

Building a Movie Management Web Application with JSON Server Integration

HTMLCODE:

```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Movie Management</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <h1>Movie Management</h1>
11
12  <div id="movieList"></div>
13
14  <div id="movieForm">
15    <h2>Add Movie</h2>
16    <input type="text" id="titleInput" placeholder="Movie" />
17    <input type="text" id="genreInput" placeholder="Genre" />
18    <input type="text" id="directorInput" placeholder="Director" />
19    <input type="text" id="releaseYearInput" placeholder="Release Year" />
20    <button id="addMovieButton">Add</button>
21  </div>
22
23  <script src="script.js"></script>
24 </body>
25 </html>
26
27
28
```

CSS CODE:

```
<> index.html # style.css X
# style.css > ...
1  /* styles.css */
2
3  body {
4      font-family: Arial, sans-serif;
5      background-color: #f5f5f5;
6      margin: 0;
7      padding: 20px;
8  }
9  body{
10     background-image: url(open-clapper-board-with-film-strip-background-design_1017-26102.avif);
11 }
12
13 h1 {
14     text-align: center;
15 }
16
17 #movieList {
18     margin-top: 20px;
19 }
20
21 .movie-item {
22     background-color: #fff;
23     border: 1px solid #ccc;
24     border-radius: 5px;
25     padding: 10px;
26     margin-bottom: 10px;
27 }
28
29 .movie-item h2 {
30     margin: 0;
31 }
32
```

```
<> index.html # style.css X
# style.css > ...
32
33 .movie-item p {
34     margin: 5px 0;
35 }
36
37 #movieForm {
38     background-color: #fff;
39     border: 1px solid #ccc;
40     border-radius: 5px;
41     padding: 10px;
42     margin-top: 20px;
43 }
44
45 #movieForm h2 {
46     margin: 0 0 10px;
47 }
48
49 #movieForm input[type="text"],
50 #movieForm button {
51     display: block;
52     margin-bottom: 10px;
53     padding: 5px;
54 }
55
56 #addMovieButton {
57     background-color: #007bff;
58     color: #fff;
59     border: none;
60     padding: 8px 12px;
61     cursor: pointer;
62 }
63
```

<> index.html

style.css



style.css > ...

```
55  
56 #addMovieButton {  
57   background-color: #007bff;  
58   color: #fff;  
59   border: none;  
60   padding: 8px 12px;  
61   cursor: pointer;  
62 }  
63  
64 #addMovieButton:hover {  
65   background-color: #0056b3;  
66 }  
67  
68 button {  
69   background-color: #eee;  
70   border: none;  
71   padding: 5px 10px;  
72   cursor: pointer;  
73 }  
74  
75 button:hover {  
76   background-color: #ccc;  
77 }  
78
```

JAVA SCRIPT CODE:

```
<> index.html JS script.js X
JS script.js > [0] fetchMovies
1  const movieList = document.getElementById('movieList');
2  const movieForm = document.getElementById('movieForm');
3  const titleInput = document.getElementById('titleInput');
4  const genreInput = document.getElementById('genreInput');
5  const directorInput = document.getElementById('directorInput');
6  const releaseYearInput = document.getElementById('releaseYearInput');
7  const addMovieButton = document.getElementById('addMovieButton');
8
9  const API_URL = 'http://localhost:3000/movies';
10
11 // Function to fetch movies
12 const fetchMovies = async () => {
13   try {
14     const response = await fetch(API_URL);
15     const movies = await response.json();
16
17     displayMovies(movies);
18   } catch (error) {
19     console.log('Error:', error);
20     movieList.innerHTML = 'Error occurred while fetching data.';
21   }
22 };
23
24 // Function to display movies
25 const displayMovies = (movies) => {
26   movieList.innerHTML = '';
27
28   movies.forEach((movie) => {
29     const movieItem = createMovieItem(movie);
30     movieList.appendChild(movieItem);
31   });
32 };
33
```

```
// Function to create a movie item
const createMovieItem = (movie) => {
  const movieItem = document.createElement('div');
  movieItem.classList.add('movie-item');

  const movieTitle = document.createElement('h2');
  movieTitle.textContent = movie.title;
  movieItem.appendChild(movieTitle);

  const movieDetails = document.createElement('p');
  movieDetails.textContent = `Genre: ${movie.genre}, Director: ${movie.director}, Release Year: ${movie.releaseYear}`;
  movieItem.appendChild(movieDetails);

  const editButton = document.createElement('button');
  editButton.textContent = 'Edit';
  editButton.addEventListener('click', () => editMovie(movie.id));
  movieItem.appendChild(editButton);

  const deleteButton = document.createElement('button');
  deleteButton.textContent = 'Delete';
  deleteButton.addEventListener('click', () => deleteMovie(movie.id));
  movieItem.appendChild(deleteButton);

  return movieItem;
};
```

```
index.html JS script.js X
JS script.js > fetchMovies
59
60 // Function to add a new movie
61 const addMovie = async () => {
62   const movie = {
63     title: titleInput.value,
64     genre: genreInput.value,
65     director: directorInput.value,
66     releaseYear: releaseYearInput.value
67   };
68
69   try {
70     const response = await fetch(API_URL, {
71       method: 'POST',
72       headers: {
73         'Content-Type': 'application/json'
74       },
75       body: JSON.stringify(movie)
76     });
77
78     if (response.ok) {
79       const newMovie = await response.json();
80       const movieItem = createMovieItem(newMovie);
81       movieList.appendChild(movieItem);
82       clearFormInputs();
83     } else {
84       console.log('Error:', response.status);
85     }
86   } catch (error) {
87     console.log('Error:', error);
88   }
89 };
90
```

```
index.html JS script.js X
JS script.js > fetchMovies
88   }
89 };
90
91 // Function to edit a movie
92 const editMovie = async (movieId) => {
93   const updatedTitle = prompt('Enter new title:');
94   if (!updatedTitle) return;
95
96   try {
97     const response = await fetch(`${API_URL}/${movieId}`, {
98       method: 'PATCH',
99       headers: {
100         'Content-Type': 'application/json'
101       },
102       body: JSON.stringify({ title: updatedTitle })
103     });
104
105     if (response.ok) {
106       const updatedMovie = await response.json();
107       const movieItem = createMovieItem(updatedMovie);
108       const existingMovieItem = document.querySelector('.movie-item[data-id="${movieId}"]');
109       movieList.replaceChild(movieItem, existingMovieItem);
110     } else {
111       console.log('Error:', response.status);
112     }
113   } catch (error) {
114     console.log('Error:', error);
115   }
116 };
117
```

```
<> index.html | JS script.js | X
JS script.js > fetchMovies
117
118 // Function to delete a movie
119 const deleteMovie = async (movieId) => {
120   if (!confirm('Are you sure you want to delete this movie?')) return;
121
122   try {
123     const response = await fetch(`${API_URL}/${movieId}`, {
124       method: 'DELETE'
125     });
126
127     if (response.ok) {
128       const deletedMovie = document.querySelector(`.movie-item[data-id="${movieId}"]`);
129       movieList.removeChild(deletedMovie);
130     } else {
131       console.log('Error:', response.status);
132     }
133   } catch (error) {
134     console.log('Error:', error);
135   }
136 };
137
138 // Function to clear form inputs
139 const clearFormInputs = () => {
140   titleInput.value = '';
141   genreInput.value = '';
142   directorInput.value = '';
143   releaseYearInput.value = '';
144 };
145
146 // Event listeners
147 addMovieButton.addEventListener('click', addMovie);
148 window.addEventListener('DOMContentLoaded', fetchMovies);
```

JASON CODE:

```
movies.json > ...
1 {
2   "movies": [
3     {
4       "title": "SpiderMan",
5       "genre": "Action",
6       "director": "James",
7       "releaseYear": "2012",
8       "id": 2
9     },
10    {
11      "title": "wanted",
12      "genre": "action",
13      "director": "khatari amit",
14      "releaseYear": "2012",
15      "id": 6
16    },
17    {
18      "title": "CR7",
19      "genre": "football",
20      "director": "cristino ronaldo",
21      "releaseYear": "2022",
22      "id": 7
23    },
24    {
25      "title": "the girl",
26      "genre": "action",
27      "director": "ugtr",
28      "releaseYear": "2010",
29      "id": 8
30    }
31  ]
32 }
```

RUNNING AND DUBGGING OF CODE:

Movie Management

× +

← → ↺ file:///D:/Second Sem/Internet Software Architecture/movies/index.html

🔥 Getting Started 🌐 Failed to log in

Movie Management

SpiderMan

Genre: Action, Director: James, Release Year: 2012

Edit Delete

wanted

Genre: action, Director: khatri amit, Release Year: 2012

Edit Delete

CR7

Genre: football, Director: cristino ronaldo, Release Year: 2022

Edit Delete

the girl

Genre: action, Director: ugtr, Release Year: 2010

Edit Delete

Add Movie

Movie	Genre	Director	Release Year	Add
-------	-------	----------	--------------	-----