Frontend Engineer Wanted

# Description

Besides designing and developing software products, our team has many passions. One of them is sports, more particular, kitesurfing. The practitioner has a board and is pulled on the water by a kite. With speeds ranging between 30-50km/h, the most exciting feeling is going from gliding on water to flying over it.

Along the years, we gathered information about kiting spots around the world. To help other kiters, we want to develop a **responsive web** application which allows the user to visualize, filter and find out detailed information about these spots.

We need **an awesome engineer, like yourself,** who can develop this app using **Angular**, **React** or **Vue**. If you want to use something else, please confirm with us beforehand.

# Implementation details

## The responsive web app will connect to an already existing backend using an API. Details about this interaction can be found in the API section.

## Users will be able to log in by using email and password (LOGIN PAGE).

## After logging in, the user will see a screen (DASHBOARD) where information about all kitesurfing spots around the world (information needs to come from the API), in the form of:

## A map with pins for all spots fetched (suggestions for 3rd party libraries **LeafletJS, Google Maps, Mapbox**)

## Table with all the spots and their information: name, latitude, longitude, country, wind probability, month when it is kiting season at that spot.

## On the dashboard we will have a filter button. The user will press it and enter filter parameters: country and wind probability. The map and table will update accordingly.

## When the user taps a location on the map, a popup will appear (DETAILS PAGE), where detailed information will be shown about that spot.

## From the DETAILS PAGE, the user will be able to **ADD TO FAVOURITES**, by pressing a button. If the spot is already a user favourite, the user should have the option to **REMOVE FROM FAVOURITES**.

## On the map and in the list, favourite spots need to be marked differently, using a yellow color.

## In the DASHBOARD, top-right, a profile button will show a menu with a single action: “Log out”.

## In the Design section below, a general idea is presented for the design of our responsive web application.

## Nice to have/Bonus:

## The list should have a search and a sorting feature, on each column available.

## Responsive Design (the application should look good on both mobile and web)

## Login and Signup

## Functionality to add a new spot (name, select for country and period of the year when in season - with date pickers for start date and end date). You can find a general idea of the design in de Design section.

## Advice

Structure the application and code with the idea that you will work for at least 2 years on it, it will have tens of screens, much more features and API endpoints implemented. Also keep in mind that other programmers will work on the code as well.

We want you to write code that you are proud of and think there is no way to optimize it anymore :)

## Upload instructions

We want you to work under a private repository on gitlab.com and to make granular and often commits. When you finish, please invite the user @AB4Systems to your repository and offer „Developer” access type. Also, please send us an email telling us the project is ready for review.

## Some of the things we look for

1. Code readability and styling
2. How you apply OOP principles and structure the application
3. How you implement network requests (communication with the API)
4. How you handle errors
5. How/If you make sure that the application does not freeze when making an API request and make sure the interface is responsive during this process

# API

BASE URL: [**https://5ddbb358041ac10014de140b.mockapi.io/**](https://5ddbb358041ac10014de140b.mockapi.io/)

To find out the URL for any endpoint described below, just append the suffix of each one to the “BASE URL”.

Example: For the “spots list” we will do a GET Request at URL-ul [**https://5ddbb358041ac10014de140b.mockapi.io/spot**](https://5ddbb358041ac10014de140b.mockapi.io/spot)

We recommend you clone the API because it will be used everyone and it has a limit of 100 entries [Error: **„Max number of elements reached for this resource!”**]. Please use your clone to avoid this and there you can control the API limits per endpoint.

You can clone the API by creating an account on [www.mockapi.io](http://www.mockapi.io/) and opening this link <https://www.mockapi.io/clone/5ddbb358041ac10014de140c> in a browser and refresh the page. The project limit on a demo acount is 1 project, so make sure you are not trying to clone it multiple times.

If you modify the API, by adding entities or other endpoints, please tell what you changed, so we can take that into account when checking the project.

## Endpoints

The API is make using MOCK API service. Some information returned will be automatically generated by the service, so please that that into consideration when parsing the data.

* **Login** 
  + - **POST /login –** simulates a login process (any email or password will work ☺) and returns “userId” for an authenticated user, **useful for fetching data from the API**.
* **User**
  + - **GET /user –** All users list
    - **GET /user/:id –** Information about a particular userId “:id”
    - **POST /user –** Add a new user
    - **PUT /user/:id –** Modify an existing user
    - **DELETE /user/:id –** Delete an existing user
* **Spot** 
  + - **GET /spot –** All spots list
    - **GET /spot/:id –** Information about a particular spotId “:id”
    - **POST /spot –** Add a new spot
    - **PUT /spot/:id –** Modify an existing spot
    - **DELETE /spot/:id –** Delete an existing spot
* **Favourites**
  + - **GET /favourites –** All favourite spots for a user
    - **GET /favourites/:id –** Information about a favourite spot favouriteId “:id”; contains the original spotId inside
    - **POST /favourites –** Add a new favourite entry containing a spotId
    - **PUT /favourites/:id –** Modify an existing favourite entry
    - **DELETE /favourites/:id –** Delete an existing favourite entry

# Testing

Along with the problem, you will receive an archive containing a file called **Kitesurf.postman\_collection.json**

For testing, please install the free app Postman ([https://www.getpostman.com](https://www.getpostman.com/)) on your computer and import the file mentioned above. Once the app is installed, opened and the file is imported, you will be able to see the “Kitesurf” collection of endpoints in the left side-menu.

**If you have any issue using the API, please write to us. This is a mocked API created only for this problem and it is possible that we missed some things.**

# Design

Along with the problem, you will receive an archive with the Screenshots below, a folder of assets used and a .sketch design file (<https://www.sketchapp.com/>). Below you can see how the screens should look.

**The design is generic and only for guidance purposes. You have full liberty to change whatever you feel you need to for UI/UX reasons.**







