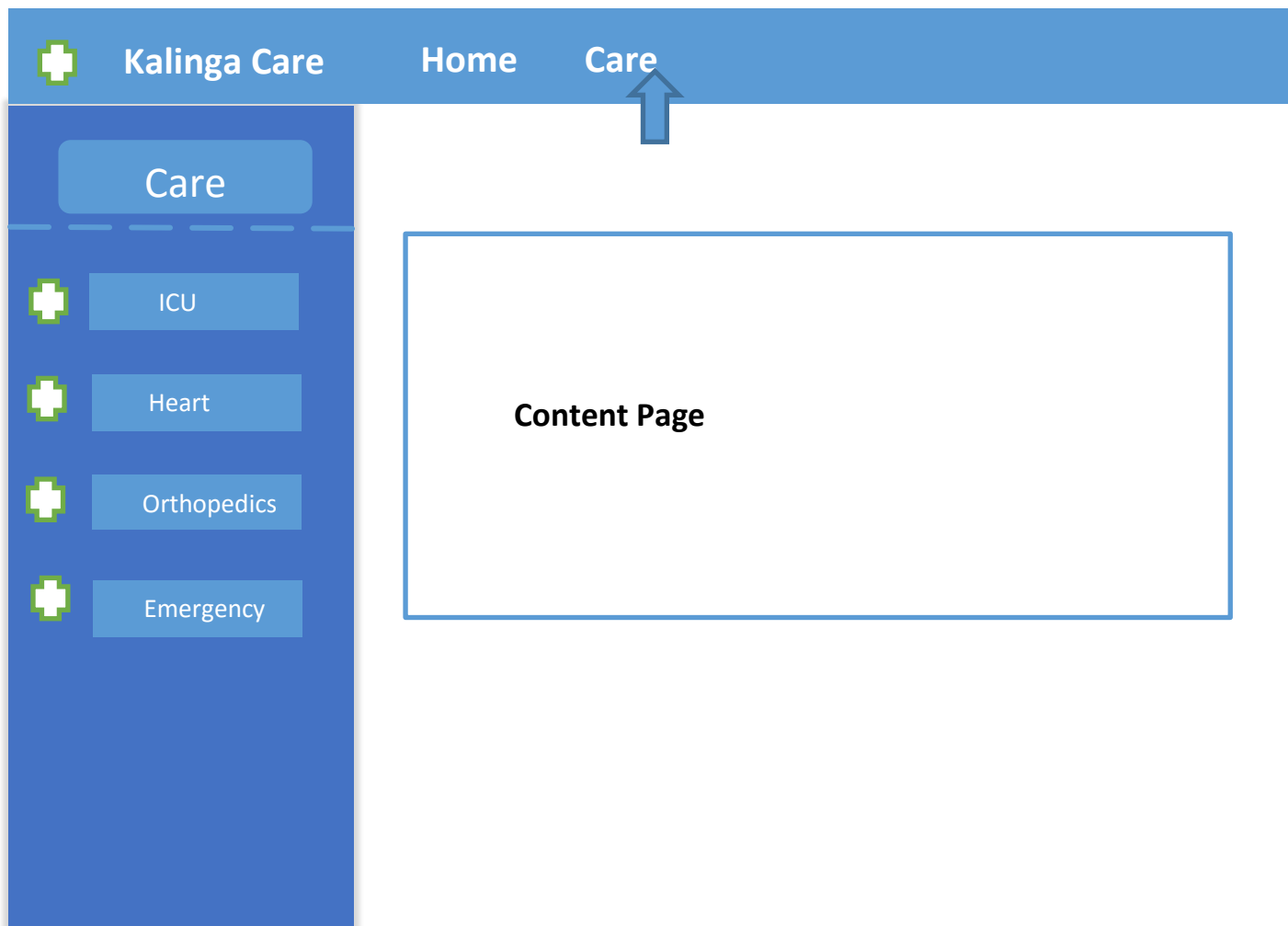
→ There should have 3 Parent Module named as **AppModule**, **HomeModule**, **CareModule**. Any one of the module should be called on clicking the router links present on the header. On clicking “Home” link

the **HomeModule** should load all the components associated with the module and like-wise other links should follow the same structure.

→ The default landing page should be the **HomeComponent** which resides in the **HomeModule** and router link “**Home**”. The landing page does not contain any sidebar, it is only Care link contains fixed sidebar.

→ On clicking “**Care**”, the **CareModule** should get loaded with different components. It should display following use cases:-

#Use Case 1:- On clicking “Care”, the left sidebar comes into the page which is fixed in nature that contains few links as shown below.



#Use Case 2:- The links on the left side should load their respective components html page which are child components of CareComponent. For eg:- On clicking the “*ICU*”, the ICU component page (which is the child component of CareComponent) should get rendered on the right side and like-wise other links.

Note:-

- i) You can use necessary lifecycle hooks and apply some Angular Material Design, Bootstrap to beautify the website to cover the Non Kos.
- ii) No need of database storage is required. Use necessary JSON files to read the data.
- iii) The above app gives a clear picture how to use routing, child routing and routing from parent to child components.

*****End of Document*****

