

Challenge - Desk Booking System

Coding School

February 2024

General information

The challenge called Desk Booking System deals with a platform on which it should be possible to book workstations in different rooms. A web app needs to be developed that enables the **admin user to create rooms including workstations**. Workstations should be able to be classified as FixDesk or FlexDesk. **FixDesk are assigned to a specific user** and cannot be booked by other users. FlexDesk are accessible to everyone and can be booked **for a week at a time**. Attention, as soon as the booking slot **goes over the weekend, the booking cannot be carried** out. E.g. Monday - Friday works, Thursday - Tuesday does not work.

You can decide for yourself when choosing the architecture. If an architecture is chosen, the entire project must be implemented in it.

The entire challenge must be implemented in React (please use TypeScript and no JavaScript). Tailwind CSS and TypeScript need to be compressed.

The entire challenge should also be responsive and have an appropriate design for mobile devices.

levy

This time too, the challenge will be completed as a group project and must be managed in a GitLab repository and submitted in its entirety.

The last commit on the master branch before the submission deadline counts as a submission.

Submission deadline: Friday March 8th, 2024 11:59 p.m

Requirements

Screen designs & project planning

No screen designs will be provided for this challenge. The screens should be implemented using common web components. Parts of it should have already been conceived by you in the UI/UX lessons last week.

For the web app, a **mockup concept** must be created in a tool of your choice (Balsamiq, Figma, etc.), which should represent the structure, structure and functionality of the web application.

In addition to the mockups, as with the second challenge, **project planning** including boards and tasks must also be done via GitLab. Plan and create the board carefully on the first day so that you can then work well together as a team.

Both designs and project planning count for 10% each in the overall evaluation.

API documentation / Swagger

During implementation, the following API should be connected. The documentation can be accessed via the following links:

<https://deskbooking.dev.webundsoehne.com/api/docs>

To import the documentation into Postman you can use this link: <https://deskbooking.dev.webundsoehne.com/api/docs-json>

Basic concept / navigation

The web app should have navigation that can be used to **navigate between favorites, booking plan, reservations, administration (only for admin users) and the user's profile.**

Feature description

Login / Registration

Login and registration takes place via HTTP endpoints of the API. When registering, a user should be able to provide their data (first name, last name, email, ...). The **login token should be saved locally** after a successful login so that the user remains logged in after a restart.

Favorites

All favorite tables of the logged in user are displayed in this view. Workplaces should also **be able to be booked quickly from there**. Optionally, the **reserved days of the week** can also be stored in the favorites so that the user can complete the booking with as few clicks as possible.

Booking plan

All available rooms are displayed here. From the respective room the user gets to the **table overview of the room**. All tables are displayed there - **Fix & FlexDesk**. Tables that **have been booked** should therefore be marked as not bookable. FlexDesks that have already been booked must also be displayed and marked accordingly. If you want to book a **FixDesk, this booking must be confirmed by an** admin. FlexDesk can be booked by anyone.

The following data should be recorded when booking:

- **Name & email of the person making** the booking - should be pre-filled and not editable
- **Reservation period - max one week at a time** / active reservations

When you click on a table you also get **information about the equipment**.

Reservations

The user should receive a **list of all future or already made reservations**. For reservations that have already expired, it should be possible to record any deficiencies in the equipment using the **comment function**. These **comments should appear on admin users to be considered**.

administration

This section is only available for admin users.

Here, user **comments can be displayed** and **FixDesk bookings can also be confirmed**.

Bonus: In the API documentation there is also an endpoint for **adding new rooms/offices as well** as tables. If the logic is implemented, bonus points are awarded.

profile

In the profile settings, all profile data can be changed and the user can be logged out.