# **Xeal Software QA Challenge**

Thank you for taking the time to complete this challenge! This directory consists of a basic working app for which you are requested to set up a simple testing infrastructure, extract the main user flows, and implement tests.

Please read below for the context and instructions.

## Application Requirements

### Background

Adam runs his own web development school, in which he teaches the fundamentals of the web to his students, over the course of several months.

Adam wishes to maintain an easy overview of the progress of his students. He wants to be able to quickly get a simple list of repos owned by a given student, and see the most basic details of each repo. He also wants to be able to easily navigate to a specific repo, in case he needs to dive deeper.

Adam likes Github, but the UI is not built for a minimalistic, non-paginated overview of repos. For this he decides to build his own simple app, leveraging the Github API.

### Acceptance Criteria

1. The UI consists of a header, a search form and a search result section
2. The header displays the title of the app
3. The search form accepts a text input as a search term. Search is activated by clicking the "Go" button, or by pressing the "Enter" key
4. For each repo found, the search result section displays a row with basic info about that repo: name and description. Clicking on the repo name takes the user to the repo's URL. In case of a missing value, `–` is displayed
5. The user sees feedback about the result of the search action. Either a success or error message are shown above the search field at the completion of a search action, for a short amount of time. If the error is due to a user not found on Github, a specific error message is displayed. Otherwise, a generic error message is displayed

## Challenge Goals

Adam has finished the first version of the app. It meets the requirements and serves him nicely, but is still very basic. He would like to enhance the app over time in iterations, adding more features. To do that well, he decides he needs confidence to continuously enhance the app without breaking it or losing one of the fundamental requirements.

### Instructions

Help Adam gain confidence in further development by automatically testing the most critical features of the app, by doing the following:

1. Download the directory, setup and run the app (see **setup.md** file for instructions)
2. Using the product description, acceptance criteria and the app itself, compile a list of several testable user flows, cases or scenarios. This doesn't have to cover every possible real-life case, please focus on what you think is most fundamental
3. Based on your assessment of importance and effort, prioritize this list. Imagine this list would be later converted to tickets to be tackled in the next few work iterations of a product development team. Please explain your reasoning briefly. Add this list and your reasoning to the empty **testable-flows.md** file
4. Out of these flows, choose two and implement tests for them
5. Create a **README.md** file and add instructions on how to run these tests

## Additional information

* Use any testing technology you'd like – Cypress, Selenium or any other you think would work well
* Feel free to modify or add whatever you need including addition of other packages, your own libraries, etc

## Challenge Submission

There are two options for submitting your challenge project:

1. Zip the entire project directory and upload the archive to the provided google drive folder.
2. Provide access to a PRIVATE git repository (github, gitlab, etc…) and reply to the challenge email with the link.