

# HTML

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Movie Manager</title>
6      <style>
7          body {
8              font-family: Arial, sans-serif;
9              max-width: 800px;
10             margin: 0 auto;
11             padding: 20px;
12         }
13         form {
14             background: #f4f4f4;
15             padding: 20px;
16             margin-bottom: 20px;
17             border-radius: 5px;
18         }
19         input {
20             margin: 5px;
21             padding: 8px;
22             width: 200px;
23         }
24         button {
25             padding: 8px 15px;
26             margin: 5px;
27             cursor: pointer;
28         }
29         .movie-item {
30             background: #fff;
31             border: 1px solid #ddd;
32             padding: 15px;
33             margin: 10px 0;
34             border-radius: 5px;
35         }
36         .movie-item button {
37             margin-right: 10px;
38         }
39         #search-input {
40             width: 100%;
41             padding: 10px;
42             margin-bottom: 20px;
43             box-sizing: border-box;
44         }
45     </style>
46 </head>
47 <body>
48     <h1>Movie Collection Manager</h1>
49
50     <form id="add-movie-form">
51         <h3>Add New Movie</h3>
52         <input type="text" id="title" placeholder="Title" required>
53         <input type="text" id="genre" placeholder="Genre">
54         <input type="number" id="year" placeholder="Release Year" required>
55         <button type="submit">Add Movie</button>
56     </form>
57
58     <hr>
59
60     <h2>Movie List</h2>
61     <input type="text" id="search-input" placeholder="Search movies by title or genre">
62
63     <div id="movie-list"></div>
64
65     <script src="movie-script.js"></script>
66 </body>
67 </html>

```

# JAVA SCRIPT

```
1  const API_URL = 'http://localhost:3000/movies';
2  const movieListDiv = document.getElementById('movie-list');
3  const searchInput = document.getElementById('search-input');
4  const form = document.getElementById('add-movie-form');
5
6  let allMovies = []; // Stores the full, unfiltered list of movies
7
8  // Function to dynamically render movies to the HTML
9  function renderMovies(moviesToDisplay) {
10     movieListDiv.innerHTML = '';
11
12     if (moviesToDisplay.length === 0) {
13         movieListDiv.innerHTML = '<p>No movies found matching your criteria.</p>';
14         return;
15     }
16
17     moviesToDisplay.forEach(movie => {
18         const movieElement = document.createElement('div');
19         movieElement.classList.add('movie-item');
20         movieElement.innerHTML = `
21             <p><strong>${movie.title}</strong> (${movie.year}) - ${movie.genre}</p>
22             <button onclick="editMoviePrompt(${movie.id}, '${movie.title}', ${movie.year}, '${movie.genre}')">Edit</button>
23             <button onclick="deleteMovie(${movie.id})">Delete</button>
24         `;
25         movieListDiv.appendChild(movieElement);
26     });
27 }
28
29 // Function to fetch all movies and store them (READ)
30 function fetchMovies() {
31     fetch(API_URL)
32         .then(response => response.json())
33         .then(movies => {
34             allMovies = movies; // Store the full list
35             renderMovies(allMovies); // Display the full list
36         })
37         .catch(error => console.error('Error fetching movies:', error));
38 }
39
40 // Search Functionality
41 searchInput.addEventListener('input', function() {
42     const searchTerm = searchInput.value.toLowerCase();
43     const filteredMovies = allMovies.filter(movie => {
44         return movie.title.toLowerCase().includes(searchTerm) ||
45             movie.genre.toLowerCase().includes(searchTerm);
46     });
47     renderMovies(filteredMovies);
48 });
49
50 // CREATE - Add a new movie (POST Method)
51 form.addEventListener('submit', function(event) {
52     event.preventDefault();
53
54     const newMovie = {
55         title: document.getElementById('title').value,
56         genre: document.getElementById('genre').value,
57         year: parseInt(document.getElementById('year').value)
58     };
59
60     fetch(API_URL, {
61         method: 'POST',
62         headers: {
63             'Content-Type': 'application/json'
64         },
65         body: JSON.stringify(newMovie)
66     })
67     .then(response => response.json())
68     .then(() => {
69         fetchMovies(); // Refresh the list
70         form.reset(); // Clear the form
71     })
72     .catch(error => console.error('Error adding movie:', error));
73 });
```

```

74
75 // UPDATE - Edit a movie (PUT Method)
76 function editMoviePrompt(id, currentTitle, currentYear, currentGenre) {
77     const newTitle = prompt('Enter new title:', currentTitle);
78     const newGenre = prompt('Enter new genre:', currentGenre);
79     const newYear = prompt('Enter new year:', currentYear);
80
81     if (newTitle && newYear) {
82         const updatedMovie = {
83             title: newTitle,
84             genre: newGenre || currentGenre,
85             year: parseInt(newYear)
86         };
87
88         fetch(`${API_URL}/${id}`, {
89             method: 'PUT',
90             headers: {
91                 'Content-Type': 'application/json'
92             },
93             body: JSON.stringify(updatedMovie)
94         })
95         .then(response => response.json())
96         .then(() => {
97             fetchMovies(); // Refresh the list
98         })
99         .catch(error => console.error('Error updating movie:', error));
100     }
101 }
102
103 // DELETE - Remove a movie (DELETE Method)
104 function deleteMovie(id) {
105     if (confirm('Are you sure you want to delete this movie?')) {
106         fetch(`${API_URL}/${id}`, {
107             method: 'DELETE'
108         })
109         .then(() => {
110             fetchMovies(); // Refresh the list
111         })
112         .catch(error => console.error('Error deleting movie:', error));
113     }
114 }
115
116 // Initial load
117 fetchMovies();

```

# JASON

```
1 {  
2   "movies": [  
3     { "id": 1, "title": "Inception", "genre": "Sci-Fi", "year": 2010 },  
4     { "id": 2, "title": "Pulp Fiction", "genre": "Crime", "year": 1994 },  
5     { "id": 3, "title": "Dune", "genre": "Sci-Fi", "year": 2021 }  
6   ]  
7 }
```