

Movie Search Application

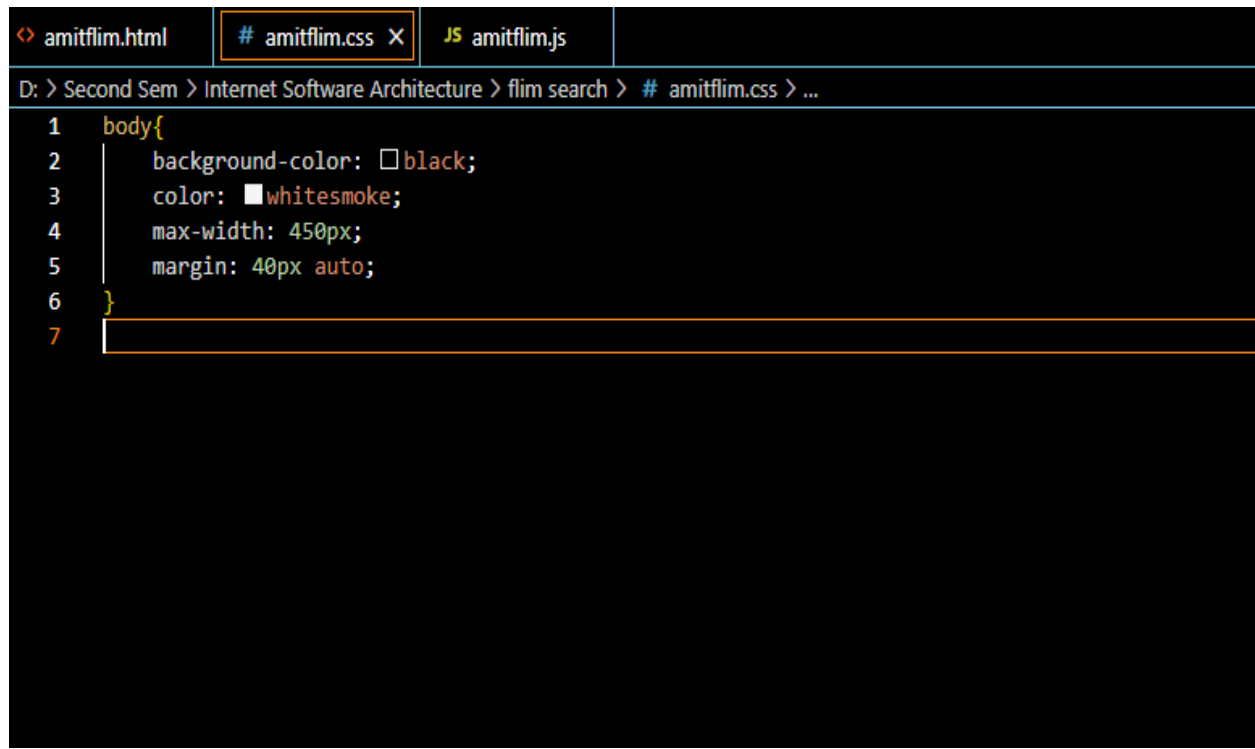
Name= Amit Babu Khatri

Student id= np03cs4s230291@heraldcollege.edu.np

HTML CODE:

```
amitflim.html x # amitflim.css JS amitflim.js
D: > Second Sem > Internet Software Architecture > flim search > amitflim.html > html > body > script
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Movie</title>
8     <link rel="stylesheet" href="amitflim.css">
9   </head>
10  <body>
11    <h1 id="movieTitle">Title: <b>Loading...</b></h1>
12    <p id="director">Director: <b>Loading...</b></p>
13    <div id="image">
14      
15    </div>
16    <div class="api">
17      <button>Search</button>
18      <input type="text" placeholder="Enter a movie name">
19    </div>
20    <script src="amitflim.js"></script>
21  </body>
22 </html>
```

CSS CODE:



The image shows a code editor with three tabs at the top: 'amitflim.html', '# amitflim.css' (which is the active tab), and 'JS amitflim.js'. Below the tabs is a breadcrumb trail: 'D: > Second Sem > Internet Software Architecture > flim search > # amitflim.css > ...'. The main editing area contains the following CSS code:

```
1 body{
2     background-color: black;
3     color: whitesmoke;
4     max-width: 450px;
5     margin: 40px auto;
6 }
7
```

JAVA SCRIPT CODE:

```
amitflim.html # amitflim.css JS amitflim.js X
D: > Second Sem > Internet Software Architecture > flim search > JS amitflim.js > ...
1 document.addEventListener("DOMContentLoaded", () => {
2
3     const defaultTitle = "Evil Dead Rise";
4
5     const apikey = "b4cdfba7";
6
7     const defaultUrl = `https://www.omdbapi.com/?t=${defaultTitle}&apikey=${apikey}`;
8
9
10
11     const movieTitle = document.getElementById("movieTitle");
12
13     const director = document.getElementById("director");
14
15     const image = document.getElementById("image");
16
17
18
19     fetch(defaultUrl)
20
21     .then((response) => response.json())
22
23     .then((data) => {
24
25         console.log(data);
26
27         movieTitle.innerHTML = `Title: <b>${data.Title}</b>`;
28
29         director.innerHTML = `Director: <b>${data.Director}</b>`;
30
31         image.innerHTML = ``;
32
33     })
34 }
```

```
amitflim.html # amitflim.css JS amitflim.js X
D: > Second Sem > Internet Software Architecture > flim search > JS amitflim.js > ...
33 })
34
35 .catch((error) => {
36
37     console.log("Error:", error);
38
39     movieTitle.innerHTML = `Title: <b>${defaultTitle}</b>`;
40
41     director.innerHTML = "";
42
43     image.innerHTML = "";
44
45 });
46
47 });
48
49
50
51 const searchButton = document.querySelector("button");
52
53 const inputField = document.querySelector("input");
54
55 const movieTitle = document.getElementById("movieTitle");
56
57 const director = document.getElementById("director");
58
59 const image = document.getElementById("image");
60
61
62
63 searchButton.addEventListener("click", () => {
64
65     const title = inputField.value;
66
67     if (title.length < 1) {
68         console.log("Title is empty");
69         return;
70     }
71
72     const defaultTitle = "Evil Dead Rise";
73
74     const apikey = "b4cdfba7";
75
76     const defaultUrl = `https://www.omdbapi.com/?t=${title}&apikey=${apikey}`;
77
78     fetch(defaultUrl)
79
80     .then((response) => response.json())
81
82     .then((data) => {
83
84         console.log(data);
85
86         movieTitle.innerHTML = `Title: <b>${data.Title}</b>`;
87
88         director.innerHTML = `Director: <b>${data.Director}</b>`;
89
90         image.innerHTML = ``;
91
92     })
93
94     .catch((error) => {
95
96         console.log("Error:", error);
97
98         movieTitle.innerHTML = `Title: <b>${defaultTitle}</b>`;
99
100         director.innerHTML = "";
101
102         image.innerHTML = "";
103
104     });
105
106 });
107 }
```

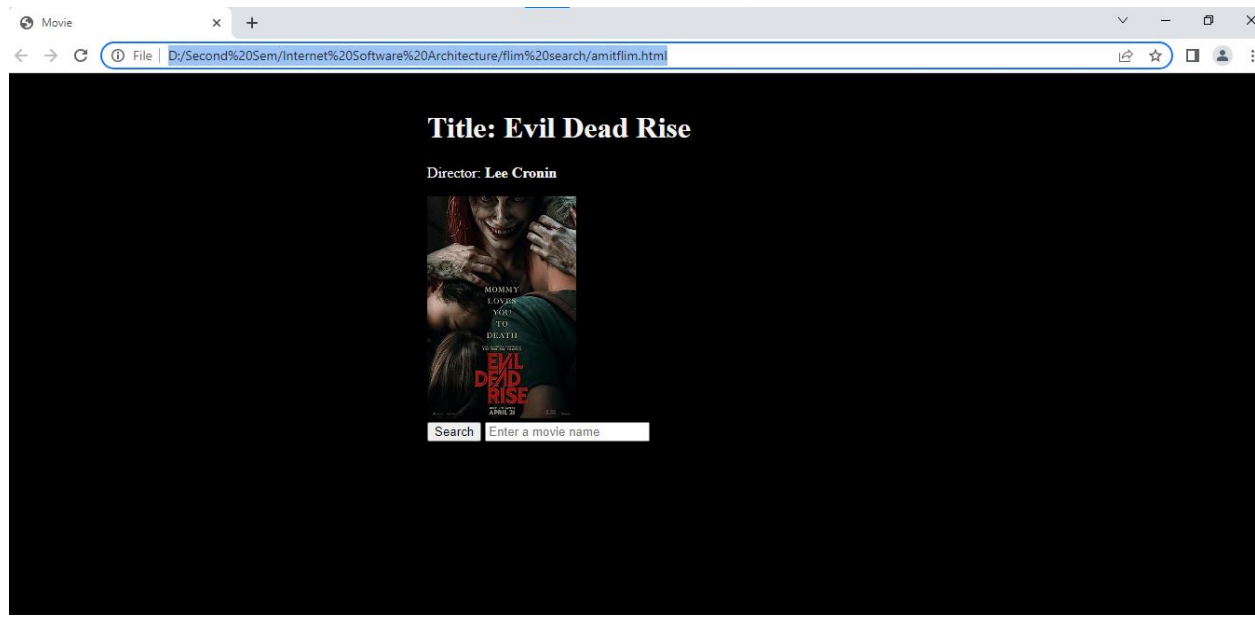
```
<> amitflim.html # amitflim.css JS amitflim.js X
D: > Second Sem > Internet Software Architecture > flim search > JS amitflim.js > ...

64
65     const title = inputField.value;
66
67     const apikey = "b4cdfba7";
68
69     const url = `https://www.omdbapi.com/?t=${title}&apikey=${apikey}`;
70
71
72
73     fetch(url)
74
75     .then((response) => response.json())
76
77     .then((data) => {
78
79         console.log(data);
80
81         movieTitle.innerHTML = `Title: <b>${data.Title}</b>`;
82
83         director.innerHTML = `Director: <b>${data.Director}</b>`;
84
85         image.innerHTML = ``;
86
87     })
88
89     .catch((error) => {
90
91         console.log("Error:", error);
92
93         movieTitle.innerHTML = `Title: <b>${title}</b>`;
94
95         director.innerHTML = "";
96     })
```

```
<> amitflim.html # amitflim.css JS amitflim.js X
D: > Second Sem > Internet Software Architecture > flim search > JS amitflim.js > ...

78
79     console.log(data);
80
81     movieTitle.innerHTML = `Title: <b>${data.Title}</b>`;
82
83     director.innerHTML = `Director: <b>${data.Director}</b>`;
84
85     image.innerHTML = ``;
86
87 })
88
89 .catch((error) => {
90
91     console.log("Error:", error);
92
93     movieTitle.innerHTML = `Title: <b>${title}</b>`;
94
95     director.innerHTML = "";
96
97     image.innerHTML = "";
98
99 });
100
101 });
102
103
```

RUNNING AND DEBUGGING OF CODE:



CONCLUSION:

Overall, this code sets up a basic implementation for movie search application using HTML, CSS, and JavaScript. User can search the movie by clicking the corresponding button. The code keep track to all the movie including image tittle and name of director.

From this code, we can learn about the following concepts:

- 1.HTML structure and elements.
- 2.CSS styling and responsive design.
- 3.JavaScript event handling and DOM manipulation.
- 4.Implementing game logic using JavaScript functions.
- 5.Using variables to store and update game data.
- 6.Reloadng the page using JavaScript.

