

# Tenkesalja Animal Welfare Protection Association/Tenkesalja Állatvédő Egyesület Report

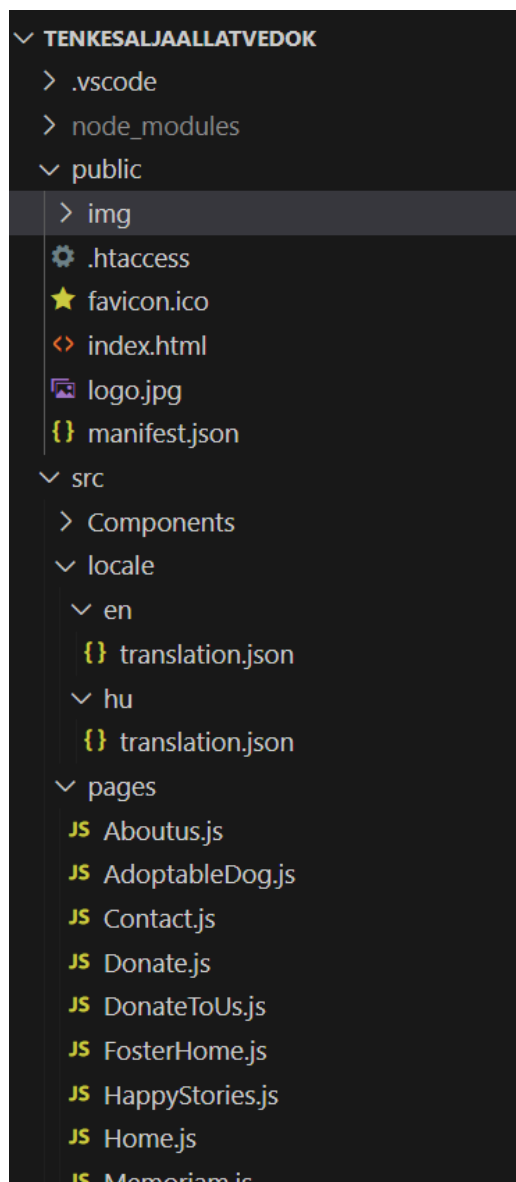
Author: Saqib Ahmad Syed

# Table of Contents

Run down of the Structure .....	3
Adding/Edit Text .....	4
/adopt Page.....	5
/in-memoriam Page.....	6
/happy-stories Page .....	7
Conclusion .....	8

This report will inform about how to update the picture or text for the content of the website “tenkesaljaallatvedok.com” for both English and Hungarian. It will give step by step instruction on how to change pictures/texts and how to add new content for the adopt, in memoriam and happy stories pages.

## Run down of the Structure



After opening the project, you can see the “locale” folder, the “pages” folder and the “img” folder. We will only need the content of these two files for what we need. The locale folder contains the translation.json file for both English and Hungarian language. For the pages folder, we will only use AdoptableDog.js (/adopt), Memoriam.js (/in-memorial) and HappyStories.js (/happy-stories). For the img folder, we will be storing all the images inside. Note that all the images need to be in .jpg format so that it can be displayed on the website later on.

## Adding/Edit Text

To change or add the texts, we first need to find where we want to change our text. For example, we will use the AdoptableDog.js file that contains all the information that will show the content to the website.

```
<div className='container p-5'>
  <div className='row justify-content-center'>
    <div className="col-lg-4 mb-4">
      <CustomCard
        imgUrl={"../img/falco.jpg"}
        title={t('adopt.subtitle-1')}
        body={t('adopt.text-1')}
      />
    </div>
  </div>
```

“t(‘adopt.subtitle-1’)” displays the subtitle and “...text-1” displays the content. To change or add new text, we need to go to the translation.json file for both en and hu.

```
"adopt": {
  "title": "Adopt",
  "subtitle-1": "Adopt a dog",
  "text-1": "He is",
  "subtitle-2": "Adopt a dog",
  "text-2": "Unfo",
  "subtitle-3": "Adopt a dog",
  "text-3": "Bobi",
  "subtitle-4": "Adopt a dog",
  "text-4": "Poco",
  "subtitle-5": "Adopt a dog",
  "text-5": "Jack",
},
```

Here we can see the “adopt” section of the json file that stores the English translation. We can adjust or add new text in this file. If we want to add a new adoptable dog for example, we will add “subtitle-6” and “text-6”. Inside the AdoptableDog.js, we will then add “t(‘adopt.subtitle-6’)” and “t(‘adopt.text-6’)”. “adopt” represents “adopt:” in the json file which encapsulates the whole text translation for the “adopt” page. “title”, “subtitle-1”, etc, represents the content of the “adopt” page. We will do the exact same process in the Hungarian json file except we will add the text translated in Hungarian accordingly. If we want to edit any text, we will have to go to these json files and change the content.

## /adopt Page



```
<div className="col-lg-4 mb-4">
  <CustomCard
    imgUrl={"../img/falco.jpg"}
    title={t('adopt.subtitle-1')}
    body={t('adopt.text-1')}
  />
</div>
```

To add a new information (new dog/cat in this case), All we have to do is to copy all the text shown in this image and paste it under “</div>”. It will look like this:



```
<div className="col-lg-4 mb-4">
  <CustomCard
    imgUrl={"../img/falco.jpg"}
    title={t('adopt.subtitle-1')}
    body={t('adopt.text-1')}
  />
</div>
<div className="col-lg-4 mb-4">
  <CustomCard
    imgUrl={"../img/bella.jpg"}
    title={t('adopt.subtitle-2')}
    body={t('adopt.text-2')}
  />
</div>
```

In the “CustomCard”, we can add the img url of our downloaded image stored in the /img folder, add the subtitle, and add the content of the text.

## /in-memorial Page

```
<div className="row justify-content-center">
  <div className="col-lg-4 mb-5">
    <div className='text-center'>
      
    </div>
    <p>{{t('memorial.text-1')}}</p>
    <p>{{t('memorial.text-2')}}</p>
    <p>{{t('memorial.text-3')}}</p>
  </div>
```

To add new information, we need to copy everything that is inside the “<div className=“row justify-content-center”>” including itself. Then, we will paste under where the div ends for the one that has been specified earlier. This will create a new row under the current one but it will duplicate the data inside the row.

```
<div className="col-lg-4 mb-5">
  <div className='text-center'>
    
  </div>
  <p>{{t('memorial.text-1')}}</p>
  <p>{{t('memorial.text-2')}}</p>
  <p>{{t('memorial.text-3')}}</p>
</div>
```

The image above shows the information that is displayed in one of the columns. To change the content, we can change the “src” path in the <img> tag and change the text accordingly. Keep in mind that if we want to have a second column with new information, we need to add “<div className='col-2'></div>” in between the two content so that it adds a gap in between.

## /happy-stories Page

```
{  
  [  imgUrl: "../img/puffi.jpg",  
    title: t('happy-stories.subtitle-1'),  
    body: t('happy-stories.text-1')  
  ],  
}
```

To add new information to this page, all we have to do is to copy everything in the bracket and paste under the comma to create new data just like the picture below.

```
{  
  [  imgUrl: "../img/puffi.jpg",  
    title: t('happy-stories.subtitle-1'),  
    body: t('happy-stories.text-1')  
  ],  
  {  
    imgUrl: "../img/picur.jpg",  
    title: t('happy-stories.subtitle-2'),  
    body: t('happy-stories.text-2')  
  },  
}
```

Then we change the path of the image to whatever image we want, and we change the text accordingly. It will do everything itself after.

## Conclusion

To conclude, this is an in-depth rundown on how to change or update text and images of the three pages for people with or without web development knowledge. This is only a report for the three pages. Also keep in mind that adding new information should have no affect on the style of the website as well as the mobile/tablet display of the website as well. If you follow the steps properly, you should be successful in changing or adding new text or images. If there are any confusion, chatgpt is a great resource to use.