

CELUS UI Challenge

This challenge is intended for **UI Engineers**, and consists of improvements and the analysis of a fictional web application.

The project is composed of a simple *Single Page Application* built in *React* and using *Redux*. The back-end is a simple mocked server providing a *JSON API*.

The challenge is divided into tasks. If you have any questions, please direct them to **career@celus.io**.

Relevant information:

- Please write your own code, and do not copy solutions from elsewhere.
- Please make sure the project works when you submit it.
- The **README.md** file on the project will provide you with instructions on how to run the application. You should start with running it and seeing how it works.
- Everything should look good on mobile devices.

What is expected:

- Employ best practices when writing code. Code quality, functionality and performance are of the utmost importance to us.
- Make sure to have all the variables and functions **typed** correctly with TypeScript (avoid using **any**).
- Tests

Background

One Penguin loves to travel and wanted to share the adventure with his/her friends, so he/she created a traveling blog.

Penguin wants your help with some tasks.

Task 1: Bug in the System

Penguin's friends reported that they could no longer access the individual posts.

He is also confused about why his logo doesn't appear next to the title.

Could you help Penguin figure out what the issues are and fix them?

Task 2: Style and implement UI requirements

Now, with the posts working as expected, some of Penguin's friends sent the feedback that it would be nice to be able to comment on posts. One of them even provided a prototype, it must be a Designer friend!

Penguin agrees that it would be a good feature to keep in touch with the friends. So he asks for your help with that.

In order to make the proposed changes, he also adds one requirement: the specific post should be a page instead of a dialog.

Please follow the prototype and implement the needed changes.

You can access the prototype via this [link](#). You'll need to login in order to see the components' specifications and requirements (e.g. sizing, font, ...)

Note that the comments list should be **scrollable**.

Task 3: General Improvements

Penguin started this app a while ago, so there are still some components with an old code structure.

Could you modernize those components? Penguin mentioned something about Header...

Task 4: Freestyle

As a proud developer, you should leave your mark on the application! Make a change that you believe would add value to the project.

It can be anything, anywhere.

FAQ

- *For Task 2, how do I know with which user I am logged in for commenting?*

Assume that you are one of the existing users already on the server (you can check [api/db.json](#) for more details). Just pick one of those in your logic, no need to implement a login flow.

- *Does the existing server support a comment endpoint?*

Yes, our server uses the file at [api/db.json](#) as the base for its responses.

The server resources are available at: <http://localhost:8080/>, in there you can check what is currently being provided.

For more information about how it works, check [json-server](#) docs.