

Project Explanation

App.js Page:

This code is for a simple Node.js application using Express and EJS. It sets up a server that allows users to submit movie names and ratings (from 1 to 10) through a form. The app stores the movie names and ratings in an array and displays them on a separate page. If the user enters invalid data, the form page shows an error message. The application also allows users to delete movies from the list. It uses EJS for rendering views and includes a static file setup for public assets.

```
 1 import express from "express";
2 import path from "path";
3 import { fileURLToPath, pathToFileURL } from "url";
4
5 const app = express();
6 const PORT = 3000;
7
8 /// dirname paths
9 const __filename = fileURLToPath(import.meta.url);
10 const __dirname = path.dirname(__filename);
11
12 /// movies array
13 const movies = [];
14
15 /// EJS setting
16 app.set("view engine", "ejs");
17 app.set("views", path.join(__dirname, "views"));
18
19 /// parse of form data
20 app.use(express.urlencoded({ extended: true }));
21
22 /// static files
23 app.use(express.static(path.join(__dirname, "public")));
24
25 /// form
26 app.get("/", (req, res) => {
27   res.render("form", { error: null });
28 });
29
```

```
 1 // form handling
2 app.post("/submit", (req, res) => {
3   const { movieName, rating } = req.body;
4
5   if (!movieName || isNaN(rating) || rating < 1 || rating > 10) {
6     res.render("form", {
7       error: "Please enter a valid movie name and rating (1-10).",
8       movieName,
9       rating,
10    });
11  } else {
12    movies.push({ movieName, rating });
13    res.render("success", { movieName, rating });
14  }
15 });
16 /// movie route
17 app.get("/movies", (req, res) => {
18   res.render("movies", { movies });
19 });
20
21 /// Deleting movie route
22 app.post("/delete/:index", (req, res) => {
23   const { index } = req.params;
24   movies.splice(index, 1);
25   res.redirect("/movies");
26 });
27
28 ///running server
29 app.listen(PORT, () => {
30   console.log(`server is running at http://localhost:${PORT}`);
31 });
32
```

Form.ejs Page:

This is an HTML form where users can submit their favorite movie and its rating (from 1 to 10). If there's an error (such as missing or invalid information), an error message will be displayed at the top of the form. The form includes fields for the movie name and rating, and upon submission, the data is sent via a POST request to the "/submit" route in the backend. It also links to a CSS file for styling.

```
● ○ ●
1 <div class="container">
2     <h1>Submit your favorite movie</h1>
3     <% if (error) { %>
4         <div class="error-message">
5             <p><%= error %></p>
6         </div>
7     <% } %>
8
9     <form class="form-container" action="/submit" method="POST">
10        <label for="movieName">Movie Name:</label>
11        <input type="text" id="movieName" name="movieName" />
12
13        <label for="rating">Rating (1-10):</label>
14        <input type="number" id="rating" name="rating" />
15
16        <button type="submit">Submit</button>
17    </form>
18</div>
```

Success.ejs Page:

This HTML page is displayed after a user successfully submits a movie and its rating. It shows a success message with the movie name and rating. The user is given two options: they can either submit another movie or view all previously submitted movies. The page is styled using an external CSS file linked at the top.

```
● ○ ●
1 <div class="container">
2     <h1 class="success-title">✓ Submission Successful!</h1>
3     <p class="success-msg">You submitted:</p>
4     <ul class="movie-details">
5         <li><strong>Movie:</strong> <%= movieName %></li>
6         <li><strong>Rating:</strong> <%= rating %>/10</li>
7     </ul>
8
9     <div class="btn-group">
10        <a class="btn" href="/">🎬 Submit Another Movie</a>
11        <a class="btn secondary" href="/movies">📋 See All Submitted Movies</a>
12    </div>
13</div>
```

Movies.ejs Page:

This HTML page displays all the movies that have been submitted, showing each movie's name and rating. If no movies have been submitted yet, a message saying "No movies submitted yet" is shown. For each movie, there is a delete button that allows the user to remove it from the list. The page also includes a button for submitting another movie.

```
● ● ●
1 <div class="container">
2   <h1>All Submitted Movies</h1>
3   <% if (movies.length === 0) { %>
4     <p>No movies submitted yet.</p>
5   <% } else { %> <% movies.forEach((movie, index) => { %>
6     <div class="movie-card">
7       <li>
8         <strong><%= movie.movieName %></strong> – Rating: <%= movie.rating %>
9       </li>
10      <form action="/delete/<%= index %>" method="POST">
11        <button type="submit" class="delete-btn">Delete</button>
12      </form>
13    </div>
14  <% }) %> <% } %>
15  <div class="btn-group">
16    <a class="btn" href="/">Submit another movie</a>
17  </div>
18 </div>
```