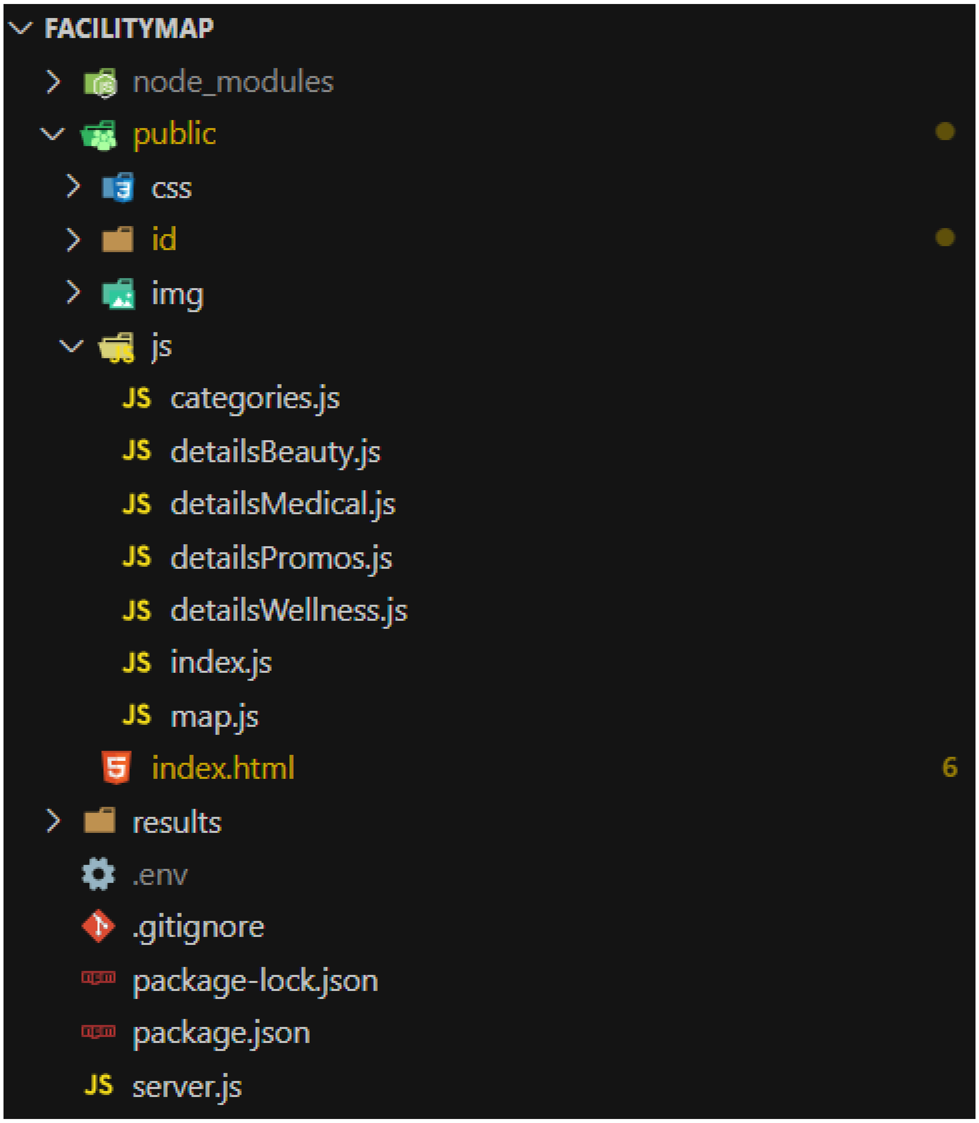
Medical Hub Map Maintenance Guide

This guide concerns the addition of,

1. *places*
2. *categories,*
3. *subcategories, &*
4. *promos*



```
FACILITYMAP
  node_modules
  public
    css
    id
    img
    js
      categories.js
      detailsBeauty.js
      detailsMedical.js
      detailsPromos.js
      detailsWellness.js
      index.js
      map.js
    index.html                                   6
  results
  .env
  .gitignore
  package-lock.json
  package.json
  server.js
```

Image I. *File Structure*

I. Basics

This guide was made in order to make it possible for those inexperienced in javascript to update the Medical Hub Map without much issue. However, it is still recommended for personnel experienced in javascript to check over once it is cleared to be deployed as a measure to prevent unwanted behavior.

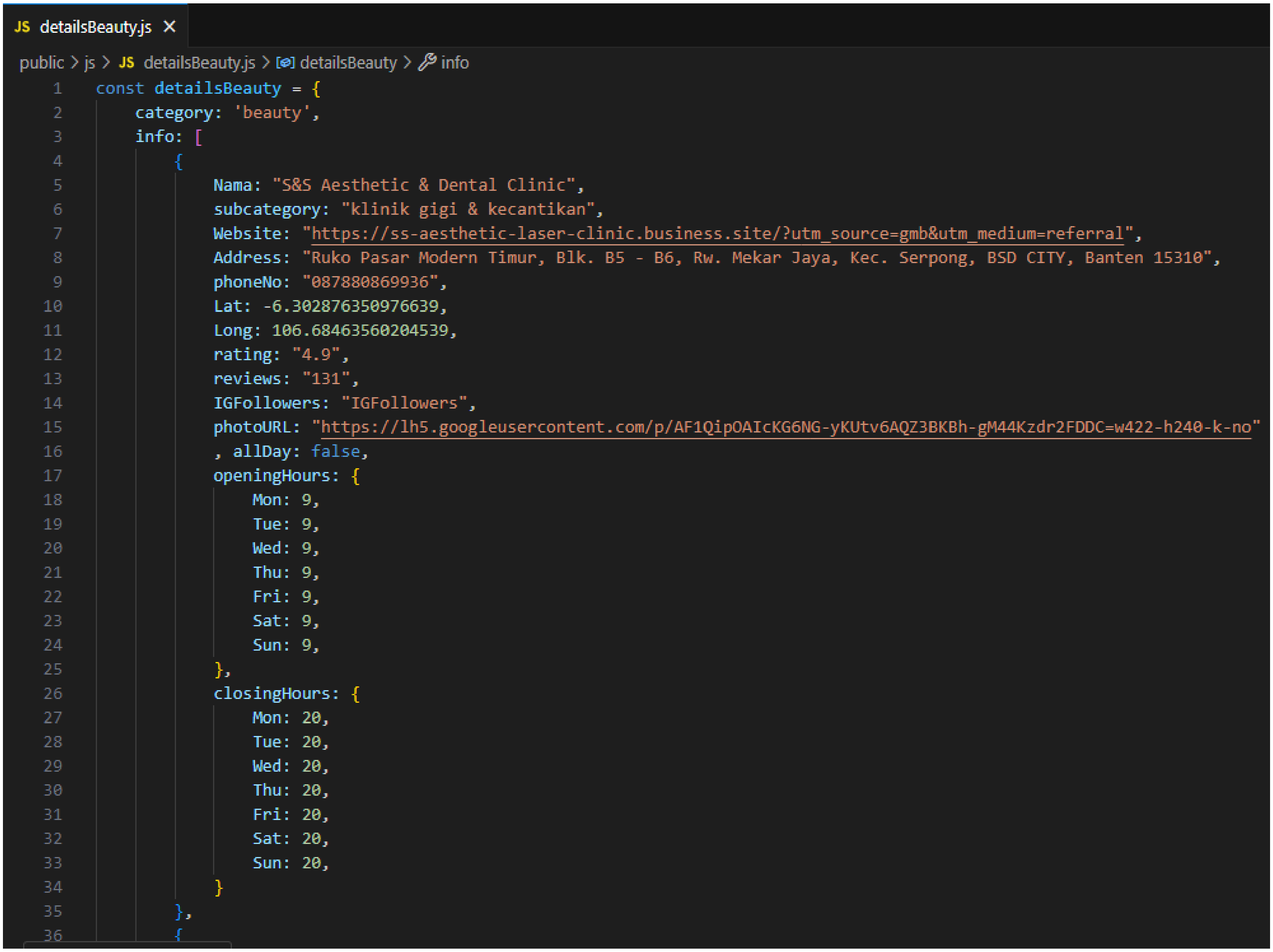
This map utilizes the Google Maps API in order to function, mainly to provide the basic static map along with the directions from a user's location. It should be understood that other than that, everything else present on the map is created from the files within the project. Thus, in order to preserve the project, most files in the project should not be modified in any way.

In order to add categories, subcategories, locations, and/or promos, the files that can be edited are the files found in the 'js' subfolder, located within the public folder in the project. Caution should be taken whilst editing the files, and they must do so only after a backup has been created.

In addition, informing IT personnel is also required in the case of an issue arising.

II. Adding Places

The simplest action that can be done is the addition of places to a predetermined category. Other than determining what category a location should be counted in, the steps to add a place involves adding it to the file of said category.



```javascript
const detailsBeauty = {
    category: 'beauty',
    info: [
        {
            Nama: "S&S Aesthetic & Dental Clinic",
            subcategory: "klinik gigi & kecantikan",
            Website: "https://ss-aesthetic-laser-clinic.business.site/?utm_source=gmb&utm_medium=referral",
            Address: "Ruko Pasar Modern Timur, Blk. B5 - B6, Rw. Mekar Jaya, Kec. Serpong, BSD CITY, Banten 15310",
            phoneNo: "087880869936",
            Lat: -6.302876350976639,
            Long: 106.68463560204539,
            rating: "4.9",
            reviews: "131",
            IGFollowers: "IGFollowers",
            photoURL: "https://lh5.googleusercontent.com/p/AF1QipOAIcKG6NG-yKUtv6AQZ3BKBh-gM44Kzdr2FDDC=w422-h240-k-no"
            , allDay: false,
            openingHours: {
                Mon: 9,
                Tue: 9,
                Wed: 9,
                Thu: 9,
                Fri: 9,
                Sat: 9,
                Sun: 9,
            },
            closingHours: {
                Mon: 20,
                Tue: 20,
                Wed: 20,
                Thu: 20,
                Fri: 20,
                Sat: 20,
                Sun: 20,
            }
        },
        {
```

Image II. Example of a File of the ‘Beauty’ Category. Filename: detailsBeauty.js

New places must be formatted within brackets, as seen in Image II. Within these brackets are the attributes of ‘Nama’, ‘subcategory’, ‘Website’, ‘Address’, ‘phoneNo’, ‘Lat’, ‘Long’, rating’, ‘reviews’, ‘IGFollowers’, ‘photoURL’, ‘allDay, ‘openingHours’, and ‘closingHours’.

Despite this long list of attributes, the only ones that are mandatory to be filled in are ‘Nama’, ‘Subcategory’, ‘Address’, ‘Lat’, ‘Long’, ‘rating’, ‘openingHours’, and ‘closingHours’. These are the ones that are required for a basic addition of a place. However, it is recommended - however not required - for as much of the attributes to be filled in as possible.

Nama

Nama is the name of the place you will be adding, and it should be named properly. And in future reference, this name should be kept true and proper so that it could be easily cross referenced and used by the code.

Subcategory

The subcategory describes the subcategory the place is in. It should be noted that this subcategory, if it already exists, should be cross referenced with those that exist in the file ‘categories.js’ and in one of the arrays corresponding to the category. Due to the rather haphazard manner in which it has been constructed, the subcategory should be copypasted from ‘categories.js’. The addition of new subcategories will be covered in a future section.

Website

The website attribute is where you put the website link.

Address

The address attribute is where you provide the address of the location. This address can be as descriptive or as non-descriptive as you like, as the place’s location on the map will be decided by the ‘Lat’ and ‘Long’ attributes.

phoneNo

The phoneNo attribute is where you put the phone number that is associated with the location.

Lat

The latitude of the location of the place. This can be checked by opening Google Maps, right clicking, and taking the first number provided by Google Maps.

Long

The longitude of the location of the place. This can be checked by opening Google Maps, right clicking, and taking the second number provided by Google Maps.

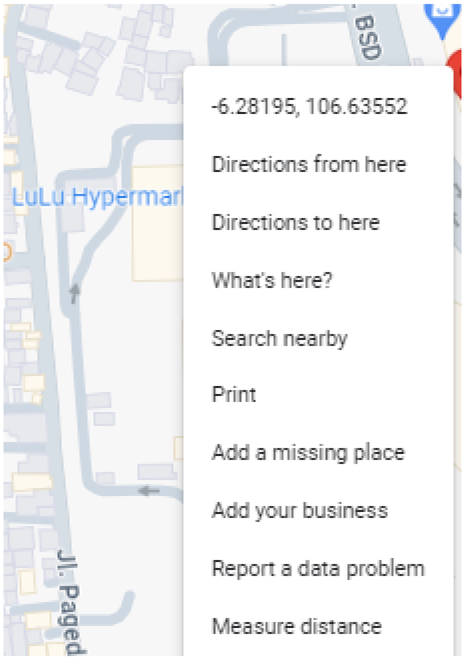


Image III. Latitude and Longitude. In this example, -6.28195 is longitude, and 106.63552 is longitude.

Rating

The rating attribute is for stating out of five stars, how many stars a location has. This rating is taken from Google Map's own rating system.

If this is not available, delete the attribute.

Reviews

The reviews attribute is to describe how many reviews were provided on the Google Maps page for it to create the Rating tag.

If this is not available, delete as well.

IGFollowers

The IGFollowers attribute is a non mandatory attribute, and this is used to describe how many instagram followers are on the instagram account associated with the location.

PhotoURL

The PhotoURL is a non mandatory attribute, and this is used to provide the program to the link of the photograph of the location. This means that it must be a ‘link’ to the photograph that ends in ‘.png’, ‘.jpg’, ‘.jpeg’.

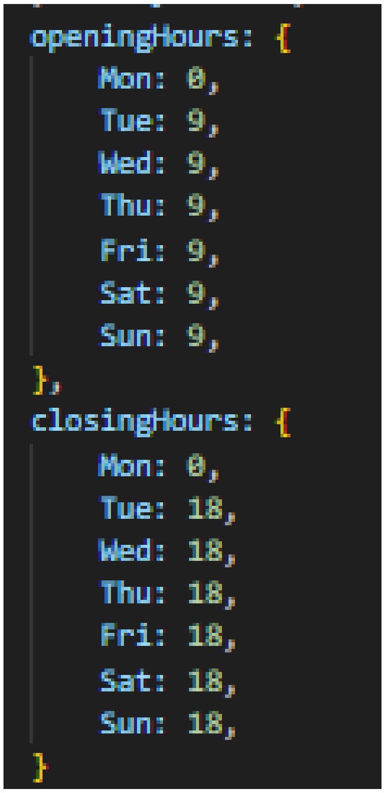
AllDay

Allday is a non mandatory attribute, and this is used to indicate if a location is open for 24 hours a day.

OpeningHours & ClosingHours

OpeningHours and ClosingHours are mandatory attributes, describing the opening and closing hours on a given day. They work on 24-hour time, and thus should use the 24 hour designation rather than the 12 hour designation.

It should be noted, to indicate if a place is closed on certain days, then the OpeningHour and ClosingHour for that particular day should be made the same.

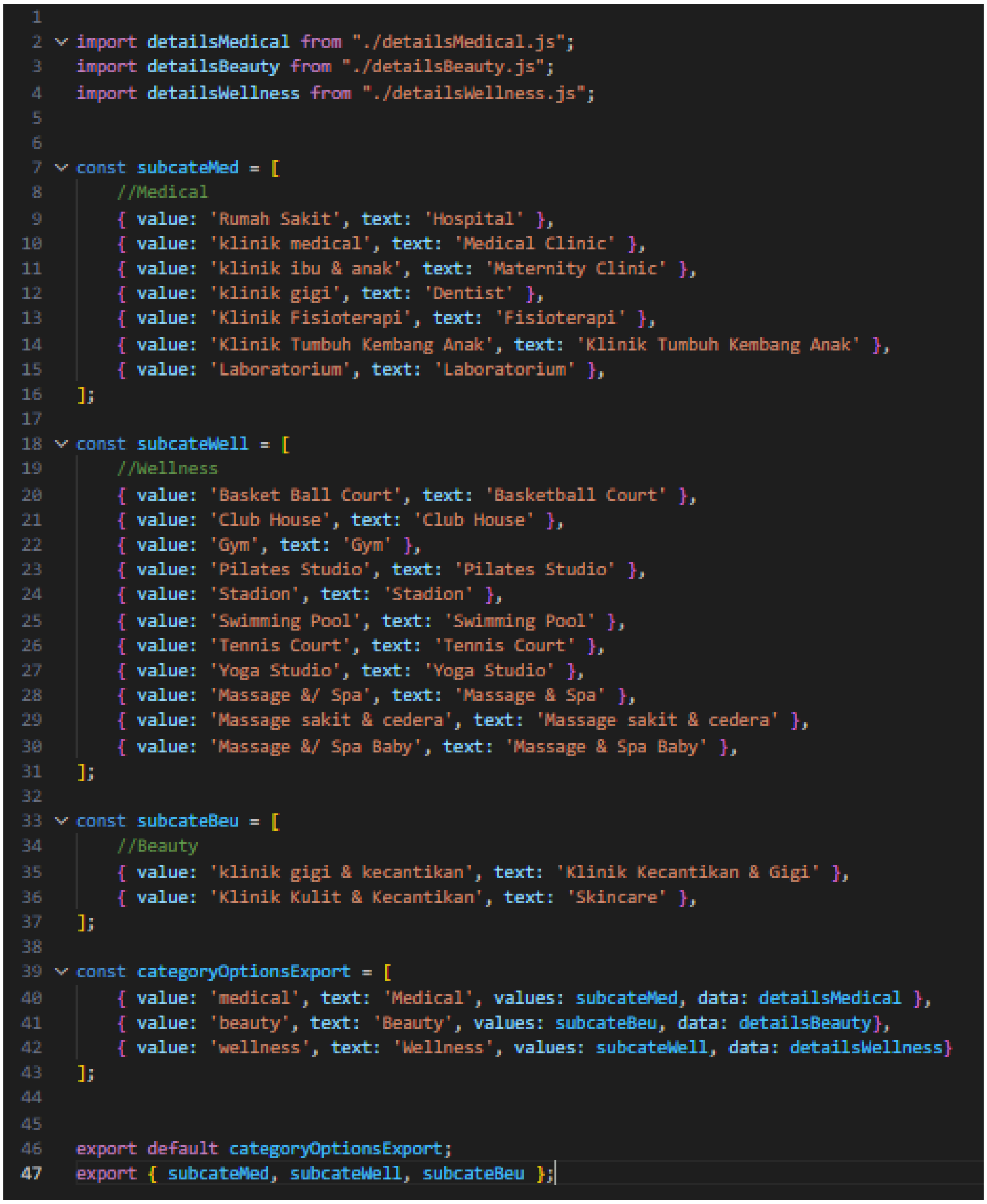


```
openingHours: {
    Mon: 0,
    Tue: 9,
    Wed: 9,
    Thu: 9,
    Fri: 9,
    Sat: 9,
    Sun: 9,
},
closingHours: {
    Mon: 0,
    Tue: 18,
    Wed: 18,
    Thu: 18,
    Fri: 18,
    Sat: 18,
    Sun: 18,
}
```

Image IV. An example of the usage of openingHours and closingHours. Here, it says the location is open Tuesday to Sunday from 9 to 18 o’clock. But Closed on Mondays as indicated by Mon in openingHours and closingHours using the same number of ‘0’.

III. Adding Categories.

To add categories, please navigate to the ‘categories.js’ file. Assuming there have been no changes to this file, this is how it will look like.



```

import detailsMedical from "./detailsMedical.js";
import detailsBeauty from "./detailsBeauty.js";
import detailsWellness from "./detailsWellness.js";


const subcateMed = [
    //Medical
    { value: 'Rumah Sakit', text: 'Hospital' },
    { value: 'klinik medical', text: 'Medical Clinic' },
    { value: 'klinik ibu & anak', text: 'Maternity Clinic' },
    { value: 'klinik gigi', text: 'Dentist' },
    { value: 'Klinik Fisioterapi', text: 'Fisioterapi' },
    { value: 'Klinik Tumbuh Kembang Anak', text: 'Klinik Tumbuh Kembang Anak' },
    { value: 'Laboratorium', text: 'Laboratorium' },
];

const subcateWell = [
    //Wellness
    { value: 'Basket Ball Court', text: 'Basketball Court' },
    { value: 'Club House', text: 'Club House' },
    { value: 'Gym', text: 'Gym' },
    { value: 'Pilates Studio', text: 'Pilates Studio' },
    { value: 'Stadion', text: 'Stadion' },
    { value: 'Swimming Pool', text: 'Swimming Pool' },
    { value: 'Tennis Court', text: 'Tennis Court' },
    { value: 'Yoga Studio', text: 'Yoga Studio' },
    { value: 'Massage &/ Spa', text: 'Massage & Spa' },
    { value: 'Massage sakit & cedera', text: 'Massage sakit & cedera' },
    { value: 'Massage &/ Spa Baby', text: 'Massage & Spa Baby' },
];

const subcateBeu = [
    //Beauty
    { value: 'klinik gigi & kecantikan', text: 'Klinik Kecantikan & Gigi' },
    { value: 'Klinik Kulit & Kecantikan', text: 'Skincare' },
];

const categoryOptionsExport = [
    { value: 'medical', text: 'Medical', values: subcateMed, data: detailsMedical },
    { value: 'beauty', text: 'Beauty', values: subcateBeu, data: detailsBeauty},
    { value: 'wellness', text: 'Wellness', values: subcateWell, data: detailsWellness}
];


export default categoryOptionsExport;
export { subcateMed, subcateWell, subcateBeu };
```

Image V. Screenshot of Categories.js as of 2024-08-22

Please take note of this, as adding categories is hand in hand with adding subcategories. However, this section will mainly only care about adding categories, adding subcategories will be handled in the following section.

Observing Image V., in order to add a category, you need to add a subcategory section, and add a line to categoryOptionsExport. When adding it, make sure it is in this format.

{value: “category”, text: “Category”, values: subcategory, data: detailsCategory},

The value and text attributes contain the name of the category you are adding. The text attribute must be noted for as that is what the endusers will see on the website.

The values (with an s) attribute will be discussed in Section IV.

The data attribute will contain the name of the file from where the specific details of what locations are in the category. For ease of navigation, it is named ‘detailsCategory’, where category is to be changed out with the actual name of the category.

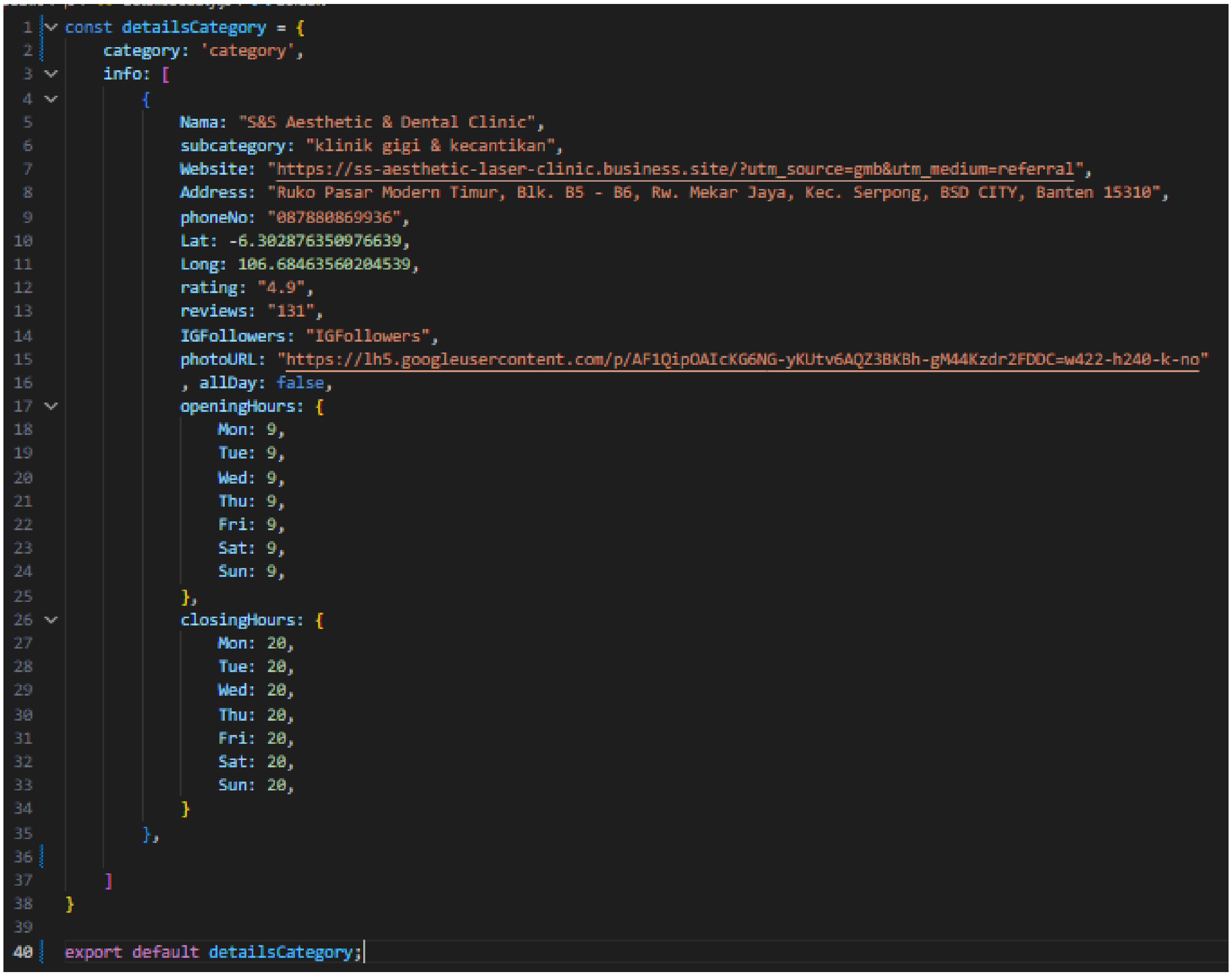
Once this is done, you must scroll back to the top of the file, and add a line of text

import detailsCategory from "./detailsCategory.js";

Category here is to be replaced with whatever category you are adding, so if you’re adding Hotels, it would be - import detailsHotel from “./detailsHotel.js”; .

Once this is done, go to the same folder where ‘categories.js’ is, and create a new file named “detailsCategory.js”. It is important you make sure that ‘.js’ is added at the end. This file will contain all the details of the locations, as according to the format provided in Section II.

As you can see in Image VI, the basic format is relatively simple.



```
const detailsCategory = {
    category: 'category',
    info: [
        {
            Nama: "S&S Aesthetic & Dental Clinic",
            subcategory: "klinik gigi & kecantikan",
            Website: "https://ss-aesthetic-laser-clinic.business.site/?utm_source=gmb&utm_medium=referral",
            Address: "Ruko Pasar Modern Timur, Blk. B5 - B6, Rw. Mekar Jaya, Kec. Serpong, BSD CITY, Banten 15310",
            phoneNo: "087880869936",
            Lat: -6.302876350976639,
            Long: 106.68463560204539,
            rating: "4.9",
            reviews: "131",
            IGFollowers: "IGFollowers",
            photoURL: "https://lh5.googleusercontent.com/p/AF1QipOAIcKG6NG-yKUtv6AQZ3BKBh-gM44Kzdr2FDDC=w422-h240-k-no"
            , allDay: false,
            openingHours: {
                Mon: 9,
                Tue: 9,
                Wed: 9,
                Thu: 9,
                Fri: 9,
                Sat: 9,
                Sun: 9,
            },
            closingHours: {
                Mon: 20,
                Tue: 20,
                Wed: 20,
                Thu: 20,
                Fri: 20,
                Sat: 20,
                Sun: 20,
            }
        },

    ]
}

export default detailsCategory;
```

Image VI. The simplest form detailsCategory.js make take

When creating a new file, you should have the following typed in.

```
const detailsCategory = {
        Category: ‘category’;
        Info: [
                {
                },
        ],
}

export default detailsCategory;
```

Unsurprisingly, detailsCategory should be changed to what you want it to be. In our example of creating a category for hotels, it would be modified to become as follows.

```
const detailsHotel = {
    Category: ‘hotel’;
    Info: [
        {
        },
    ],
}

export default detailsHotel;
```

And once that is done, you would have created a new category. The places should be added as according to the guide provided in Section II.

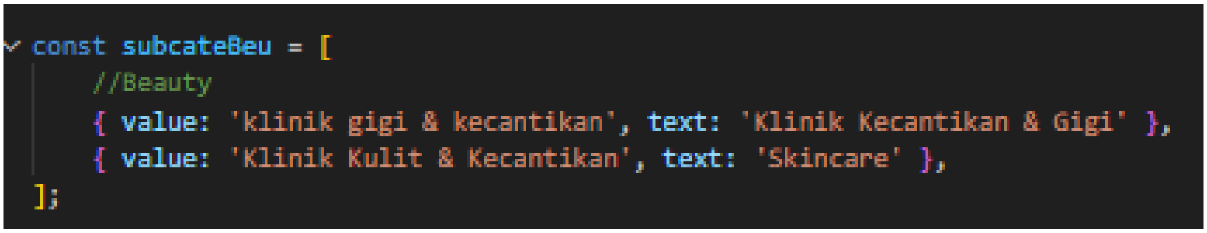
IV. Adding Subcategories

To add a subcategory, please navigate to ‘categories.js’.

As discussed in section III, the basic format for adding new Categories is as follows.

{value: “category”, text: “Category”, values: subcategory, data: detailsCategory},

Now, subcategory, in this line of text, actually refers to an array you have to make in the file. An example of which can be seen in Image VII.



```
const subcateBeu = [
    //Beauty
    { value: 'klinik gigi & kecantikan', text: 'Klinik Kecantikan & Gigi' },
    { value: 'Klinik Kulit & Kecantikan', text: 'Skincare' },
];
```

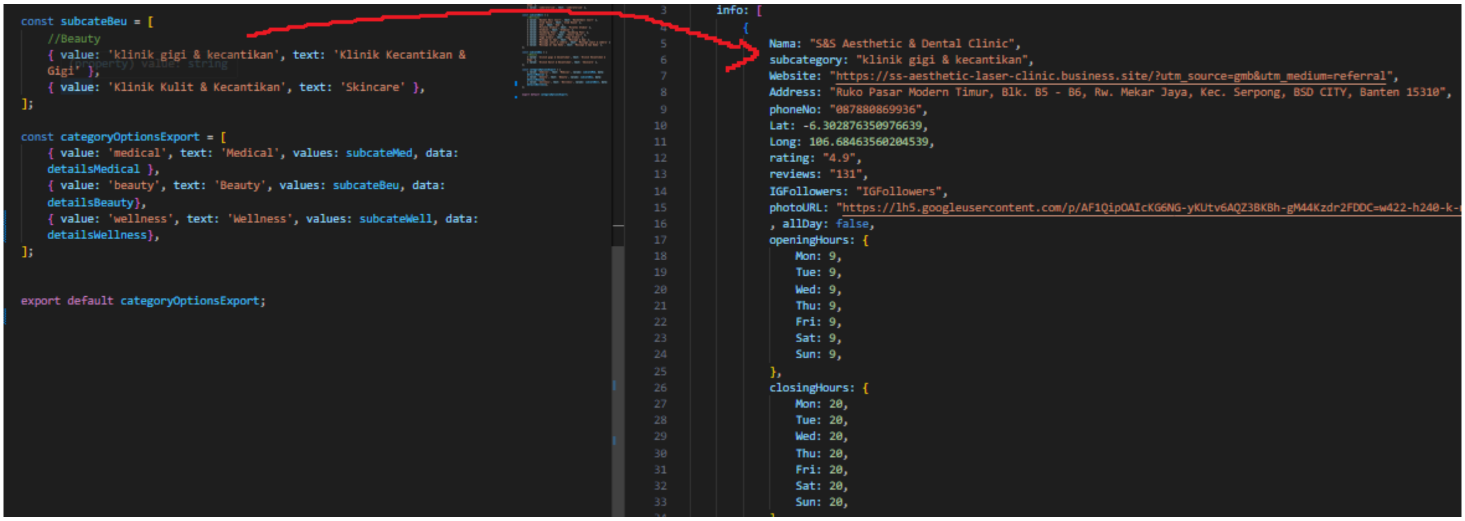
Image VII. Beauty Subcategories.

The name of the array can technically be anything you want, but for ease of understanding, the naming format should be ‘subcateCategory’. Where Category is changed out for the category you’re applying subcategories to. So if you’re subcategories for hospitals, it would be named ‘subcateHospital’.

You have to follow the format as presented in Image VII, but for ease here is a copy paste template.

const subecateCategory = [
{value: ‘subcategoryid’, text: ‘subcategoryname’},
];

It should be noted that value is for internal identification, and this will be what you fill in for a place’s subcategory, as seen in Subsection II. So, when adding a location, you fill in the attribute from the value subcategory, not from the text attribute. The text attribute is what the end user will see.



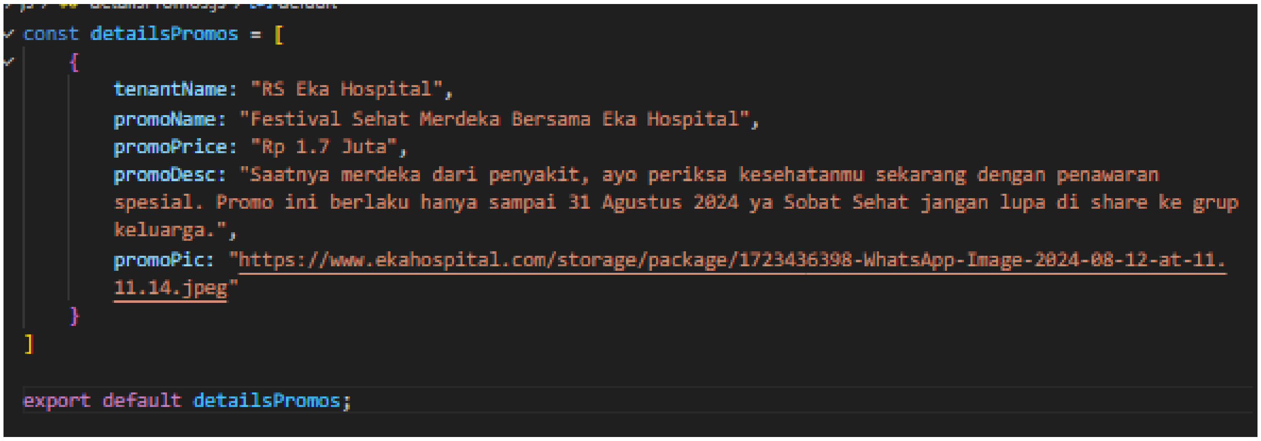
```
const subcateBeu = [
    //Beauty
    { value: 'klinik gigi & kecantikan', text: 'Klinik Kecantikan &
    Gigi' },
    { value: 'Klinik Kulit & Kecantikan', text: 'Skincare' },
];

const categoryOptionsExport = [
    { value: 'medical', text: 'Medical', values: subcateMed, data:
    detailsMedical },
    { value: 'beauty', text: 'Beauty', values: subcateBeu, data:
    detailsBeauty},
    { value: 'wellness', text: 'Wellness', values: subcateWell, data:
    detailsWellness},
];


export default categoryOptionsExport;
```

```
    info: [
        {
            Nama: "S&S Aesthetic & Dental Clinic",
            subcategory: "klinik gigi & kecantikan",
            Website: "https://ss-aesthetic-laser-clinic.business.site/?utm_source=gmb&utm_medium=referral",
            Address: "Ruko Pasar Modern Timur, Blk. B5 - B6, Rw. Mekar Jaya, Kec. Serpong, BSD CITY, Banten 15310",
            phoneNo: "087880869936",
            Lat: -6.302876350976639,
            Long: 106.68463560204539,
            rating: "4.9",
            reviews: "131",
            IGFollowers: "IGFollowers",
            photoURL: "https://lh5.googleusercontent.com/p/AF1QipOAIcKG6NG-yKUtv6AQZ3BKBh-gM44Kzdr2FDDC=w422-h240-k-
            , allDay: false,
            openingHours: {
                Mon: 9,
                Tue: 9,
                Wed: 9,
                Thu: 9,
                Fri: 9,
                Sat: 9,
                Sun: 9,
            },
            closingHours: {
                Mon: 20,
                Tue: 20,
                Wed: 20,
                Thu: 20,
                Fri: 20,
                Sat: 20,
                Sun: 20,
```

Image VIII. Example of Subcategory being used in a Place’s Subcategory.

It should be noted and emphasized that when being referenced in location, the system is Case Sensitive. So that means capitals and spaces should be done as exactly as it is written in ‘categories.js’.

V. Adding Promos



```
const detailsPromos = [
    {
        tenantName: "RS Eka Hospital",
        promoName: "Festival Sehat Merdeka Bersama Eka Hospital",
        promoPrice: "Rp 1.7 Juta",
        promoDesc: "Saatnya merdeka dari penyakit, ayo periksa kesehatanmu sekarang dengan penawaran
        spesial. Promo ini berlaku hanya sampai 31 Agustus 2024 ya Sobat Sehat jangan lupa di share ke grup
        keluarga.",
        promoPic: "https://www.ekahospital.com/storage/package/1723436398-WhatsApp-Image-2024-08-12-at-11.
        11.14.jpeg"
    }
]

export default detailsPromos;
```

Image IX. *detailsPromos.js*

Promos are specified in detailsPromos.js’. As you can see, it has five attributes. tenantName, promoName, promoPrice, promoDesc, and promoPic.

tenantName, promoName, promoPrice, and promoDesc are all mandatory. And they are relatively self explanatory in what they do.

Due to the way the system has been made, the tenantName must be exactly the same as it has been specified in system. So for this example, as in the image, the tenantName for Eka Hospital must be exactly the same as it was specified in detailsMedical.js, this is so that the system can actually understand that the promo is for that place specifically.

promoPic, functions the same as PhotoURL in section II, a link to the photograph that you want to attach for the Promo. Meaning it must end in either png, jpg, or jpeg. However, it is not mandatory.

Due to the way the system has been made, only one promo can be attached per location. So it is recommended that once a promo has run its course, it is to be deleted.

VI. Contact

If things go wrong, please contact your IT Department or to those in charge of maintaining this project in order to resolve the issues.