

Disclaimer

The information contained in this document is intended only for the person(s) or entity that has signed our NDA and may contain copyright, confidential and/or privileged material. Any review, retransmission, dissemination or other use of, or taking of any action in reliance upon, this information by persons or entities other than the intended recipient is prohibited. Any violation of this would be considered as a breach of the NDA and result in immediate termination of this engagement and further action if required.

Scope of the webapp

As part of this test we are developing a test web application based on Angular version 8 or above. We want to test some core principles of Angular and your understanding of these principles.

Test Requirements

The technology to be used are the following

∙ Angular (8 or higher at the time of this writing), and Typescript

∙ Local environment setup with Angular 8 or above

∙ Bootstrap or Material

∙ HTML5 and SASS

∙ https://github.com/creativetimofficial/material-dashboard-angular2 (**Highly recommended to save you time**)

∙ Git (your local GIT repository)

Core Angular Principles

Emphasis of this test is on following areas of Angular development in decreasing order of priority. Your test output will be evaluated on existence of the following highlighted points

**1. Modular approach**

2. Implementing **Reusable Components** using **TemplateReference** in the test 3. Your understanding and implementation of **ViewChild / ViewChildren / ContentChild / ContentChildren** in the test

4. Understanding of state management using **ngRx**

5. Events-based communication using standard reactive patterns

Important Instructions for the Test

∙ You can use a starter dashboard (SB-Admin-BS4-Anuglar-5) or you can develop something from scratch – your preference.

∙ Please make sure you use GIT to make local commits

∙ No need to store data remotely (remote database). Hence no need for database development. ∙ No services or any backend integration required.

STRICTLY PRIVATE AND CONFIDENTIAL ©Healum Limited 2021. All Rights Reserved.



Test

Functional Implementation Requirements (to implement in the test) A **hypothetical** company called “Acme Company Ltd” wants to a build a note taking web app. They have come with the idea that the world needs different types of notes. There are found modules in this webapp. The modules are

∙ Text Notes Module

∙ Picture Notes Module

**Sample data model**

- Text Note input – { description }

- Picture note input – { title, description, picture }

Each of the above module has the following requirements

1. As a user I would like to create a note from different types of modules as mentioned above – text module, or picture module

2. As a user I would like to see various different types of notes after they are created, on the landing page of their respective modules (root screen of each module)

**3. Please use NgRx for state management of any one type of note in your code (or all if it’s easier) – to view detail of the created note using store, action and effects**

Rules for the test

∙ You **CAN NOT** use ng-if or show/hide to display different types of notes

∙ You **CAN NOT** use ‘note.type’ as a component parameter to display different types of notes

Future functional requirements from “Acme Company Ltd”

For “Acme Company Ltd” this is a growing product with complex future customer requirements. (not to be implemented but these requirements **should guide your software design decisions in your current implementation**)

- As a company we would like to add **15** different types of notes (and respective modules) with each note different number of form fields (5-10 fields) that are **NOT** same as each other. - Design your **reusable components** keeping in mind that new types of notes (and modules) as mentioned above will be added with each module having some additional functional requirement. - These new types will heavily rely on the existing reusable components (to be built part of the test) but will have unique functionality of their own that will be in addition to the **existing reusable component**.

- As a company we want to be able to view details of the different note types by clicking on a note - As a company we would like to browse different notes and their respective details whilst being able to create a new note on the home screen.

STRICTLY PRIVATE AND CONFIDENTIAL ©Healum Limited 2021. All Rights Reserved.



Guidance for the test

1. Implement the test keeping future requirements of the company “Acme Company Ltd” – The code and software design needs to be developed keeping future requirements of the company in mind.

2. **Code Design**- We want to focus on modular, reusable code-design for this test webapp that allows easier maintenance of code and is based on sound software design principles. 3. **Modular approach** - We want the application to be able to lazyload different feature based modules which will be based on user access rights and permissions.

4. **Reusable Components** - UI Components used within each note module should easy to reuse and modify for module specific requirements. How will you handle that?

5. **Functional Extensibility** – The above principles should allow the Acme Company to implement the future requirements with minimal code changes.

6. Understanding and usage of **TemplateReference** in conjunction with **ViewChild / ViewChildren / ContentChild / ContentChildren** to implement some of the **reusable UI components**

7. State management – State communication between different components needs to be managed using NgRx.

STRICTLY PRIVATE AND CONFIDENTIAL ©Healum Limited 2021. All Rights Reserved.