

```
getFavouriteMovie()
{
    return this.http.get<Movies[]>(`${this.apiUrl}/GetFavMovie`);
}

updateMovie(movie: UpdateMovie) {
    return this.http.put(`${this.apiUrl}/UpdateMovie/${movie.id}`, movie);
}

deleteMovie(movieId: number) {
    return this.http.delete(
        `http://localhost:5176/Movie/DeletebyId?movieId=${movieId}`, {
            responseType: 'text'
        }
    );
}
```

```
getFavMovie()
{
    this.moviesService.getFavouriteMovie().subscribe({
        next: (response: any[]) => {
            this.FavMovieess = response;
            console.log("Movies loaded:", this.FavMovieess);
        },
        error: (err) => {
            console.error("Error fetching movies", err);
        }
    });
}

onSubmit() {
    this.moviesService.updateMovie(this.FavMovieess).subscribe({
        next: (res) => {
            console.log('Update success:', res);
            alert('Movie updated successfully!');
        },
        error: (err) => {
```

```

        console.error('Update failed:', err);
        alert('Failed to update movie!');
    }
});
}

```

```

export interface UpdateMovie
{
    id:number,
    movieName:string,
    director:string,
    rating:number,
    collection:number
}

```

```

<div class="container mt-4">
  <h2 class="text-center mb-3">🎬 Favourite Movie</h2>

  <table class="table table-bordered table-striped text-center
align-middle">
    <thead class="table-dark">
      <tr>
        <th>ID</th>
        <th>Movie Name</th>
        <th>Director</th>
        <th>Rating</th>
        <th>Collection (Cr)</th>
      </tr>
    </thead>
    <tbody>
      <tr >
        <td>{{ FavMovie.id }}</td>
        <td>{{ FavMovie.movieName }}</td>
        <td>{{ FavMovie.director }}</td>
        <td>{{ FavMovie.rating }}</td>
        <td>{{ FavMovie.collection }}</td>

```

```

        </tr>
      </tbody>
    </table>
  </div>

<div class="container py-5 d-flex justify-content-center">
  <div class="card shadow-lg border-0 rounded-4 p-4 w-50">

    <h3 class="text-center fw-bold mb-4 text-primary"><img alt="clapperboard icon" data-bbox="675 255 695 270"/> Update Movie</h3>

    <form (ngSubmit)="onSubmit()" #movieForm="ngForm" novalidate>

      <div class="mb-3">
        <label for="id" class="form-label fw-semibold">Movie ID</label>
        <input type="number" id="id" class="form-control rounded-3
shadow-sm"
          [(ngModel)]="FavMovieiess.id" name="id" required>
      </div>

      <div class="mb-3">
        <label for="movieName" class="form-label fw-semibold">Movie
Name</label>
        <input type="text" id="movieName" class="form-control rounded-3
shadow-sm"
          [(ngModel)]="FavMovieiess.movieName" name="movieName"
required>
      </div>

      <div class="mb-3">
        <label for="director" class="form-label
fw-semibold">Director</label>
        <input type="text" id="director" class="form-control rounded-3
shadow-sm"
          [(ngModel)]="FavMovieiess.director" name="director" required>
      </div>

      <div class="mb-3">
        <label for="rating" class="form-label fw-semibold">Rating</label>
        <input type="number" id="rating" class="form-control rounded-3
shadow-sm"

```

```
                [(ngModel)]="FavMovieiess.rating" name="rating" required
min="0" max="10">
            </div>

            <div class="mb-3">
                <label for="collection" class="form-label
fw-semibold">Collection</label>
                <input type="number" id="collection" class="form-control rounded-3
shadow-sm"
                        [(ngModel)]="FavMovieiess.collection" name="collection"
required>
            </div>

            <button type="submit"
                    class="btn btn-primary w-100 rounded-3 fw-bold shadow-sm
py-2">
                🚀 Update Movie
            </button>
        </form>
    </div>
</div>
```

Interceptor

```
import { Injectable } from '@angular/core';
import {
  HttpRequest,
  HttpHandler,
  HttpEvent,
  HttpInterceptor
} from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable()
export class Interceptor implements HttpInterceptor {

  constructor() {}

  intercept(request: HttpRequest<unknown>, next: HttpHandler):
    Observable<HttpEvent<unknown>> {
    const authToken = localStorage.getItem("authToken")

    request =
request.clone({headers:request.headers.set("Authorization","Bearer
"+authToken)})
    return next.handle(request);
  }
}
```



Update Movie

Movie ID

1

Movie Name

Leo

Director

Lokesh

Rating

5

Collection

800

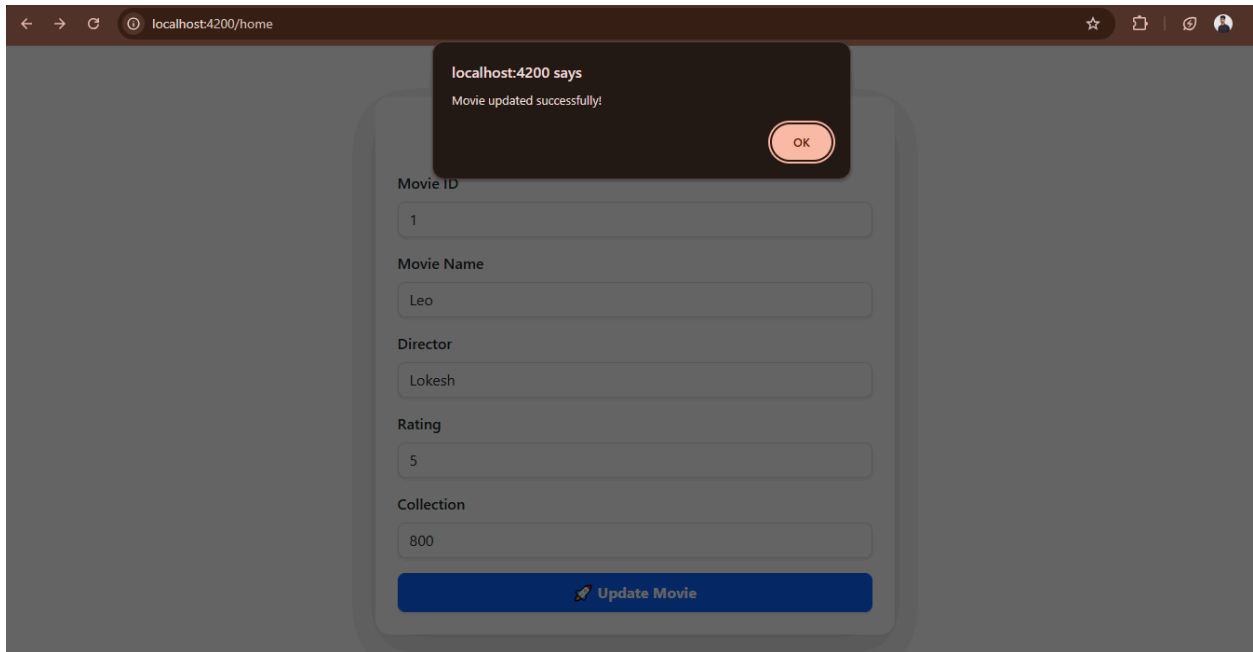


Update Movie



Favourite Movie


ID	Movie Name	Director	Rating	Collection (Cr)
1	Leo	Lokesh	5	800



 **Delete Movie**

Delete

After:

 **Delete Movie**

Delete

✔ **Movie deleted successfully!** ✕