**PREFACE**

Following are the items I have worked upon.

I have formatted the items, based on the existing issues that were asked to be worked upon, the fix or fixes I have used .

I have tried to add screenshots that exhibit the changes from user point of view. I have added technical explanations on fix details and approaches I have taken.

I have attempted all the TODOs including bugs, requests etc except Technical debt.

W.R.T the technical debt, I do not have experience in Structure map and the building blocks around it. I do understand Structure map enables DI/IOC. I have awareness and working knowledge of a similar tool-ABP(Asp Boiler Plate), if that helps.

BUG #1

**Bug #1: Movies are not sortable**

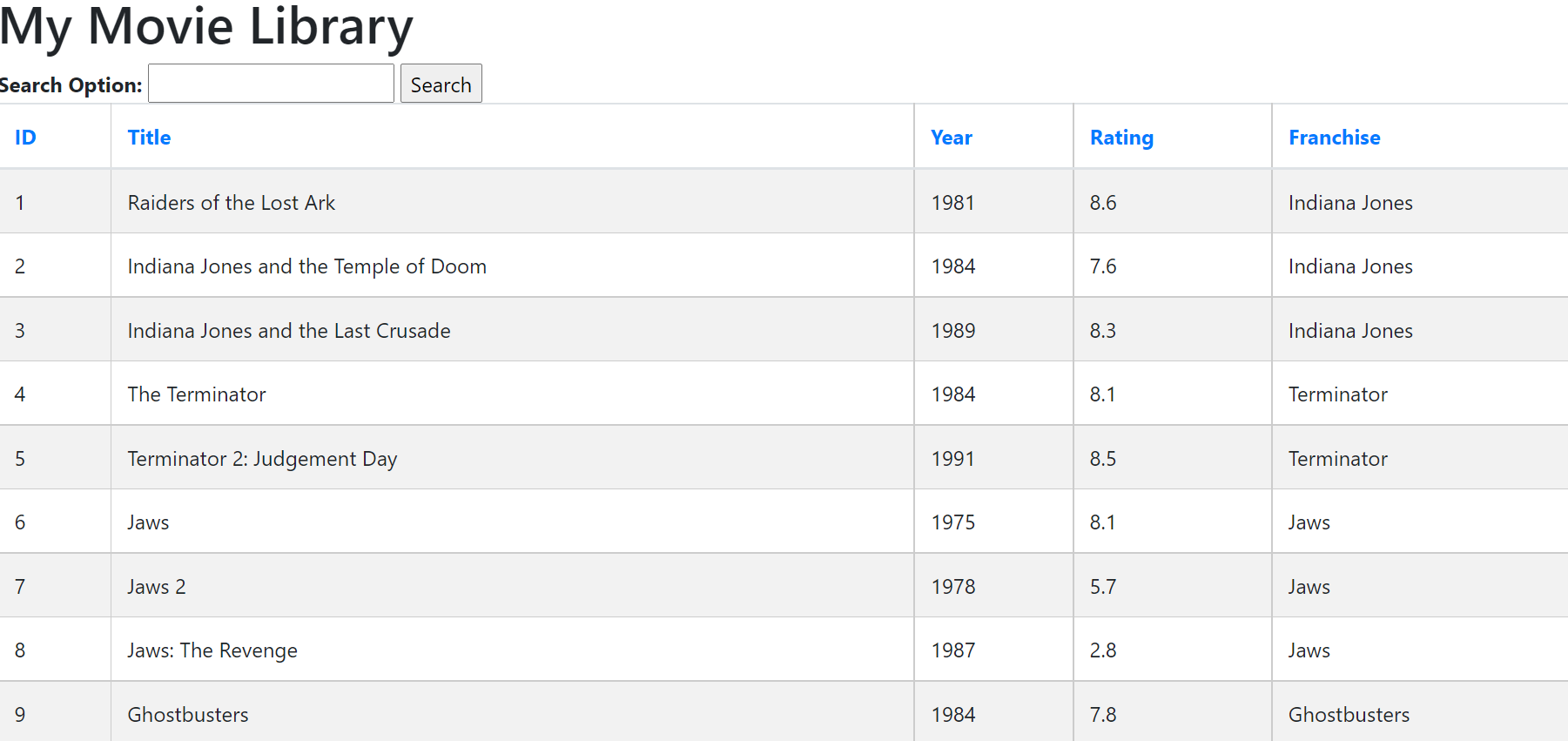
Currently, the movies are unable to be sorted. There are UI links in place to trigger a sort, but they do nothing. When complete, each link should be able to sort the data by their respective columns.

Some of the titles have leading articles (a/an/the). When sorting by title, provide a mechanism that excludes the leading articles, so that only the portion of the title following the article is matched on when sorting. For example, when the movie title "The Terminator" is included in the sortable list, only the title portion "Terminator" should be evaluated.

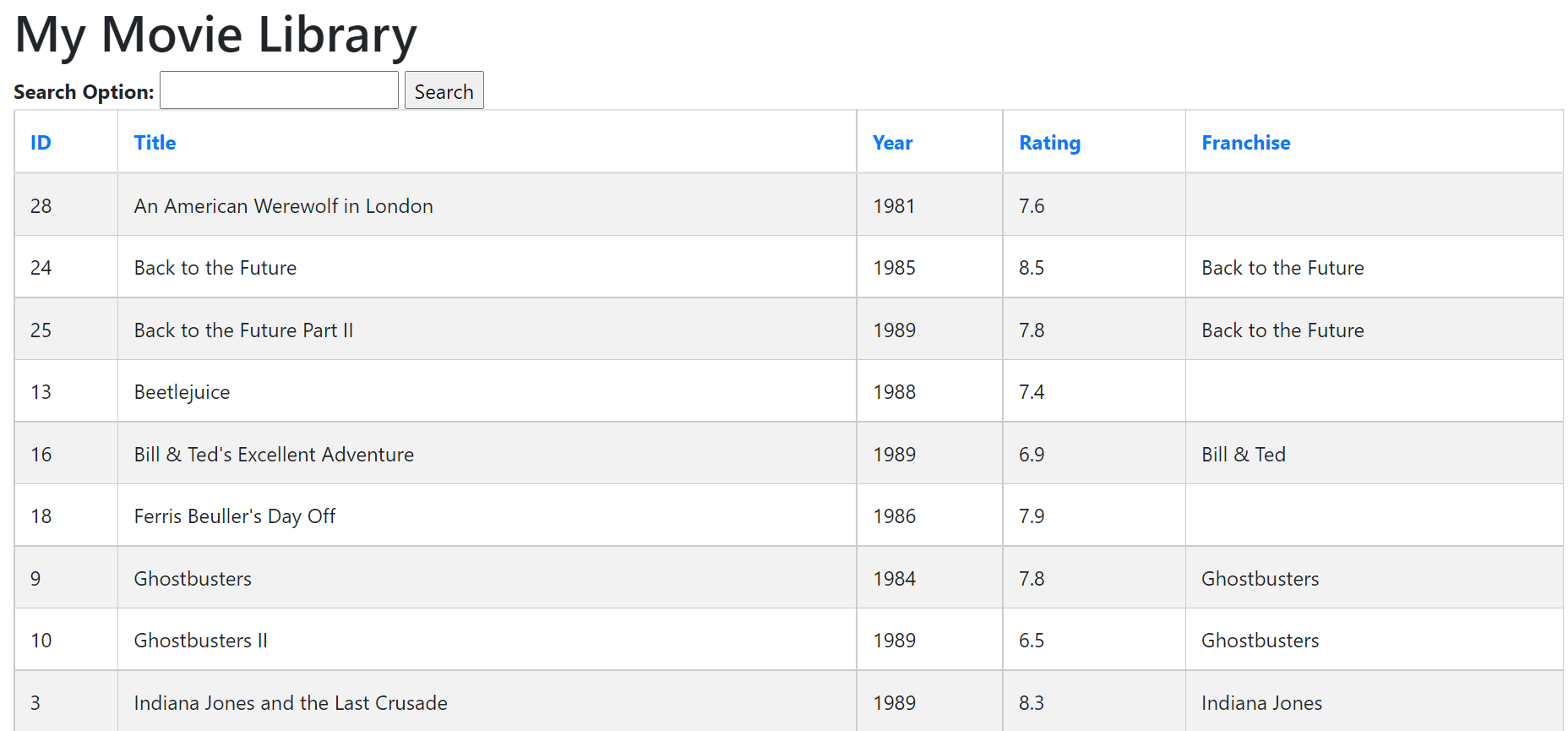
Note: There are also unit tests wired to the sort methods on the LibraryService which currently fail. When this is corrected, each unit test should pass.

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Root Cause | Fixes applied | Details |
| Movies cannot be sorted. | Upon investigation, it was seen that the sortColumn even though passed from Controller to Data Layer, is not getting used. Hence even though the HTML page has the sorting UI experience, no sorting is happening | LibraryService.Service.SearchMovies implementation has been changed to accommodate sortcolumn sent from UI. | Using reflection the sortcolumn has been determined, and the code has sorted based on the sort column. The amount of code has been reduced with this approach. The if/else based on sortcolumn have been reduced. |
| Leading articles like a/an/the should not evaluated for sorting | Default sorting was working with the whole string taken into consideration. Hence “The Terminator”’s “The” was used for sorting as well | Using Rgex , “a/an/the” has been factored in to strip the order by evaluation .”The Terminator”’s “The” has been taken off at the time of orderby.  The projection has been on the original string. | A screenshot has been added below for reference. |
|  |  |  |  |

Screenshot when page loads the first time



After hitting Sort on Title first time



After hitting on Sort second time



As seen above, even though “The Princess Bride” has a “The”, “The” has not been evaluated for sorting.

“P” of “Princess has been used to sort.

### Bug #2: Remove duplicate movies

Currently, there are movies in the library which are duplicated by Title. They show in the UI as a duplicate. Without physically removing the entries from the data store(library.xml), ensure that they no longer show in the UI.

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Root Cause | Fixes applied | Details |
| Duplicate movies are showing up in the list page | There is no functionality to remove duplicates from the list of movies, the library.xml has two entries for The NeverEnding Story  <Movie>  <ID>19</ID>  <Title>The NeverEnding Story</Title>  <Year>1984</Year>  <Rating>7.4</Rating>  <Franchise>**The NeverEnding Story**</Franchise>  <Date>1984-08-05</Date>  </Movie>  <Movie>  <ID>20</ID>  <Title>**The NeverEnding Story**</Title>  <Year>1984</Year>  <Rating>7.4</Rating>  <Date>1984-09-05</Date>  </Movie> | Distinct() clause has been added to remove the duplicate entry | Please refer screenshot below |

As can be seen below, duplicates have been removed



### Request #1: Filter movies by title

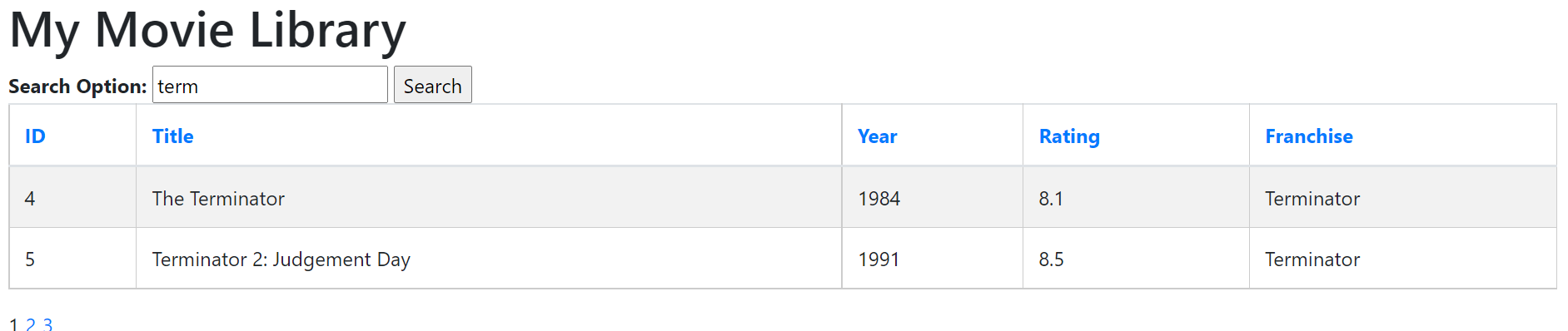
Currently, the Movie Library page shows all movies within the library. Implement functionality that allows the user to search movies by title.

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Root cause | Fixes applied | Details |
| User cannot search on Movie Title. | There was no input in the HTML where user can enter the Title and search on that. | Input has been added to HTML.  The DataAccess Library service has been changed to include the title in the filtering | Please refer screenshot below. |

Page upon loading



Upon Filtering on “Title”



### Request #2: Show movies by franchise

A number of movies in the library belong to a franchise (for instance, Star Wars IV, V, & VI all belong to the Star Wars franchise). Update the Movie Library page so that the franchises are shown. The franchises should also be sortable.

Note: The list of movies and what franchise they belong to are included in the UI Project under MovieFranchises.txt.

|  |  |  |  |
| --- | --- | --- | --- |
| Issue | Root Cause | Fixes applied | Details |
| Franchise doesn’t show up in the lovie list page | The library XML does not have the franchise details.  Also the models do not support Franchise | The Franchise details have been manually added from MovieFranchises.txt to library.xml as below  <Movie>  <ID>1</ID>  <Title>Raiders of the Lost Ark</Title>  <Year>1981</Year>  <Rating>8.6</Rating>  **<Franchise>Indiana Jones</Franchise>**  <Date>1981-06-05</Date>  </Movie>  Also a string property “Franchise” has been added to the movie model | Please refer screenshot below |

Movie List page with Franchise details



### Request #3: Implement a movie library API

Using whichever technology, pattern, and practice you feel is appropriate, implement a Movie Library API which may return any number of the following results:

1. Movies that are above or below a certain Rating
2. Movies between a date range
3. Movies belonging to a given franchise

A movieAPI has been built on .net core. It supports searches as requested.CORS has been evaluated so that the API can be called from the Angular APP.

**Example: Title search**

Postman has been used to test the API

1. Movies in the library, filtered by title

URL: https://localhost:5001/movie/

Request:

{

    "title": "Terminator",

    "ratingAbove": 0,

    "ratingBelow": 0,

    "franchise": "",

    "startDate":**null**,

    "endDate":**null**

}

Response:

[

    {

        "id": 4,

        "title": "The Terminator",

        "year": 1984,

        "rating": 8.1,

        "franchise": "Terminator",

        "date": "1984-01-05T00:00:00"

    },

    {

        "id": 5,

        "title": "Terminator 2: Judgement Day",

        "year": 1991,

        "rating": 8.5,

        "franchise": "Terminator",

        "date": "1991-02-02T00:00:00"

    }

]

**Example : Date Search**

Request

{

    "title": "",

    "ratingAbove": 0,

    "ratingBelow": 0,

    "franchise": "",

**"startDate":"1980-06-05",**

**"endDate":"1995-06-05"**

}

Response

[

    {

        "id": 1,

        "title": "Raiders of the Lost Ark",

        "year": 1981,

        "rating": 8.6,

        "franchise": "Indiana Jones",

        "date": "1981-06-05T00:00:00"

    },

    {

        "id": 2,

        "title": "Indiana Jones and the Temple of Doom",

        "year": 1984,

        "rating": 7.6,

        "franchise": "Indiana Jones",

        "date": "1984-01-05T00:00:00"

    },

    {

        "id": 3,

        "title": "Indiana Jones and the Last Crusade",

        "year": 1989,

        "rating": 8.3,

        "franchise": "Indiana Jones",

        "date": "1989-06-06T00:00:00"

    },

    {

        "id": 4,

        "title": "The Terminator",

        "year": 1984,

        "rating": 8.1,

        "franchise": "Terminator",

        "date": "1984-01-05T00:00:00"

    },

    {

        "id": 5,

        "title": "Terminator 2: Judgement Day",

        "year": 1991,

        "rating": 8.5,

        "franchise": "Terminator",

        "date": "1991-02-02T00:00:00"

    },

    {

        "id": 8,

        "title": "Jaws: The Revenge",

        "year": 1987,

        "rating": 2.8,

        "franchise": "Jaws",

        "date": "1987-05-01T00:00:00"

    },

    {

        "id": 9,

        "title": "Ghostbusters",

        "year": 1984,

        "rating": 7.8,

        "franchise": "Ghostbusters",

        "date": "1984-05-25T00:00:00"

    },

    {

        "id": 10,

        "title": "Ghostbusters II",

        "year": 1989,

        "rating": 6.5,

        "franchise": "Ghostbusters",

        "date": "1989-07-05T00:00:00"

    },

    {

        "id": 11,

        "title": "Labyrinth",

        "year": 1986,

        "rating": 7.4,

        "franchise": **null**,

        "date": "1986-07-30T00:00:00"

    },

    {

        "id": 12,

        "title": "Top Gun",

        "year": 1986,

        "rating": 6.8,

        "franchise": **null**,

        "date": "1985-03-05T00:00:00"

    },

    {

        "id": 13,

        "title": "Beetlejuice",

        "year": 1988,

        "rating": 7.4,

        "franchise": **null**,

        "date": "1988-07-01T00:00:00"

    },

    {

        "id": 14,

        "title": "The Karate Kid",

        "year": 1984,

        "rating": 7.2,

        "franchise": "The Karate Kid",

        "date": "1984-08-05T00:00:00"

    },

    {

        "id": 15,

        "title": "The Karate Kid, Part II",

        "year": 1986,

        "rating": 5.8,

        "franchise": "The Karate Kid",

        "date": "1986-08-30T00:00:00"

    },

    {

        "id": 17,

        "title": "The Princess Bride",

        "year": 1987,

        "rating": 8.2,

        "franchise": **null**,

        "date": "1987-06-20T00:00:00"

    },

    {

        "id": 18,

        "title": "Ferris Beuller's Day Off",

        "year": 1986,

        "rating": 7.9,

        "franchise": **null**,

        "date": "1986-07-05T00:00:00"

    },

    {

        "id": 19,

        "title": "The NeverEnding Story",

        "year": 1984,

        "rating": 7.4,

        "franchise": "The NeverEnding Story",

        "date": "1984-08-05T00:00:00"

    },

    {

        "id": 23,

        "title": "Star Wars Episode VI: Return of the Jedi",

        "year": 1983,

        "rating": 8.4,

        "franchise": "Star Wars",

        "date": "1983-01-05T00:00:00"

    },

    {

        "id": 24,

        "title": "Back to the Future",

        "year": 1985,

        "rating": 8.5,

        "franchise": "Back to the Future",

        "date": "1985-09-01T00:00:00"

    },

    {

        "id": 25,

        "title": "Back to the Future Part II",

        "year": 1989,

        "rating": 7.8,

        "franchise": "Back to the Future",

        "date": "1989-06-05T00:00:00"

    },

    {

        "id": 26,

        "title": "A Nightmare on Elm Street",

        "year": 1984,

        "rating": 7.5,

        "franchise": **null**,

        "date": "1984-07-05T00:00:00"

    },

    {

        "id": 27,

        "title": "A Nightmare on Elm Street 2: Freddy's Revenge",

        "year": 1985,

        "rating": 5.3,

        "franchise": "A Nightmare on Elm Street",

        "date": "1985-09-05T00:00:00"

    },

    {

        "id": 28,

        "title": "An American Werewolf in London",

        "year": 1981,

        "rating": 7.6,

        "franchise": **null**,

        "date": "1981-10-05T00:00:00"

    },

    {

        "id": 29,

        "title": "Jurassic Park",

        "year": 1993,

        "rating": 8,

        "franchise": "Jurassic Park",

        "date": "1983-12-05T00:00:00"

    }

]

**Example: Rating**

Request:

{

    "title": **null**,

**"ratingAbove": 8**,

    "ratingBelow": 0,

    "franchise": "",

    "startDate":**null**,

    "endDate":**null**

}

Response:

[

    {

        "id": 1,

        "title": "Raiders of the Lost Ark",

        "year": 1981,

        "rating": 8.6,

        "franchise": "Indiana Jones",

        "date": "1981-06-05T00:00:00"

    },

    {

        "id": 3,

        "title": "Indiana Jones and the Last Crusade",

        "year": 1989,

        "rating": 8.3,

        "franchise": "Indiana Jones",

        "date": "1989-06-06T00:00:00"

    },

    {

        "id": 4,

        "title": "The Terminator",

        "year": 1984,

        "rating": 8.1,

        "franchise": "Terminator",

        "date": "1984-01-05T00:00:00"

    },

    {

        "id": 5,

        "title": "Terminator 2: Judgement Day",

        "year": 1991,

        "rating": 8.5,

        "franchise": "Terminator",

        "date": "1991-02-02T00:00:00"

    },

    {

        "id": 6,

        "title": "Jaws",

        "year": 1975,

        "rating": 8.1,

        "franchise": "Jaws",

        "date": "1975-03-05T00:00:00"

    },

    {

        "id": 17,

        "title": "The Princess Bride",

        "year": 1987,

        "rating": 8.2,

        "franchise": **null**,

        "date": "1987-06-20T00:00:00"

    },

    {

        "id": 21,

        "title": "Star Wars Episode IV: A New Hope",

        "year": 1977,

        "rating": 8.7,

        "franchise": "Star Wars",

        "date": "1977-06-05T00:00:00"

    },

    {

        "id": 22,

        "title": "Star Wars Episode V: The Empire Strikes Back",

        "year": 1980,

        "rating": 8.8,

        "franchise": "Star Wars",

        "date": "1980-06-01T00:00:00"

    },

    {

        "id": 23,

        "title": "Star Wars Episode VI: Return of the Jedi",

        "year": 1983,

        "rating": 8.4,

        "franchise": "Star Wars",

        "date": "1983-01-05T00:00:00"

    },

    {

        "id": 24,

        "title": "Back to the Future",

        "year": 1985,

        "rating": 8.5,

        "franchise": "Back to the Future",

        "date": "1985-09-01T00:00:00"

    }

]

**Example: Franchise**

Request

{

    "title": **null**,

    "ratingAbove": 8,

    "ratingBelow": 0,

**"franchise": "Indiana Jones",**

    "startDate":**null**,

    "endDate":**null**

}

Response:

[

    {

        "id": 1,

        "title": "Raiders of the Lost Ark",

        "year": 1981,

        "rating": 8.6,

        "franchise": "Indiana Jones",

        "date": "1981-06-05T00:00:00"

    },

    {

        "id": 3,

        "title": "Indiana Jones and the Last Crusade",

        "year": 1989,

        "rating": 8.3,

        "franchise": "Indiana Jones",

        "date": "1989-06-06T00:00:00"

    }

]

### Request #4: Improve UI performance by implementing a JavaScript UI framework

Currently, the web application renders using MVC and Razor. Implement a basic UI framework of your choice (ReactJS, Angular, etc) so that the screens sort, filter, and page without reloading the entire screen.

Note: Bonus points if Request #1 (filtering by title) auto-refreshes as the user types.

An angular app has been built-Name: AngularMovieProject

Necessary services and components have been added to support the following

-Search on Title

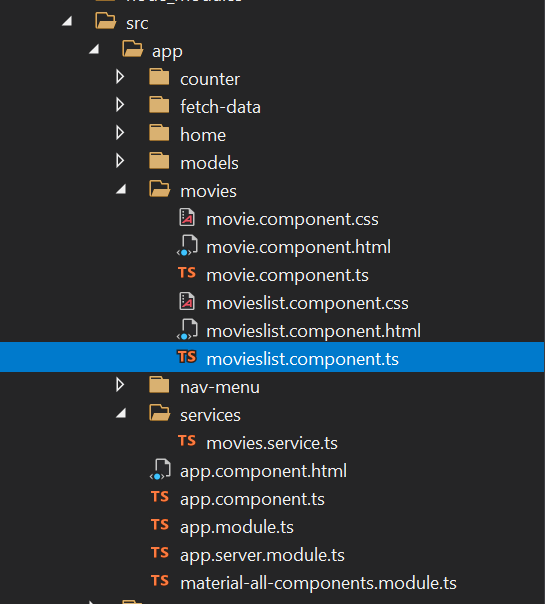
-Listing of movies

-Angular app calls the movieAPI from the services layer .The movie API needs to run when running the angular app.

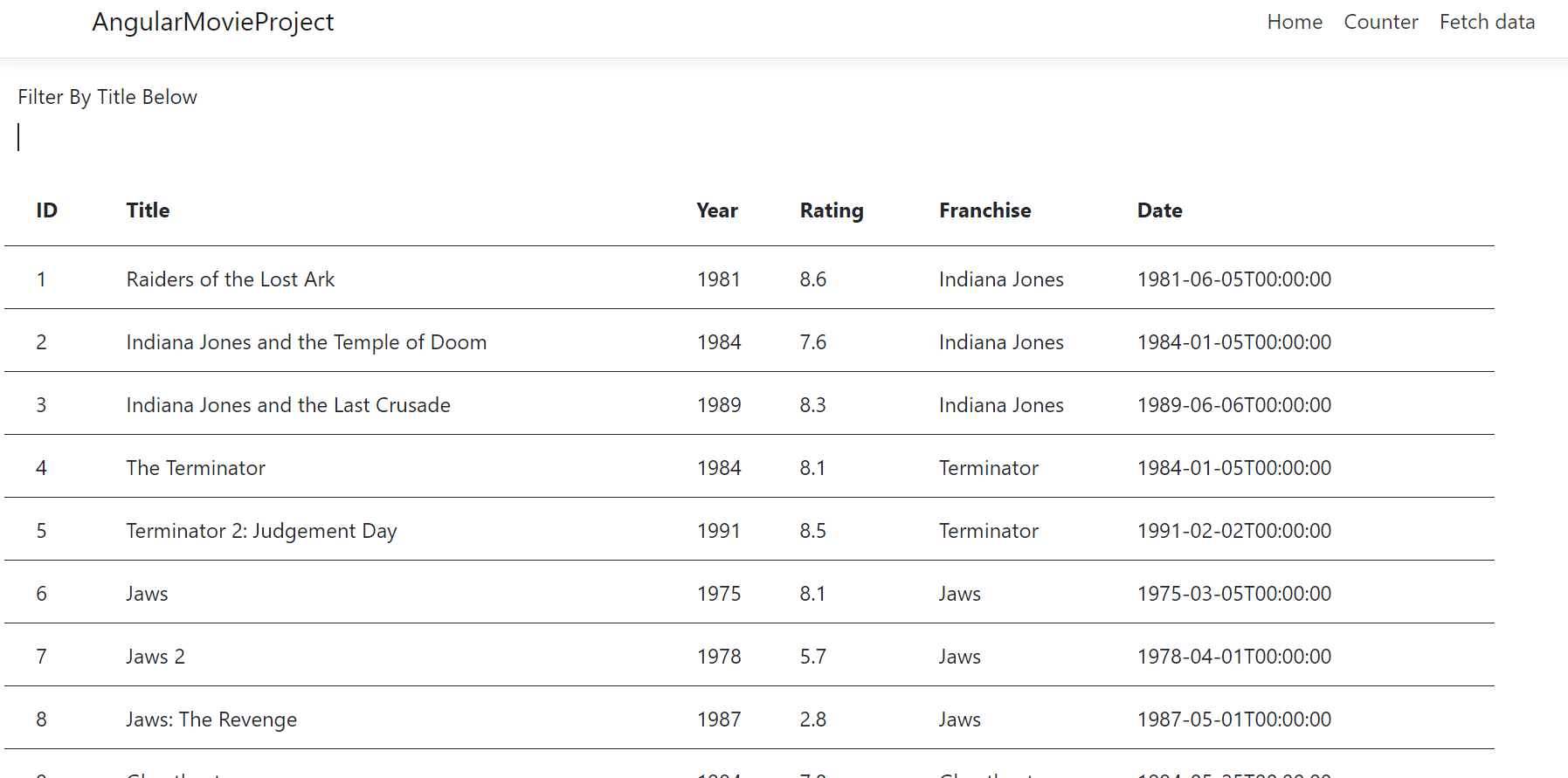
-Angular material has been used.

- Sorting, paging, filtering has been implemented at the UI layer. The page does not get refreshed.

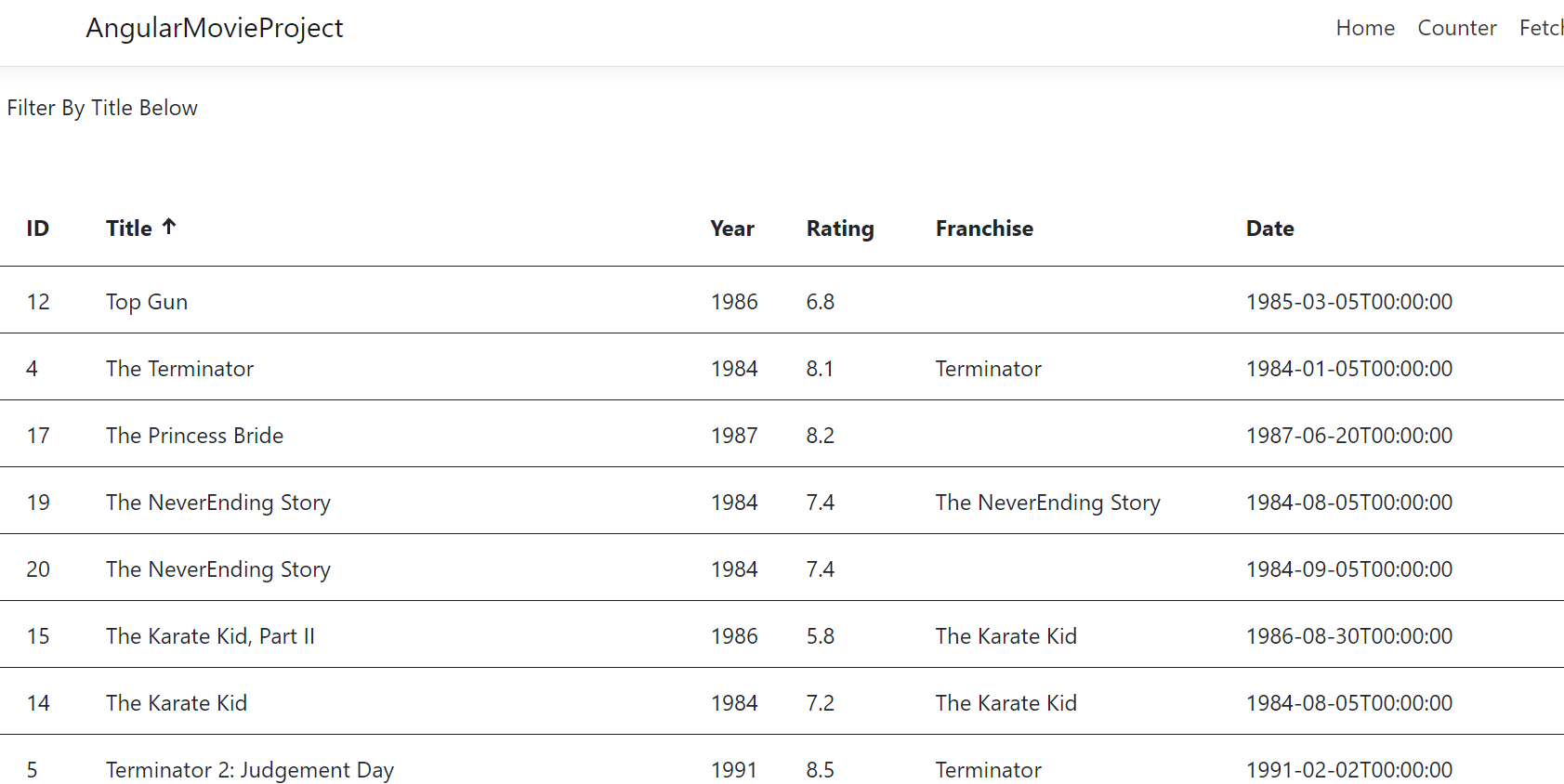
- The key files that implement the said functionality has been mentioned below for reference



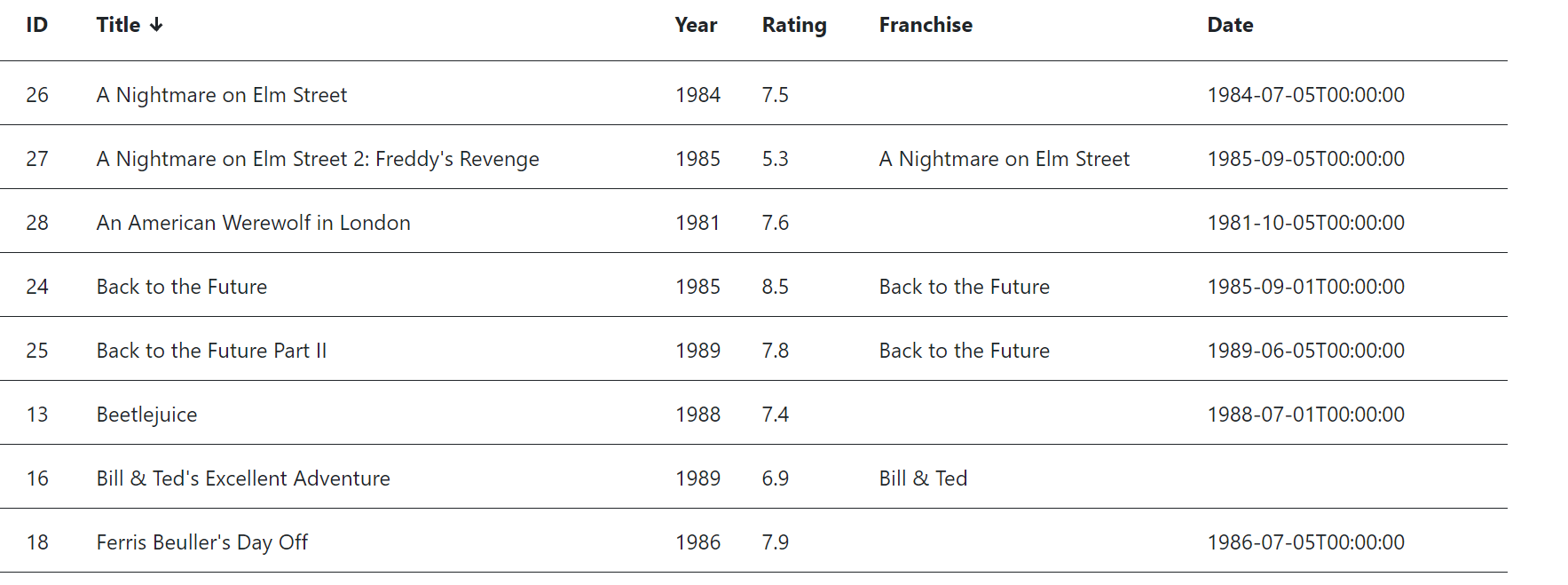
Screenshot upon loading



Sorting on Title-First Time:

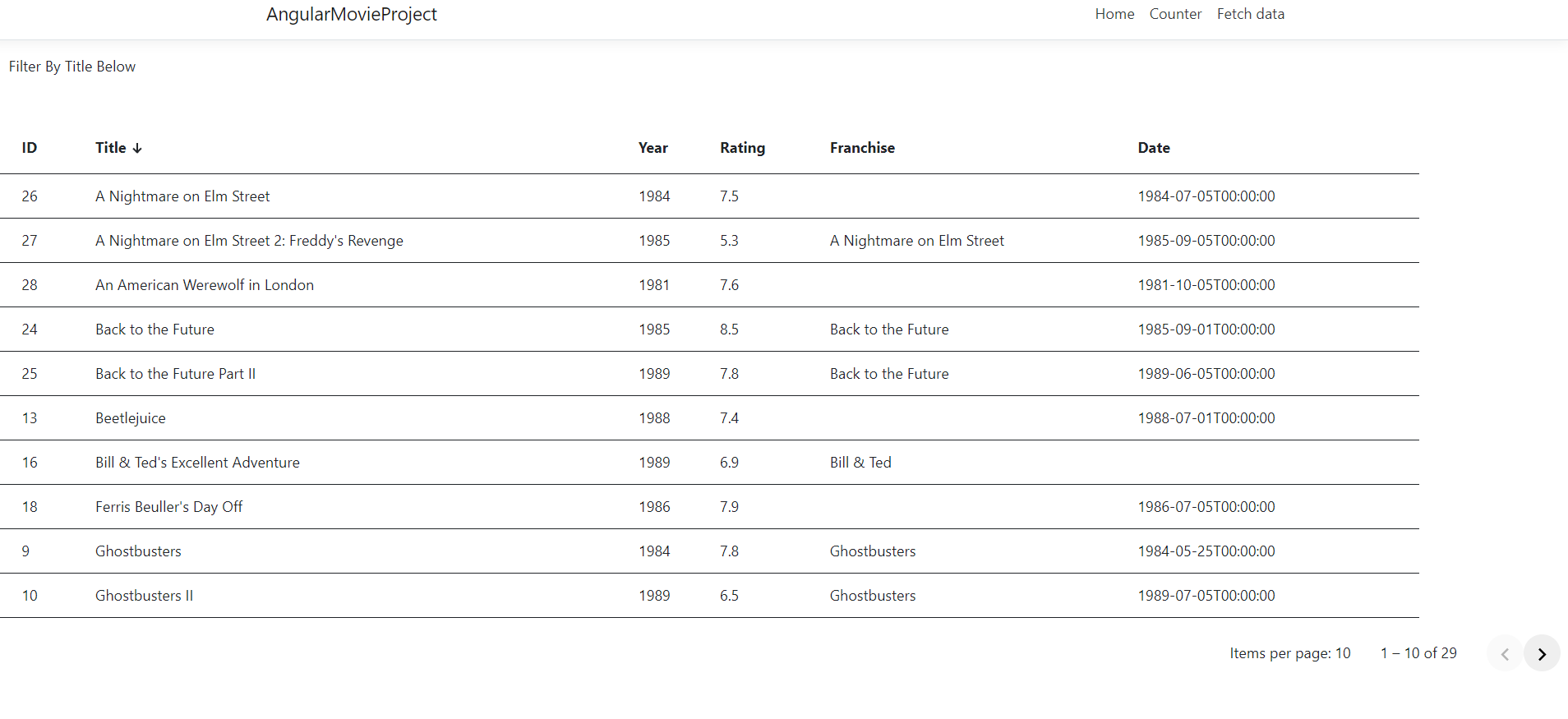


Sorting On Title-Second Time

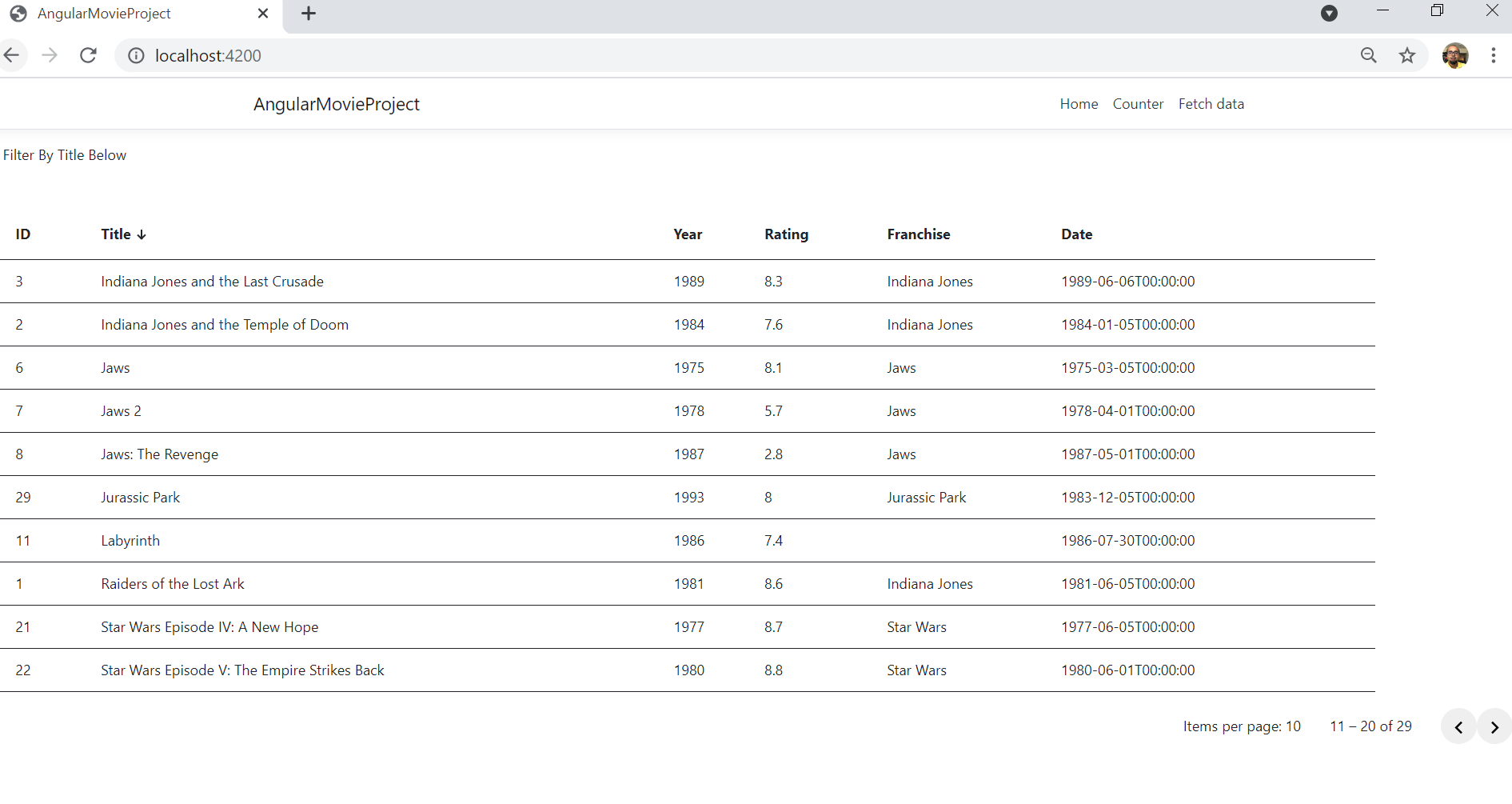


Paging

Upon loading



On paging



Filtering on Title:

