ICP 4: jQuery & APIs

Joe Moon
Email: jmn5y@umsystem.edu
Github: https://github.com/joemoon-0/WebMobile-2022Spring/tree/main/2022Spring-Web/Web_ICP4

Nathan Cheney
Email: ncxn8@umsystem.edu
Github: https://github.com/nathancheney/Web-Mobile-Programming/tree/main/ICP-4

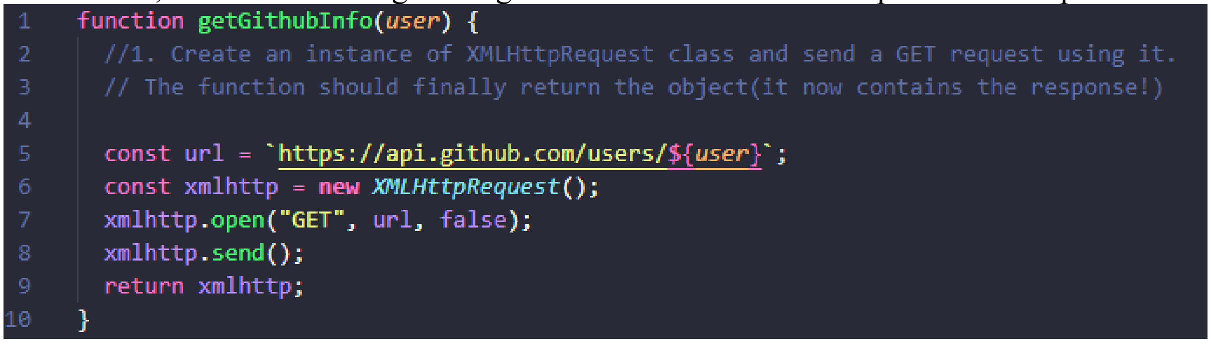
Introduction

With websites/web apps performing increasing complex operations, JavaScript libraries are constructed to help streamline such operations for easier development. jQuery is one such example of a JavaScript library which is designed to simplify HTML DOM tree traversal/manipulation, along with event handling, CSS animation, and AJAX requests.

In addition to increased complexity, web apps draw upon APIs for information/content which can also be facilitated by JavaScript libraries. This ICP will explore use of both.

GitHub User Finder

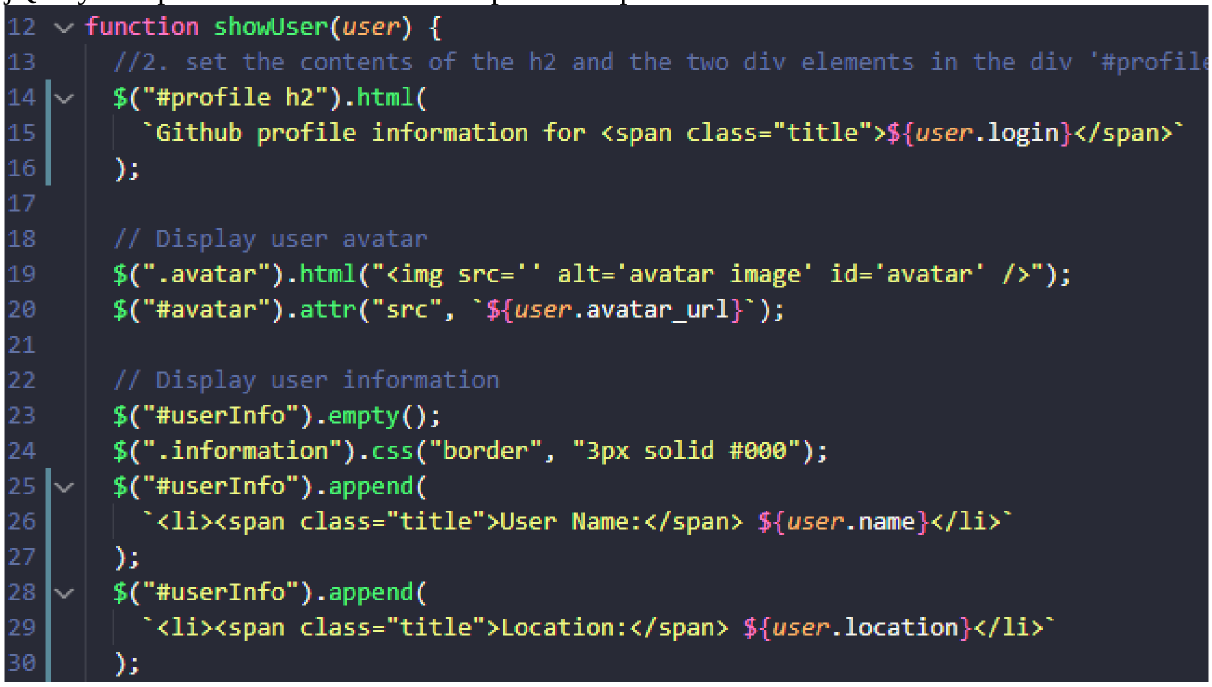
This task involved using a simple text-field interface that would receive a GitHub user's login name and then search, through GitHub's API, for that user's information. Using the given source code, a HTTP request was made to retrieve the corresponding user information, with the user's login being a retrieved from the UI and passed in as a parameter:



```
1   function getGithubInfo(user) {
2     //1. Create an instance of XMLHttpRequest class and send a GET request using it.
3     // The function should finally return the object(it now contains the response!)
4
5     const url = `https://api.github.com/users/${user}`;
6     const xmlhttp = new XMLHttpRequest();
7     xmlhttp.open("GET", url, false);
8     xmlhttp.send();
9     return xmlhttp;
10  }
```

As shown on line 5, the user's login (provided through the UI) is appended to the API endpoint from which a GET request is made to read that information (if it exists). Regardless of what is received, it is saved into the **xmlhttp** variable and then returned to the driving function.

If the user is found, information pertaining to that user’s account is displayed on the UI using jQuery that pulls data from the JSON parsed response of the API.

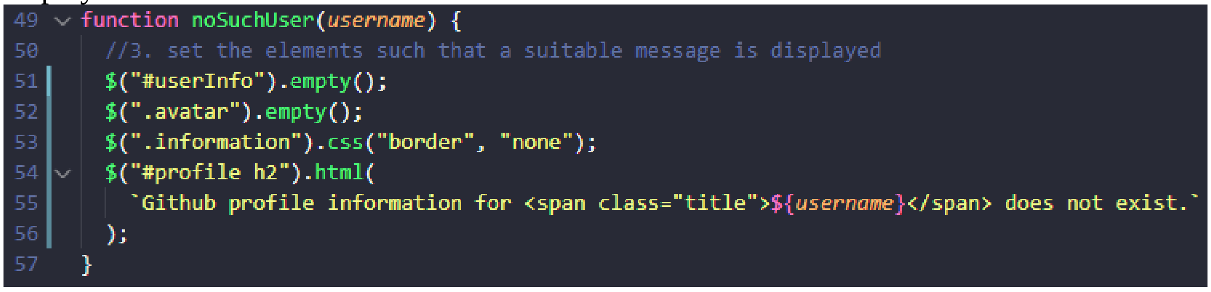


```
function showUser(user) {
  //2. set the contents of the h2 and the two div elements in the div '#profile
  $("#profile h2").html(
    `Github profile information for <span class="title">${user.login}</span>`
  );

  // Display user avatar
  $(".avatar").html("<img src='' alt='avatar image' id='avatar' />");
  $("#avatar").attr("src", `${user.avatar_url}`);

  // Display user information
  $("#userInfo").empty();
  $(".information").css("border", "3px solid #000");
  $("#userInfo").append(
    `<li><span class="title">User Name:</span> ${user.name}</li>`
  );
  $("#userInfo").append(
    `<li><span class="title">Location:</span> ${user.location}</li>`
  );
```

Alternatively, if the user does not exist, then the UI is cleared and a corresponding message is displayed.



```
function noSuchUser(username) {
  //3. set the elements such that a suitable message is displayed
  $("#userInfo").empty();
  $(".avatar").empty();
  $(".information").css("border", "none");
  $("#profile h2").html(
    `Github profile information for <span class="title">${username}</span> does not exist.`
  );
}
```

<u>Contribution</u>

All members contributed equally to this report and ICP.