//Inheritance and fetch

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>GitHub Repo Details</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 20px;

padding: 20px;

background-color: #f4f4f4;

}

ul {

list-style-type: none;

padding: 0;

}

li {

background: #fff;

margin: 5px 0;

padding: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

</style>

</head>

<body>

<h1>GitHub Repository Details</h1>

<input type="text" id="username" placeholder="Enter GitHub Username">

<button id="fetchButton">Fetch Repos</button>

<ul id="repoList"></ul>

<script>

class GitHubClient {

constructor(username) {

this.username = username;

this.baseUrl = 'https://api.github.com/users/';

}

async fetchRepos() {

try {

const response = await fetch(`${this.baseUrl}${this.username}/repos`);

if (!response.ok) {

throw new Error('Network response was not ok');

}

const repos = await response.json();

return repos;

} catch (error) {

console.error('Fetch error:', error);

return [];

}

}

}

class GitHubRepoDetails extends GitHubClient {

constructor(username) {

super(username);

}

async displayRepos() {

const repos = await this.fetchRepos();

const repoList = document.getElementById('repoList');

repoList.innerHTML = ''; // Clear previous results

if (repos.length === 0) {

repoList.innerHTML = '<li>No repositories found.</li>';

return;

}

repos.forEach(repo => {

const li = document.createElement('li');

li.textContent = `ID: ${repo.id}, Name: ${repo.name}, Repo URL: ${repo.html\_url}`;

repoList.appendChild(li);

});

}

}

document.getElementById('fetchButton').addEventListener('click', () => {

const username = document.getElementById('username').value;

const githubDetails = new GitHubRepoDetails(username);

githubDetails.displayRepos();

});

</script>

</body>

</html>