Anusha Nelluri – anhx2@umsystem.edu

GitHub Link - https://github.com/NelluriAnusha/Demo_Remote/tree/main/Webpart/ICP4

Achyuth Kumar Valeti – avgh3@umsystem.edu

GitHub Link - https://github.com/AchyuthValeti/Demo_Remote/tree/main/Webpart/ICP4

ICP4 (JavaScript, jQuery)

Introduction:

Javascript can be used to create both client-side and server-side applications. It is used to improve the responsiveness and creativity of websites. It's not just for website development; it's also utilized in numerous desktop and server programs and certain databases.

JQuery is a javaScript framework that evolved from JavaScript. It is the most widely used JavaScript library, and it is both free and open-source, as well as cross-browser compatible. The goal of jQuery is to make it easy for the public to create webpages and browser-based applications using javaScript. In simple terms, "JQuery is a library to deliver a better client-side web page development" environment to the developer in a clear manner.

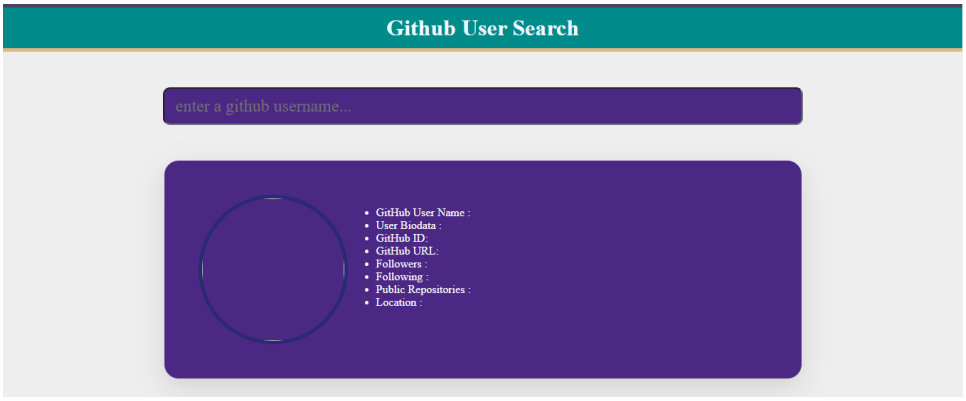
Tasks:

1.Github User Finder:

We created a web application by learning the basics of the GitHub API, then displaying the data on the front end.

This web application will help us in retrieving and printing GitHub User information depending on the data(username) given in the input text box.

We used HTML, CSS, JavaScript and jQuery for developing the following web application.



We have followed the steps below to create this web application.

1. Firstly, we have taken the source code and followed the ToDo list mentioned in all files.
2. In HTML file, we have gone through the provided code and updated the text in specific fields.
3. We also used CSS for printing data with different colors and properties.
4. Using javascript, we established a connection to the GitHub API Server.
5. Here, we have taken help from XMLHttpRequest which will help us to connect to GitHub's API service and request data. The main advantage of XMLHttpRequest is that they are built-in almost all browsers, and they allow us to request, receive, and transfer data without having to refresh the page.
6. So, username which we entered in input text field was passing as an argument to "getGithubInfo" function. We can dynamically pass any github username which we want to get the info for.
7. In this function, I have created an instance for 'XMLHttpRequest' class and sent a GET request using it. And assigned the GitHub URL to one new variable along with 'username' argument.
8. Then, we need to open our connection and declare that we will be using a GET request rather than a POST request, which means we will be sending data, because we are requesting data from GitHub.

Here is the code for establishing a connection to GitHub's server:

```
//Instantiating a new instance of an XMLHttpRequest
const xHttpReq = new XMLHttpRequest()
//dynamically passing specified username to GitHub
const user_url = `https://api.github.com/users/${user}`

//opening a new XMLHttpRequest connection, using a GET request with user_url
xHttpReq.open( method: 'GET',user_url, async: true);
```

9. After we've established a connection to the GitHub API, we can use the '**.onload'** function to define what we want to do with our data.
10. Here, we also need to send our request to GitHub using **'.send'** function.
11. Once our request has been received, we need to run a function to parse the data. And, we need to parse the returned data into JSON like below.

```
// Once the request has been received, we need to execute the following steps
xHttpReq.onload = () => {
    if(xHttpReq.status == 200) {
        //parsing API user data into JSON
        const user_data = JSON.parse(xHttpReq.response)
        showUser(user_data)
    } else{
        noSuchUser(username)
    }
}
```

12. When User information is found on GitHub API, then **'showUser'** method will be called otherwise **'noSuchUser'** method will be invoked.
13. In '**showUser'** method, we used *document.getElementById() method for* setting the contents of html tags with the user content.

```
document.getElementById( elementId: 'nameOfTheUser').innerText = user.login;
document.getElementById( elementId: 'userBiodata').innerText = user.bio;
document.getElementById( elementId: 'IDofUser').innerText = user.id;
document.getElementById( elementId: 'githubimage').src = user.avatar_url;
document.getElementById( elementId: 'userLink').href = user.url;
document.getElementById( elementId: 'userLink').innerText = user.url;
document.getElementById( elementId: 'userFollowers').innerText = user.followers;
document.getElementById( elementId: 'userFollowing').innerText = user.following;
document.getElementById( elementId: 'userPublicRepos').innerText = user.public_repos;
document.getElementById( elementId: 'userLocation').innerText = user.location;
```

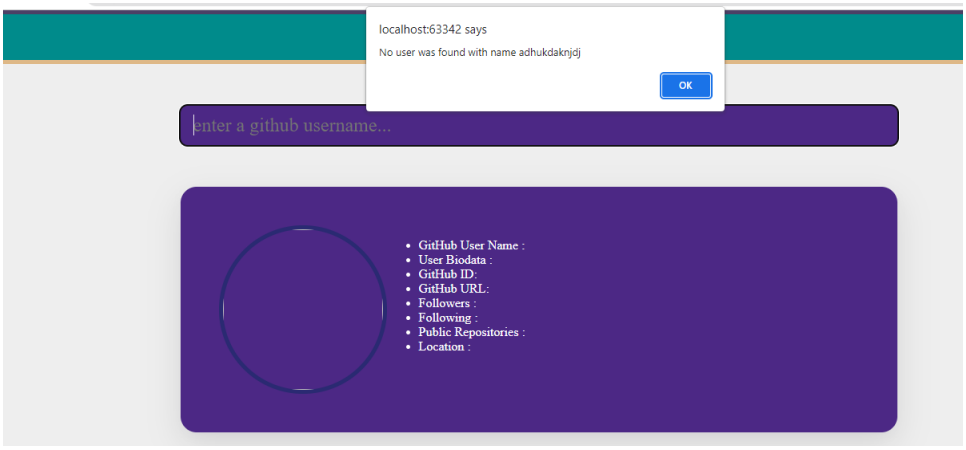
14. When user information was not found on the server, then we were displaying the same information and refreshing the page.

```
function noSuchUser(username) {
    //3. set the elements such that a suitable message is displayed
    window.alert("No user was found with name "+username);
    window.location.reload();
}
```

15. Finally in HTML, we are retrieving & printing all User Information & images using the tags which were assigned with the user values in **showUser**() method.

```
    <img id="githubimage" class="avatar">
<div class="information">
    <ul class="user-info">
        <li>GitHub User Name : <strong id="nameOfTheUser"></strong></li>
        <li>User Biodata : <strong id="userBiodata"></strong></li>
        <li>GitHub ID: <strong id="IDofUser"></strong></li>
        <li>GitHub URL: <strong id ="userLink"></strong></li>
        <li>Followers : <strong id="userFollowers"></strong></li>
        <li>Following : <strong id="userFollowing"></strong></li>
        <li>Public Repositories : <strong id="userPublicRepos"></strong></li>
        <li>Location : <strong id="userLocation"></strong></li>
    </ul>
</div>
```

Here is the final output for our webpage:



Contribution:

We have contributed equally.

Conclusion:

In this ICP, we have learned javascript, jQuery and developed a web application using the same.

Challenges:

We have not faced any major challenges while doing the assignment.