**Movie Collection Project Summary**

**Introduction:**

An application to manage a personal movie collection, allowing users to add, view, edit, and delete movie entries. The UI includes a filterable and sortable list, a modal for adding/editing, and real-time search suggestions.

***Team Structure:***

***The entire class worked as a single team to deliver this project.***

***The earlier projects were done individually and delivered on Git-Hub.***

***Working as team and using tools to do the work brought a lot of learning. Working on a single understanding of the entire project, communicating & collaborating was challenging and lots of mistakes made. But this lead us to understand the real experience of a project and appreciate the various processes like common understanding or Customer Requirements and the others mentioned above.***

The Movie **Collection Project** was divided into **7 main modules**, with each module assigned to a **Module Lead** and supported by **two Engineers**. This structure ensured clear ownership, parallel development, and efficient integration.

**Modules & Responsibilities:**

|  |  |  |
| --- | --- | --- |
| **Module No** | **Module Name** | **Module Responsibility** |
| **01** | Add movie module | To create a responsive and working add movie form with client side and server-side validation and metadata should be stored in MySQL database  And message should be toast messages |
| **02** | Movie list module | Fetch the movie details from the DB and display the movies in home page in clickable cards with responsive design |
| **03** | Edit/Delete module | To create and edit form with client side and server-side validation and delete button that deletes the movie and messages has to be in toast message |
| **04** | Search module | A responsive search bar which will show results based on title or genre and upon entering characters it should show clickable keyboard navigable live suggestions list |
| **05** | Filter module | A module where the movies will be listed based on genre and genre should be listed in navigable clickable list |
| **06** | Sort module | A component using which user will be able to list movies based on ascending descending and alphabetical order |
| **07** | Rating module | A component to submit edit ratings of the movie |

**Features Implemented**

* **Created a responsive Add Movie Form with:**
  + Client-side and server-side validation.
  + Toast messages for success and error feedback.
  + Data storage in a MySQL database.
* **Fetched and displayed movies as responsive clickable cards on the home page.**
* **Developed Edit/Delete form:**
  + Edit with validation and update support.
  + Delete with confirmation dialog and toast messages.
* **Added a responsive Search Bar:**
  + Live suggestions based on any part of the movie title.
  + Keyboard-navigable dropdown list.
* **Implemented Genre-based navigation module:**
  + Clickable genre list to filter movies.
* **Developed Sorting component:**
  + Sort by title (ascending, descending, alphabetical, year).
* **Created a Rating component:**
  + Supports adding, editing, and resetting ratings**.**

**Issue Identified During Testing:**

* **Alphabetical sorting showed:**
  + Special characters and numbers (e.g., “12 Angry Men”) before alphabetic titles.
  + Required behavior : numeric titles should appear *after* A–Z.
* **Notes section overflowed on cards, breaking layout.**
* **Card heights were inconsistent, causing uneven grid alignment.**
* **Search feature limitations:**
  + Only matched beginning of titles.
* **Extra metadata fields like language and year were added mistakenly.**
* **Unapproved genres were added due to miscommunication.**
* **Text input was used for genre instead of a dropdown.**
* **Backend validation errors didn’t show messages or toast notifications.**
* **Toaster animations were missing for form actions.**

**Integration Challenges & Fixes**

* Function name duplication caused rendering and behavior issues.
* The module are does not supporting with the other modules so we reached the module leads and we assign the work together which is overlapping with other
* Port conflicts across developer environments led to errors.
* Inconsistent base URLs and paths created API/routing issues.
* Accidental data deletions during editing – resolved by:
  + Moving delete to a separate flow with confirmation.
* Default HTML validation replaced with custom JS validation.
* Missing feedback messages added for all API actions.
* No delete confirmation – added a confirmation prompt before deletion.

**CONCLUSION:**

The movie management system was successfully completed with full functionality, including adding, editing, deleting, rating, searching, sorting, and genre filtering. Initial issues like incorrect sorting, inconsistent UI, limited search logic, validation gaps, and feedback handling were identified during testing. After final integration, these were resolved through improved sorting rules, UI fixes, enhanced search functionality, proper form validation, and consistent toast notifications. The final product is stable, user-friendly, and meets all specified requirements.

**Following is the Folder structure, Stylings And Port Number :**

**standardised for the entire team:**

**Folder Structures :**

**Movie Collection**

**├── client/frontend**

**│ └── src**

**│ ├── pages**

**│ │ ├── HomePage.js**

**│ │ └── MovieDetails.js**

**│ ├── components**

**│ │ ├── AddMovieForm.js**

**│ │ ├── Search.js**

**│ │ ├── Filter.js**

**│ │ ├── Sort.js**

**│ │ ├── MovieList.js**

**│ │ ├── EditDeleteForm.js**

**│ │ ├── Ratings.js**

**│ │ └── NavBar.js**

**│ └── css**

**│ ├── ... .css**

**│ └── ... .css**

**├── server/backend**

**│ └── server.js**

**Stylings And Port Number :**

**Font-size :** Heading 32 px, Paragraph 18 px

**Font-Style :** Seoge UI ,sans-serif

**Bg-color and color :** bg-black ,color-white, buttons-red,grey.

**PORT NO : 5000**

**The final integration was far easier with the above standardisation.**

**Test case were built and executed, bugs were found, reported and fixed and update to the final version was created.**

**The test cases built were rendered on Bugzilla, after bugs were found they were logged into Bugzilla.**