Mazaady Task Documentation

Project structure

* Assets
* Components
* Helpers
* Layouts
* Locales
* Mixins
* Pages
* Plugins
* Static
* Store

1 – Assets

This folder contains css style && images

* CSS

1. Variables => contains all css global variable such all colors
2. Mixins => refer to all global sass mixins an functions
3. Main => refer to main css file in the project and you can find it in global css in nuxt.config.js

Variable & Mixins => you can find these files below styleResources in nuxt.config

* Images

Contains more folders , global folder contains the global images in the project like shared icons etc, another folders every folder of these folders refer to a page

2 – Components

This folder contains almost of used components in the project

* Shared Folder => refer to the shared components like app navbar and footer
* Reuseable => refer to the reusable components in almost of pages such as AppButton or PrdouctCard in our application , and these components depends on props and some emits event to pass data or handle and share data from these components to the parents etc..
* Modules => this part in components contains more folders depends on pages for ex inside it I put categories folder contain all components I use it in categories page etc…

3 – Helpers

This folder for handle response after each API call we can use it to do somethings like that

4 – Layouts

This folder refer to the layouts we will use it in the project so it can be just the default layout or more than one depends on the project so I used the default layout here and wrapped in it Main App Navbar and Footer ,

* Wrapped all things inside Main => this context is a best practice for structuring the HTML document also providing better clarity for developers and organized and accessibility and SEO
* Nuxt => this for routes and switch between pages etc..

5 – Locales

This folder contains all languages folders in our project like en, ar etc..

6- Mixins

This folder contains shared methods we can use it global in our project I added one for example to remove hyphen from text so for ex- we have string text like “hello-i-am-mohamed-almearag” and want to remove – from the text e can use the method I added it there.

Example --

removeHyphenFromText(text) {

   let removeHyphen = text.replace(/-+/g, ' ')

   return removeHyphen

},

You will see this ( -+ ) => first thing ( - ) this what I want to remove and ( + ) that meaning remove one hyphen or more if found in the text

7 – Pages

This folder contains all pages in our project

8 – Plugins

This folder contains the npm packages I use it in mazaady task

* Nuxtjs/axios – for handler APIs call and status of response messages etc..
* Nuxtjs/i18n – for localization and switch between application multiple languages
* vue-multiple-select – used it in dropdowns and make search in it ec.. you can check [**this link**](https://vue-multiselect.js.org/#sub-getting-started) for documentation
* vee-validate – for validate form inputs before submit any form and send data to the backend check [**this link**](https://vee-validate.logaretm.com/v2/guide/localization.html#localization-api), in this file you will see these steps

1. import vue – to register components vee-validate global and use it in all pages
2. import main vee-validate components and extend from vee-validate rules and this extend to use it to handle required validation and the error messages
3. {\_field\_} => this filed object with \_ refer to the filed name in any form you can pass it with simply way to the provider container so this will allow you to handle the error messages with name for example input name is bio for the user so the error message will be like this “the bio is required” , another thing this filed not depend on the v-model value so if you use v-model for description and this key will send it to the backend and you want to make the error message with bio this will allow you to do that .

* Nuxtjs/toast – you can use it for any messages or handle error on it etc.. for example in mazaady task I used it after I click to add to favorite heart I display a notification message after that tell the user this product added successfully you will see these things on this file in trigger notify method

// trigger notification method

TriggerNotify(type, message) {

this.$toasted.show(message, {

type: type,

duration: 1000, // refer to the time when display to hide

position: 'top-right', // position for notification

})

},

1. Type – refer to the message type , success, warning , error , etc , you can use response type also
2. Message – refer to the message content , you can use localization

9 – Static

This folder contains logo , and sprite.svg file this sprite we can make icons on it with simple way just we wrap all icons inside svg tag and each icon with symbol and make unique id for every symbol and when we use any icon from it inside the application with simple way like this

Example for use this file

<svg

   <use xlink:href="~/static/sprite.svg#search"></use>

</svg>

Search => refer to the unique id we put it to the symbol

10 – Store

This for our statemangment

**Explanation for main form component**

This component contains 3 main things

1. Template for the HTML markup
2. Script for component name, methods , data, props, event emits, watch, etc
3. Scoped style just for this component

Let’s see the script details

emits: ['formSubmitted'],

this refer to the emits event in main form component and I use this when submit the from to send all user selected data to the TABLE component to display it so the user can see his chosen category

props: {

    categories: {

      type: Array,

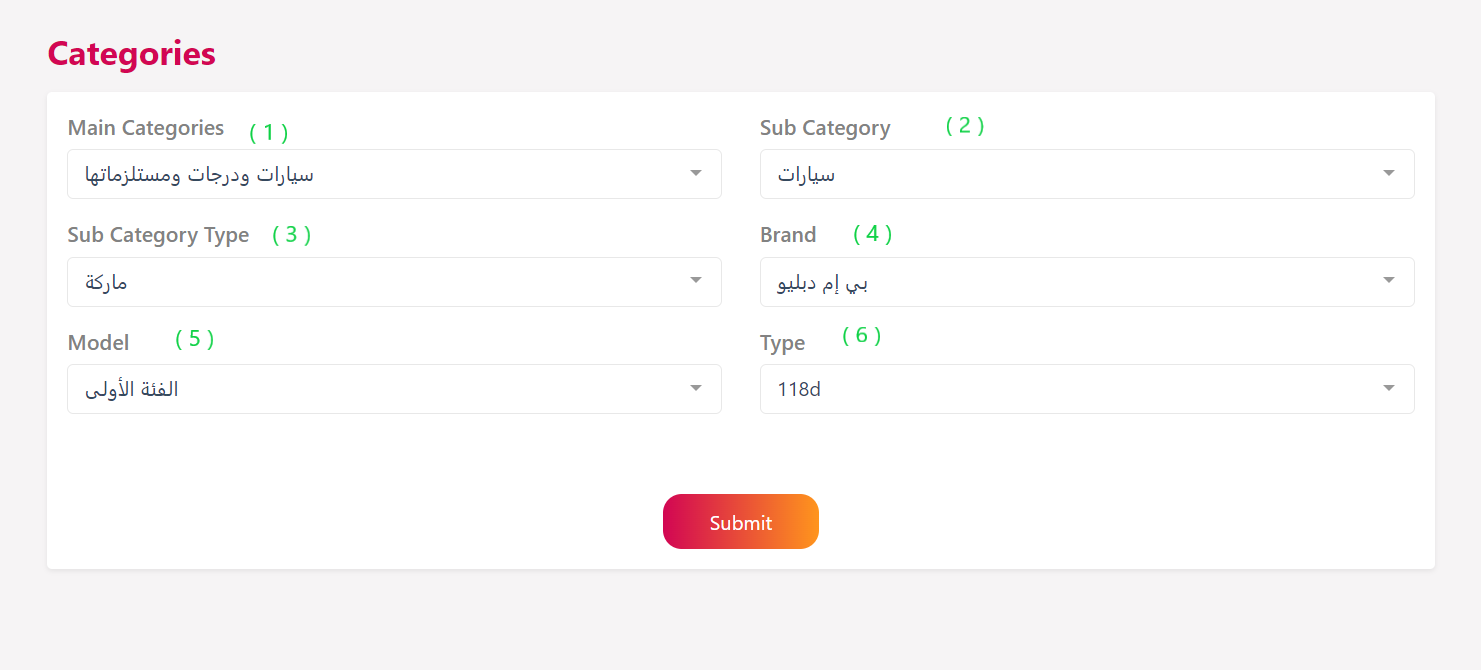
      required: true,

    },

  },

This refer to the main categories data and pass it as a prop when use this component inside the categories page .

This image shows the fields inside the form



Let’s see the data inside the main form component to know the purpose from these numbers

  data() {

    return {

      selectedMainCategory: null,

      subCategories: [],

      selectedSubCategory: null,

      subCategoryProperties: [],

      selectedSubCategoryProperty: null,

      selectedSubCategoryPropertyOption: null,

      isAnotherOption: false,

      otherOptionValue: null,

      PropertyOptions: null,

      selectedPropertyOptions: null,

      PropertyOptionTypes: null,

      selectedPropertyType: null,

      subCategoryEmpty: true,

    }

  },

So the numbers and dropdown options with this things

1. Main Categories

* V-model – selectedMainCategory
* Dropdown options – contains all categories that we get it from this api <https://staging.mazaady.com/api/v1/get_all_cats> through the categories prop

1. Sub Category

* V-model – selectedSubCategory
* Dropdown options – depends on the main category’s children

1. Sub Category Type