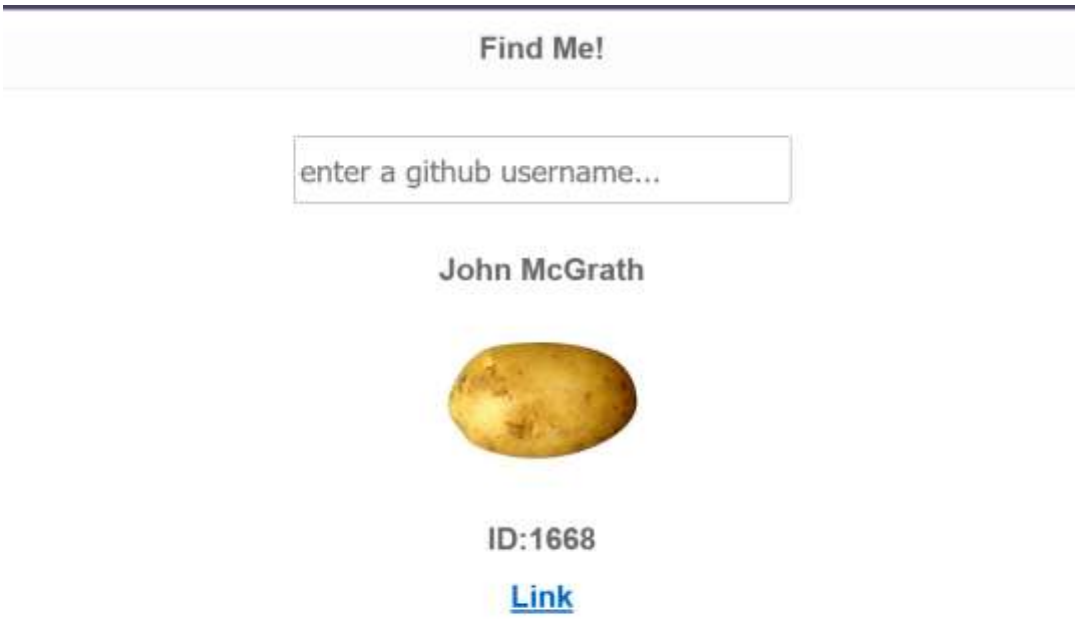
ICP-4: APIs and AJAX



I. Introduction

In this assignment the requirements were as follows: we were asked to design and webpage that provides and API to find a GitHub user. The skills for this ICP were JavaScript (OO) and jQuery, being JavaScript one with bigger code load to add behavior to the page. Also, technologies as HTML to draft the page and CSS to adjust the image size.

II. Description

For this project we divided the task into two parts: the API interface and the HTML displaying code. The API code fetches the data from the Github API and then stores it as a JSON. Then the display code takes that, extracts the data from the JSON, and displays it as HTML elements using AJAX selection and display toggling..

II. Learning from the lesson

In this ICP we learnt how to work use JavaScript when we want to add behavior to our page, and we learnt how to aggregate interactions with HTML file. Working with the display side help to understand better how all this technology merge themselves together to make the page more interactive and easier to use.

III. Code Snippets

```
function getGithubInfo(user) {
    //1. Create an instance of XMLHttpRequest class and send a GET request using it.
    // The function should finally return the object(it now contains the response!)
    const ROOT = "https://api.github.com/";
    let URL =  ROOT + "users/" + user;
    let request = new XMLHttpRequest();
    request.open( method: "GET", URL,  async: false);
    request.send();
    return request;
}
```

For the API code it was a little obtuse because the XMLHTTPRequest method requires things like the GET method specified as a string, and the request to be formed and then sent. However otherwise it was straightforward as APIs go.

```
function showUser(user) {
    //Creates a new user struct to store relevant request information
    if (user["name"] == null){
        this.name = user["login"];
    }
    else{
        this.name = user["name"];
    }
    avatar = user["avatar_url"];
    id = user["id"];
    link = String(user["html_url"]);

    //2. set the contents of the h2 and the two div elements in the div '#profile' with the user content
    $('#id').text('ID:' + id);
    $('#name').text(name);
    $('#avatar').attr("src", avatar);
    $('#avatar').show();
    $('#html_url').text("Link");
    $('#html_url').attr("href", link);
}
```

Then we extract the response data we care about from the parsed JSON, the methods for which was provided in the template code.

Github Repository