

## CHALLENGE



A school wants software that allows them to create rooms/courses. They need a screen with a form to create the room and another screen where they can view the details of the room and the people who are part of it. The school's second requirement is the ability to add students to the system, specifying which room they belong to, their age, gender, and any other relevant data. Finally, the school expects to be able to view a student's page by clicking on it, to see their personal information, including whether the student has a sibling in another course and who this sibling is. A student may have one or more siblings, and this information would be used to apply a discount on the fee if applicable.

They need an admin section to make the Create and Edit operations on rooms and students, it will be a plus if you make user authentication to this section.

### Requirements:

- CRA, Vite, or custom webpack for the front end infrastructure, taking into account for the architecture of the project hierarchy and organization of directories, routing, conventions and good practices of clean code, good design patterns for the app itself but also for the react components
- For the app state, React Hooks
- Add a descriptive "readme" file, it should also include links to any resources and documentation used to solve the challenge.

### The web should behave as follows:

- The landing page will show the rooms already created. Plus: add search by name
- A sidebar will be available to go to the admin section. Plus: add user authentication to this page
- When the user clicks a room, a page with room details and a list of the students will be displayed. Plus: if there is a session in auth state, you could add an edit room button here
- When clicking on a student, a page will be displayed with the personal information of that student and a list of relatives that the student has in the school as other students. Plus: add storage to the backend and allow to upload profile image to the student info
- In the admin section, the admin should be capable of creating, editing, and editing rooms and students
- Deploy in Vercel or any alternative of your choice would be a plus, otherwise describe in detail the steps to execute the frontend pointing to the deployed backend.



### The backend should behave as follows:

- CRUD endpoints for Room and student entities
- Make use of an ORM like Supabase, Sequelize, or any of your choice.
- Connect with a database
- Deploy in Heroku, AWS, or any of your choice.

## Evaluation

The challenge is simple, so we will hold the solution to the highest quality standard, considering:

- Aesthetic, clean code (use linter)
- Senseful folder structure
- Scalable solution

### Upon completion:

Upon completion: Submit your solution to a private GitHub repository. Keep the organization of branches and commits as if you were working in a team.

Give access to the following list of GitHub users:

- fedecaccia
- LucaCevasco

Include in Readme file explanation on env vars required for running the project. Include also some pictures or video showing the project working.

Deploy the application to vercel, netlify or any free hosting service for quick testing