

**8. Create a Movie Review System using Angular. Users can login to home page and add new movie reviews, rate movies, and view existing reviews. Demonstrate routing, two-way binding, services, form validation, and component communication.**

**/src/app/login/login.component.html**

```
button[type="submit"]:disabled {
  background-color: #a0c9f8;
  cursor: not-allowed;
}
<div class="login-container">
  
  <div class="logo">Rotten Potatoes</div>
  <div class="desc">
    Rotten Potatoes is the world's most trusted recommendation resources for
    quality entertainment. As the leading online aggregator of movie and TV
    show
    reviews from critics, we provide fans with a comprehensive guide to what's
    Fresh – and what's Rotten – in theaters and at home. If you're an
    entertainment fan looking for a recommendation, or to share an opinion,
    you've come to the right place.
  </div>
  <div class="login-form">
    <div class="form-group">
      <label for="uname">Username</label>
      <input
        type="text"
        name="uname"
        class="form-control"
        [(ngModel)]="username"
      />
    </div>
    <div class="form-group">
      <label for="pword">Password</label>
      <input
        type="password"
        name="pword"
        class="form-control"
        [(ngModel)]="password"
      />
    </div>
    <div class="btn-container">
      <div class="error">{{ errorMsg }}</div>
      <button class="btn btn-dark" (click)="login()">Login</button>
    </div>
  </div>
</div>
```

## /src/app/login/login.component.ts

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from '../services/auth.service';

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent implements OnInit {

  username = "";
  password = "";
  errorMsg = "";

  constructor(private auth: AuthService, private router: Router) { }

  ngOnInit(): void {
  }

  login() {
    if (this.username.trim().length === 0) {
      this.errorMsg = "Username is required";
    } else if (this.password.trim().length === 0) {
      this.errorMsg = "Password is required";
    } else {
      this.errorMsg = "";
      let res = this.auth.login(this.username, this.password);
      if (res === 200) {
        this.router.navigate(['home']);
      }
      if (res === 403) {
        this.errorMsg = "Invalid Credentials";
      }
    }
  }
}
```

## /src/app/movie/movie.component.html

```
<app-header></app-header>
<div class="movie-container">
  <div class="top">
    <div class="left">
```

write

```

<div class="name">{{ movie?.name }}</div>
<div class="rating">
  <app-star-rating
    [rating]="movie?.rating"
    [isReadOnly]="true"
  ></app-star-rating>
  <span class="rating-value">{{ movie?.rating }}</span>
</div>
<div class="cover">
  <img [src]="movie?.cover" alt="" />
</div>
</div>
<div class="right">
  <div class="title">Rate this movie</div>
  <div class="rating-form">
    <div class="form-group">
      <label for="uname">Name</label>
      <input type="text" name="uname" class="form-control" />
    </div>
    <div class="form-group">
      <label for="rating">Rating</label>
      <div class="rating">
        <app-star-rating
          [rating]="0"
          [isReadOnly]="false"
        ></app-star-rating>
      </div>
    </div>
    <div class="form-group">
      <label for="review">Review</label>
      <textarea name="review" rows="3" class="form-control"></textarea>
    </div>
    <div class="btn-container">
      <button class="btn btn-dark">Submit</button>
    </div>
  </div>
</div>
</div>
<div class="bottom">
  <h4 class="title">Reviews</h4>
  <div class="reviews" *ngIf="movie?.reviews?.length > 0">
    <div class="review" *ngFor="let review of movie?.reviews">
      <div class="review-details">
        <div class="review-left">
          <div class="author">{{ review.author }}</div>
          <div class="author_dt">
            {{ review.published_on | date: "dd-MMM-YYYY" }}
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

    </div>
    <div class="review-right">
      <div class="rating">
        <app-star-rating
          [rating]="review.rating"
          [isReadOnly]="true"
        ></app-star-rating>
        <span class="rating-value">{{ review.rating }}</span>
      </div>
    </div>
  </div>
  <div class="review-text"></div>
</div>
<div class="reviews" *ngIf="movie?.reviews?.length === 0">
  <div class="no-reviews">No reviews yet</div>
</div>
</div>

```

*fill here*

**/src/app/services/auth.service.ts**

```

import { Injectable } from '@angular/core';
import { Router } from '@angular/router';

@Injectable({
  providedIn: 'root'
})
export class AuthService {

  constructor(private router: Router) { }

  login(username: string, password: string) {
    if (username === 'suren' && password === '1234') {
      return 200;
    } else {
      return 403;
    }
  }

  logout() {
    this.router.navigate(['login']);
  }
}

```

**src/app/feature/star-rating/star-rating.component.html**

```

<ng-template #t let-fill="fill">
  <span class="star" [class.full]="fill === 100">

```

```

    <span class="half" [style.width.%]="fill">&hearts;</span>
    >&hearts;
  </span>
</ng-template>

<ngb-rating
  [(rate)]="rating"
  [starTemplate]="t"
  [readonly]="isReadOnly"
  [max]="5"
></ngb-rating>


```



ROTTEN POTATOES
Logout

## Free Guy

★★★★★ 4.15



Rate this movie

Name

Rating
★★★★★

Review

Submit

Reviews

No reviews yet