Instructions challenge WaarneemApp

Introduction

During this challenge you will be building a simple CRUD page using VueJS, Nuxt.js and the Vuex store to handle the dataset. You have **3 hours** to complete as much as you can. During the assessment we will mainly asses you on the following aspects:

**1. Quality of you code**

Here you can think of KISS, DRY, YAGNI, inheritance, readability, consistency, architectural structure (SOLID principle) and so on.

**2. Quantity of the code**

Here it is not about how much you code, but how much you can code with a high quality. Its about being productive and knowing what's important in coding and what's not.

**3. Decisions you make**

We also look at what decisions you have made and why. Ultimately this will determine your insight in the underlying understanding of the code and your ability to produce a viable product within a short amount of time.

You may use existing packages but realise they will also be judged by the same criteria. Make sure you put the code in a repository with short instructions on how to make it run locally. Please use bitbucket for this.

Dependencies

Install Nuxt.js and Vuex. You are free to use any css framework. You are free to use a vuejs component library, we use Buefy but if you have little experience with it, feel free to use another one. You will not use a database as you will save the data in the Vuex store. A database is not part of the project.

Assignment

The assignment is to create a CRUD a simple page to create and edit shifts for healthcare workers. This is a public page, no login functionality is required. It should be a SPA. The page consists of two parts.

1. Read component (overview)

2. Create/Edit component (sidebar)

**1. Overview**

Here you can see all the added vacancies. A vacancy can contain multiple shifts. A vacancy contains:

Title - required, max 100 characters

Description - Optional, max 500 characters

Dates: required, dd-mm-yyyy, minimal 1 entry, max 10. The dates contain the following data:

Starttime - required hh:mm

Endtime - required hh:mm, should be > starttime

Price - required, can be 0.

Type - required, must be one of array [Consultation, Telephone, Ambulance]

Filters - There should be a slider to filter the by price. The slider should be dynamic in that the range is the calculated min-max of the records.

**2. Create/Update/Delete drawer**

Use the same component for these actions. When saving. you should not allow for overlap of time in the same day when using the same type.

*a. Create*

When the user clicks on the ADD SHIFT button, the drawer should appear. There should be validation based on the requirements shown above.

The datapicker can select multiple dates. We suggest to use a package for this (eg vue-calendar).

For every date entry create a new card with the corresponding inputs. This card can be destroyed again.

*b. Edit*

By using the edit button in the shift card the drawer should appear and the records should be editeble.

Design/UX

Design Already Provided in Separate file named “**Design.png**”

You are free to make changes on the design if you feel this improves the overal quality of the application.

Questions

During the challenge you cannot ask us questions. You need to figure it out for yourself. Goodluck!