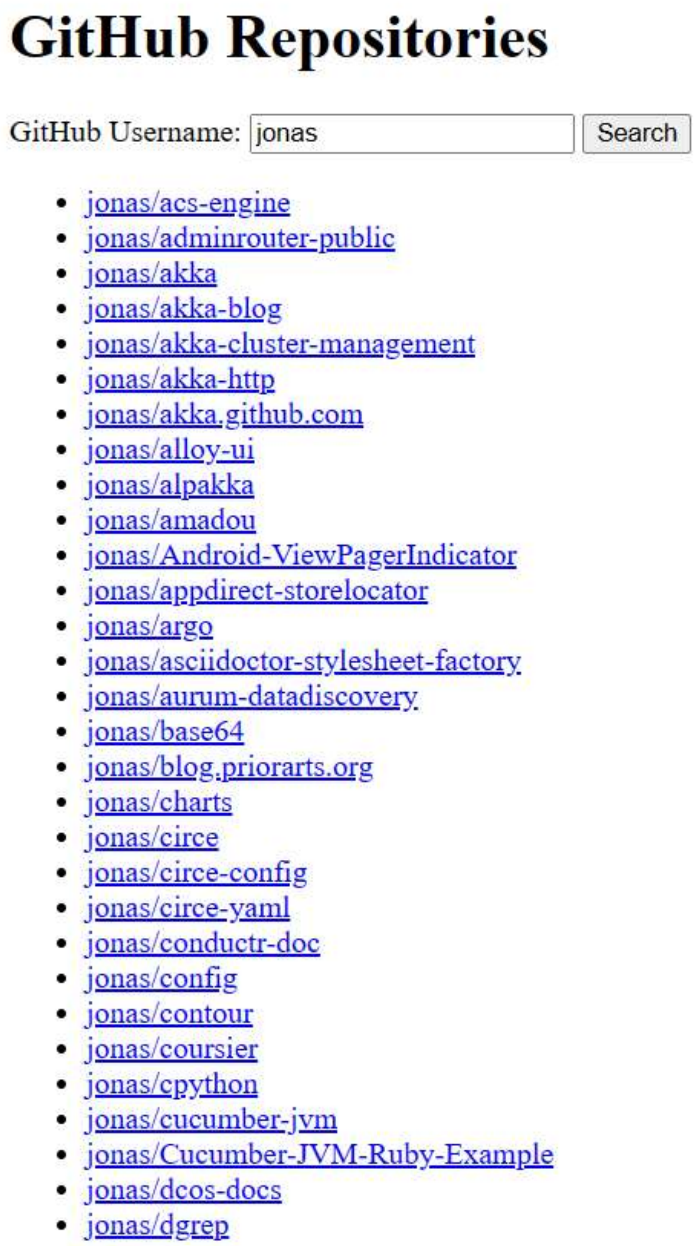
1. Github repo

Create a web page that displays a list of GitHub repositories for a given user. The web page should have an input field where the user can enter a GitHub username. When the user clicks a "search" button, the page should use the GitHub API to retrieve the repositories for that user and display them in a list on the page.

(https://api.github.com/users/${username}/repos)

```
<!DOCTYPE html>
<html>
  <head>
    <title>GitHub Repositories</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>GitHub Repositories</h1>
    <label for="username">GitHub Username:</label>
    <input type="text" id="username" name="username">
    <button id="search-btn">Search</button>
    <ul id="repos-list"></ul>

    <script>
      const searchBtn = document.getElementById("search-btn");
      const reposList = document.getElementById("repos-list");

      searchBtn.addEventListener("click", async () => {
        const username = document.getElementById("username").value;
        try {
          const response = await
fetch(`https://api.github.com/users/${username}/repos`);
          const repos = await response.json();
          reposList.innerHTML = "";
          for (const repo of repos) {
            const repoItem = document.createElement("li");
            repoItem.innerHTML = `<a href="${repo.html_url}"
target="_blank">${repo.full_name}</a>`;
            reposList.appendChild(repoItem);
          }
        } catch (error) {
          reposList.innerHTML = `<li>${error.message}</li>`;
        }
      });
    </script>
  </body>
```

```
</html>
```



GitHub Repositories

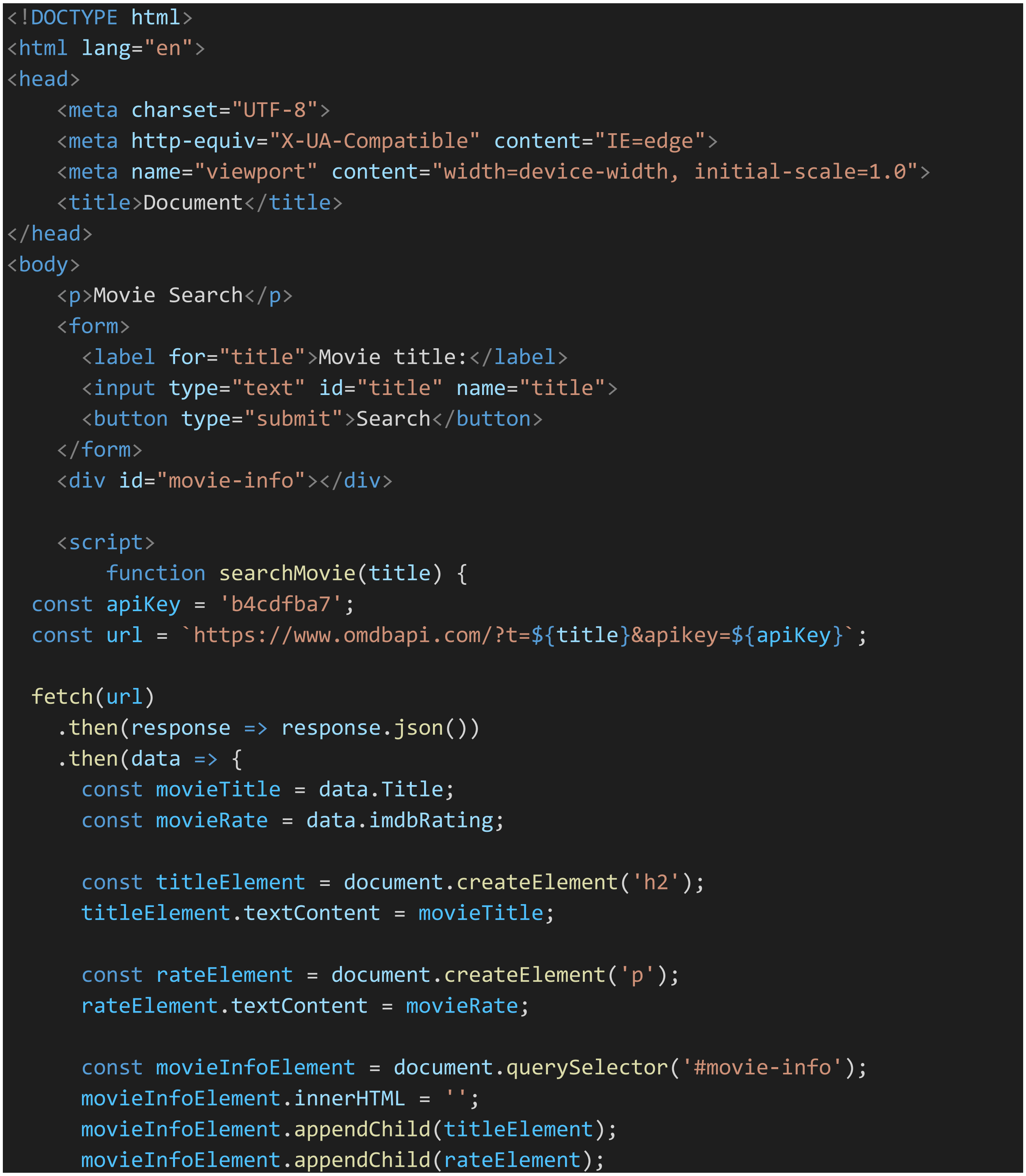
GitHub Username: jonas

- jonas/acs-engine
- jonas/adminrouter-public
- jonas/akka
- jonas/akka-blog
- jonas/akka-cluster-management
- jonas/akka-http
- jonas/akka.github.com
- jonas/alloy-ui
- jonas/alpakka
- jonas/amadou
- jonas/Android-ViewPagerIndicator
- jonas/appdirect-storelocator
- jonas/argo
- jonas/asciidoctor-stylesheet-factory
- jonas/aurum-datadiscovery
- jonas/base64
- jonas/blog.priorarts.org
- jonas/charts
- jonas/circe
- jonas/circe-config
- jonas/circe-yaml
- jonas/conductr-doc
- jonas/config
- jonas/contour
- jonas/coursier
- jonas/cpython
- jonas/cucumber-jvm
- jonas/Cucumber-JVM-Ruby-Example
- jonas/dcos-docs
- jonas/dgrep

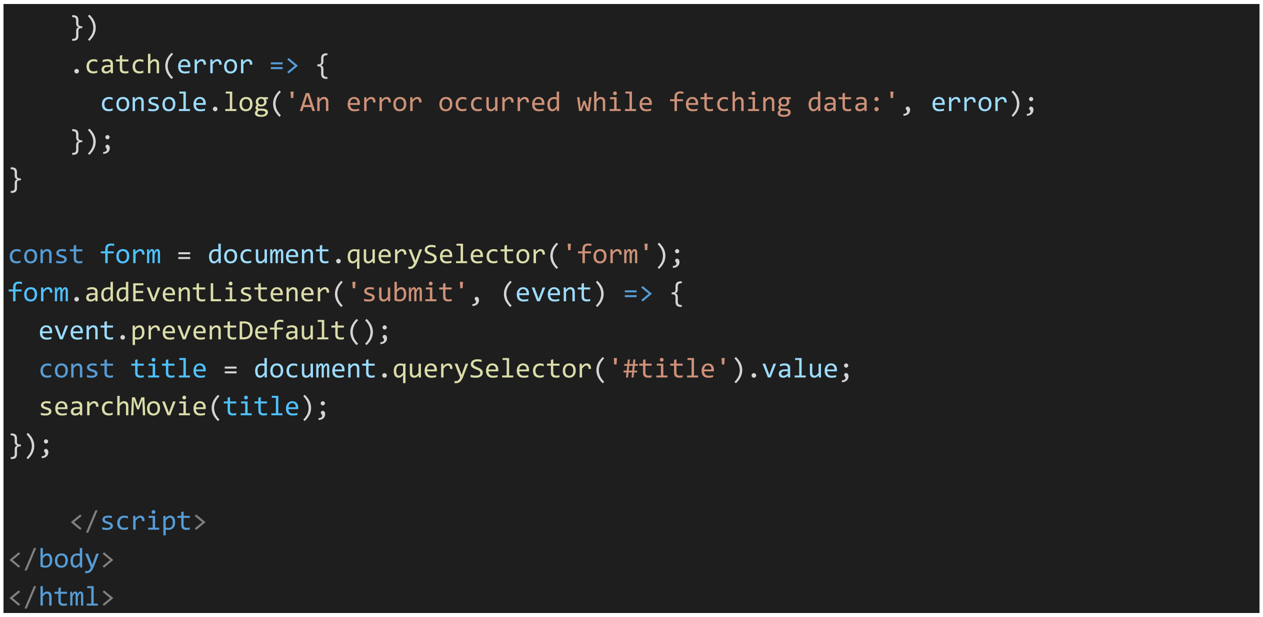
2. JSON movie api

create a web application that displays information about a specific movie. You can use the OMDB API to retrieve the movie data in JSON format. Create a JavaScript function that uses async/await to retrieve the movie data for a given title and displays it on the page

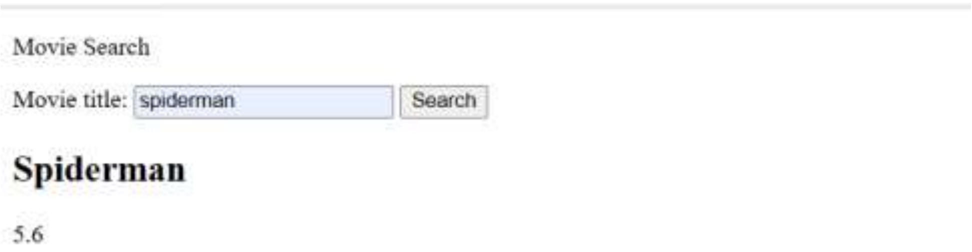
(https://www.omdbapi.com/?t=${title}&apikey=${apiKey})



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <p>Movie Search</p>
    <form>
      <label for="title">Movie title:</label>
      <input type="text" id="title" name="title">
      <button type="submit">Search</button>
    </form>
    <div id="movie-info"></div>

    <script>
        function searchMovie(title) {
  const apiKey = 'b4cdfba7';
  const url = `https://www.omdbapi.com/?t=${title}&apikey=${apiKey}`;

  fetch(url)
    .then(response => response.json())
    .then(data => {
      const movieTitle = data.Title;
      const movieRate = data.imdbRating;

      const titleElement = document.createElement('h2');
      titleElement.textContent = movieTitle;

      const rateElement = document.createElement('p');
      rateElement.textContent = movieRate;

      const movieInfoElement = document.querySelector('#movie-info');
      movieInfoElement.innerHTML = '';
      movieInfoElement.appendChild(titleElement);
      movieInfoElement.appendChild(rateElement);
```



```
    })
    .catch(error => {
      console.log('An error occurred while fetching data:', error);
    });
}

const form = document.querySelector('form');
form.addEventListener('submit', (event) => {
  event.preventDefault();
  const title = document.querySelector('#title').value;
  searchMovie(title);
});

    </script>
</body>
</html>
```



Movie Search

Movie title: spiderman

Spiderman

5.6