SE/COM S 3190 - Construction Of User Interfaces

Assignment 02

Total Points: 100

Published On: Mar 1, 2025 12:00 AM CST

Due Date: Mar 8, 2025 11:59 PM CST

This assignment is to be done as a team. Please take a close look at the rubric, there will be points allocated for having proper communication and collaboration. Please do this the right way.

If you need help, try office hours, the schedule is on Canvas. Lots of help is also available through the Piazza discussions. Please start the assignment as soon as possible to get your questions answered right away.

Table Of Contents

Prepared By: Pranava Sai Maganti

Overview

In this assignment, you as a team will develop a Movie Review Manager web application using HTML, CSS, and Javascript. The application will allow users to browse a paginated list of movies, submit reviews, and view the submitted reviews.

Please note that you need to work on this idea as it is and are not allowed to use your own concept, as the specifications are designed to fit this concept only and might not be suitable for another idea.

This assignment will require you to make at least three pages using HTML and use Javascript to dynamically update its content respectively.

You will use only HTML, CSS and Javascript and no frameworks like React, Angular or Vue. Bootstrap is allowed.

Design Requirements

The application should have three main views: Movie Listing, Review Submission and Review Summary.

1. Movie Listing View
   a. Display three pages of movies (preferably one language or genre per page) with at least 6 movies per page) with navigation controls like Previous/Next Buttons or NavBar
   b. Movie listings are to be loaded from a JSON file.
   c. Each movie should have a title, release year, genre, cast, director, image/poster and a short description.
   d. Each movie should have a *Write Review* button
   e. A search bar should allow filtering movies by title
   f. A View Reviews button should navigate to the review summary view

2. Review Submission View
    a. Users can select a movie from a dropdown menu
    b. A text area will allow users to write their review
    c. A rating system (1 - 5 stars) should be implemented using radio buttons or dropdown.
    d. A submit button will save the review and navigate back to the movie listing view
    e. A Cancel button will navigate back without saving any input added

3. Review Summary View
    a. Displays a list of submitted reviews, including movie title, review text, and rating
    b. Includes a Back to Movies button, which resets the review form and navigates back to the movie listing view.

What to Submit?

On Canvas, the assignment is created to accept submission as a team. So, submitting your Repository URL is the only submission you will have to submit on Canvas.

What to Put in GitLab?

As done for Assignment 01, we will again use GitLab as the submission platform where you will be making changes to the files and pushing them. Please note that the required number of pushes for this assignment is 2 pushes per person in the team. You will be given access to your team repositories on Mar 1, 2025 12:00 PM CST. The list of what to include is as follows:

1. *Source Code Files:*
    a. `index.html`: Contains the design for page 1 of movie listings
    b. `index2.html`: Contains the design for page 2 of movie listings
    c. `index3.html`: Contains the design for page 3 of movie listings
    d. `styles.css`: Contains custom styling for the entire web application.
    e. `script1.js`: Contains Javascript logic for page navigation, search, navigation, review handling and review submission and review summary for page 1 movies.
    f. `script2.js`: Contains Javascript logic for page navigation, search, navigation, review handling and review submission and review summary for page 2 movies.
    g. `script3.js`: Contains Javascript logic for page navigation, search, navigation, review handling and review submission and review summary for page 3 movies.
    h. `movies1.json`: Contains movie details for page 1 movie listings.
    i. `movies2.json`: Contains movie details for page 2 movie listings.
    j. `movies3.json`: Contains movie details for page 3 movie listings.

2. *Brief Video Explanation:*
    a. *Technical:* Explain how the application works, including JavaScript functionality, DOM manipulation, page navigation and event handling. For this part briefly explaining what code you have written and showing it will be accepted.
    b. *Procedural:* Demonstrate how users interact with the system. For this part, run your application and explain it while demonstrating each functionality.

Note: Please keep the length of the video to a maximum of 3 minutes and only mp4 format. Put this video into a folder called **Documents**.

3. *PDF Document:* Make a single page pdf that includes the information of individual contributions - like what Team Member A did and what Team Member B did, and how did you all collaborate and communicate.

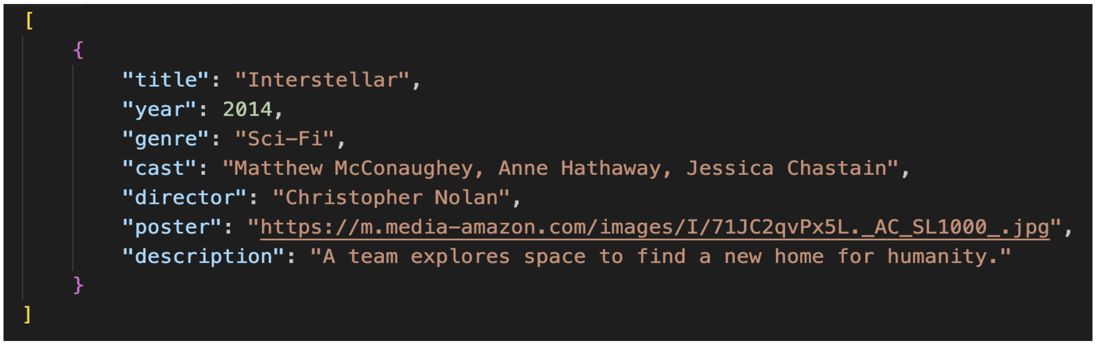
Grading Rubric

Item	Points
Communication & Collaboration	20
Movie Listings	45
Review Submission	15
Review Summary	10
Video Explanation	10
Total	100

Additional Notes

1. The JSON format for this assignment should follow a structured schema to ensure consistency across all teams. Each movie entry should include fields such as title, year, genre, cast, director, poster, and description, stored in an array within a JSON file, enabling seamless integration into the Movie Listing view.

   Required Format



```
[
    {
        "title": "Interstellar",
        "year": 2014,
        "genre": "Sci-Fi",
        "cast": "Matthew McConaughey, Anne Hathaway, Jessica Chastain",
        "director": "Christopher Nolan",
        "poster": "https://m.media-amazon.com/images/I/71JC2qvPx5L._AC_SL1000_.jpg",
        "description": "A team explores space to find a new home for humanity."
    }
]
```

2. Each team member must make at least two substantial pushes, where one push relates a feature implementation (such as adding movie listing functionality or review submission) and the other push includes a bug fix or enhancement (such as improving UI, fixing errors, or refining functionality). This policy ensures that all members actively contribute to the development process rather than just making minimal changes to fulfill the requirement.