



# Web Frontend Entry Exercise

Dear Pietro,

First of all, we are pleased to see that you are interested in joining the Web team inside the DevOPS department in order to help IVAO and its community have the best possible experience on our network !

If you have any questions, do not hesitate to contact us on [web@ivao.aero](mailto:web@ivao.aero) and we will be happy to answer your questions.

## Main review points

**There is no perfect code nor answer for this type of exercise.**

We will be looking at the following points when reviewing your submission :

- Easy to read/understand : At DevOPS you will be working on projects with other developers that might look at your code months/years later. They need to be able to understand quickly how your code works
- Comment when necessary : You don't need to comment every single line. Your code should be self-explanatory but still comment the functions where complex processing is happening
- Good readability/optimization ratio : The projects you will be working on are going to be accessed by thousands of users around the world, that's why optimization is important for us but we don't want to over-complexify the code so it stays maintainable over time.
- Security flaws : The code deployed on our infrastructure is implementing basic security measures to prevent users accessing unauthorized data and impersonating other users. We don't expect state-of-the-art security on this exercise but you should keep it in mind.

# Subject

## Introduction

We are asking to code a simple Web App that allows users to book ATC positions (simpler version of the scheduling system that was available on the main website).

## Structure

You will need to follow some guidelines to build your website, close to those we use in the Web Team.

### Backend

Since this is an exercise to test your skills in frontend only, feel free to use what's the best for you, this part won't be taken into consideration.

The goal is to provide a simple backend for your frontend part so we can check how you make API calls.

### Frontend

The frontend should be built using [ReactJS](#) following, at least, the general coding guidelines provided in their documentation examples. You can stick to [CRA](#), no need to bother with other frameworks (Next, Remix, etc...)

You are free to organize your code and pages as you wish as long as it stays consistent.

You can use plain Javascript or Typescript for the code.



## Submission

Once you have completed the exercise, please send it to us in a ZIP file.

Please **DO NOT** include the *node\_modules* folder, as it is heavy and useless since we are going to install the dependencies locally

Feel free to add a README file to explain how to run it or anything you want to clarify. You can also provide a base SQL file if you want, but not required.

## User identification

We aren't asking for any advanced user authentication but we are asking to send user's VID as a HTTP header on all endpoints that need to identify a specific user (create/update/delete/list own booking).

We are going to consider that the end-user can't forge headers for security measures ;)

We are suggesting that you put an form input for the user to enter his VID but if you have a better idea, we are very open

## Frontend Design

You are allowed to use any CSS library (bootstrap, tailwind, etc...) in your frontend.

We aren't looking for the perfect design, but something that would follow general IVAO guidelines (like [fpl.ivao.aero](https://fpl.ivao.aero) or [tracker.ivao.aero](https://tracker.ivao.aero)) can be a good thing. You can use [brand.ivao.aero](https://brand.ivao.aero) if you want to have a general idea.

We are also very interested to see how you take User Experience in account when building this website. It should be pretty straightforward for the user to understand how to use the web app.

## Required features

Here are the main features your website should implement. Please read carefully what is asked, it is not here to trick you but to guide you.

If you want, you can add additional features but this won't directly impact your application since it's not mandatory. Ideas of improvements will be given at the end.

## List all future bookings

### Public feature

Display in a simple table all future bookings, sorted by position identifier, with the following information :

- Date
- From/To Date and Hour
- User's VID
- Position identifier (callsign)

## List all bookings for a specific date

### Public feature

Same as before but for any specific date, including those in the past.

## Book a position

### User-identified feature

The frontend should have a simple form to book a position, similar to the old one with :

- Position Identifier
- From/To Date and Hour

## Edit a booking

### User-identified feature

Allow the user to edit one of his future bookings.

## Delete a booking

### User-identified feature

Allow the user to delete one of his future bookings.

## Additional ideas

As stated previously, adding features isn't mandatory, we prefer having good core features than spending too much time on less important features.

Here are some ideas you can implement, but if you have other ideas, you are more than welcome to do it :

- Set the booking as a training, event or exam
- Display user's rating (AS3, ADC, ACC, etc...)