

Hey! 🙌

We're happy for you to start working on our test assignment!

Our [verification engine](#) is based on checks. Every verification session has a set of required checks that must be answered for the decision to be made. You would need to implement an answering checks interface – the part of our [Verification Tool](#):

Images not submitted from a device screen

Yes No

Images not submitted from a printout

Yes No

Submitted document appearance matches that of official specimen

Yes No

Document data cross-reference successful

Yes No

SUBMIT

To start, please, fork [this codesandbox](#) and read the detailed description below. There is also a demonstration video attached to the email.

## Details

In `api.js` file you can see a simple implementation of mock API with 2 functions:

- `fetchChecks` – returns a list of checks in the format `{ id: string, priority: number, description: string }[]`
- `submitCheckResults` – intended to submit check results in the format of `{ checkId: string, value: string }[]`

You need to fetch and display a list of checks, allow them to be answered as Yes / No and display the Submit button. When Submit is clicked, check results will be submitted and show a success screen.

Use `fetchChecks` function in `api.js` to fetch checks. Displayed checks should be **sorted by priority** (1.. 2.. 3.. etc). Each check should show its description and possible answers – Yes and No.

Images not submitted as a slideshow

Next checks should be **disabled** unless all checks above are answered. Submit button becomes available when either:

1. All checks are answered as Yes
2. At least one check is answered as No

Checks can be answered by either mouse (click on Yes or No) or via **keyboard shortcuts**. Keyboard shortcuts work as follows:

- `Arrow Up` and `Arrow Down` to move between checks
- `1` to answer check as Yes
- `2` to answer check as No

The demonstration video attached to the email shows all the logic including the keyboard shortcuts.

---

We expect you to **follow the provided design**. You may include tests to cover your implementation as well. You can use TypeScript and any libraries you think are needed. Also, you can definitely use the preferred IDE instead of Codesandbox.

If you have any questions, please ask them here and we will be happy to answer them. You are not limited by time, but please send us an estimation of when to expect the completed task.

As you complete the task, please send us back a link to it (Codesandbox, Github repository, or whatever you prefer) and a brief description of the solution.

Good luck 🍀