



# Oowlish React/JS Challenge

## Javascript Developer

Dear Candidate,

First, we are happy you get this far on our hiring process, congratulations. We believe that interviews are probably not the best place to test the coding skills of a promising candidate. We created this challenge for you to show your coding skills. We hope you enjoy it as much as we did.

This challenge was created in a way to give you maximum freedom so that you can show us the best of your expertise. Roughly you have up to two weeks to send it in.

### Objective:

Your task is to build a web app to control the working hours for a company's employees. The app will be used by each employee to manage and keep account of:

### Info:

- **Worked** hours;
- **Arriving** and **Exiting** hours;
- **Lunch** breaks;

### Features:

- Input:
  - **Arrive** and **Exit** times;
  - **Lunch** break times;
- Show the **amount** of hours above or below the **expected working hours**;

### Requirements:

- Use **React**;
- Save (and fetch) data to an external REST API (you can use something like json-server or Firebase if you want).

### Non-required features:

- Well written documentation.
- Setup an authentication mechanism;
- Automated (CI-driven) deployment;
- Tests;
- Build your own API (if you do, it should be in one of PHP, Python or Node);

### How to submit:

1. Create a **README.md** with the instructions for us to run your project;

2. Send us your code using a private **Gitlab** repository. Add to the project the user @oowlish-career as a "Maintainer";
3. Send the repo link replying your application email to [career@oowlish.com](mailto:career@oowlish.com);

*We kindly ask you not to publish your implementation, to avoid online spoilers.*

Feel free to use any libs you deem necessary, but know that we'll also consider your choices around this topic.

## Our review

The focus of the review is on answering some questions such as:

- How clearly can you separate concerns in your code?
- Are SOLID and DRY principles violated?
- How good have you designed your models? Do they make sense?
- Are your docs clear?
- Does your solution contain any dead/redundant code?
- Does your application launch?
- Does the application do what it promises?
- Can we find bugs or minor flaws?
- Did you provide a complete product ready to deploy?

### **The following things most probably won't impress us:**

- Overengineering;
- Implementing additional features;
- Using global variables;
- Typos, grammar mistakes in a variable naming.

**That said, good luck and happy coding! ;)**