Project Planning & Management (Technical Report)

Software Development Project 2

Submitted to: Shahram Jalaliniya

Section 005

Group number: Team 2

Group Members:

Rohan Juneja

Dinara Sharipova

Arielle Mueller

Nikita Melyanov

Vladislav Ostrikov

Geerthan Kanthasamy

Hai Truong Tran

Dinesh Balakrishnan

ADDED CHANGES: Iteration planning for Release 2, Test planning for Release 2, Conclusions and Recommendations

1.0 Abstract (5 pts.)

This technical report is focused on providing all the necessary details about our Movie Advisor Website that will be built by our group. All details listed here are relevant to the project and should be implemented during the development.

Sometimes people want something to burn time with, sometimes people want to spend that time watching a movie or two but have trouble finding what's in theatres near them. To solve this problem, our team came up with the development of a web service where moviegoers can see new movies as well as movies tailored to their preferences. Our website will help to centralize the actions of searching for movies, providing a preview and direct to a venue where it can be watched.

Our app will allow users to register with multiple levels of accounts based on user wants, even having a level specialized in providing reviews for other users to judge with.

2.0 Introduction (5 pts.)

The vast majority of movies are out there in the market, movie industries are becoming bigger and stronger from time to time. Therefore, it is quite overwhelming for a person to find a suitable movie in their little spare time at night, and obviously it is going to be relatively unenjoyable to them for how long they spend to find one movie they actually prefer. The team has come up with an idea that can solve the problem for everyone that enjoys watching movies.

Movie Advisor Project is created based on how people can quickly find a recommendation or find a good review of a movie. The project is also used by many movie critics which leads to all the reviews posted on this site are extremely reliable and well-written. People are obviously aware that there are several websites that review movies such as: IMDB rating, RottenTomato, IGN movie review, etc. However, every site does not contain enough features that a normal person needs when visiting a website which means every site lacks other features. Project is going to be based on the JavaScript language with a database that stores user accounts, movie archives.

Furthermore, Movie Advisor features 2 core purposes which are movie recommendation and movies review. The team only aims to make the project stay as a movie recommendation and review website, not as a streaming platform because there are already so many platforms that have been established. The project also has basically features: user registration for every account type, leaving comments-reviews, search for movies that the user prefers, movies recommended based on trending.

Finally, there will be an issue that could occur during the developing process, the issue is how the stored videos on the project’s database can run smoothly when it is viewed by the users.

3.0 Architecture: Development Perspective (10 pts.)

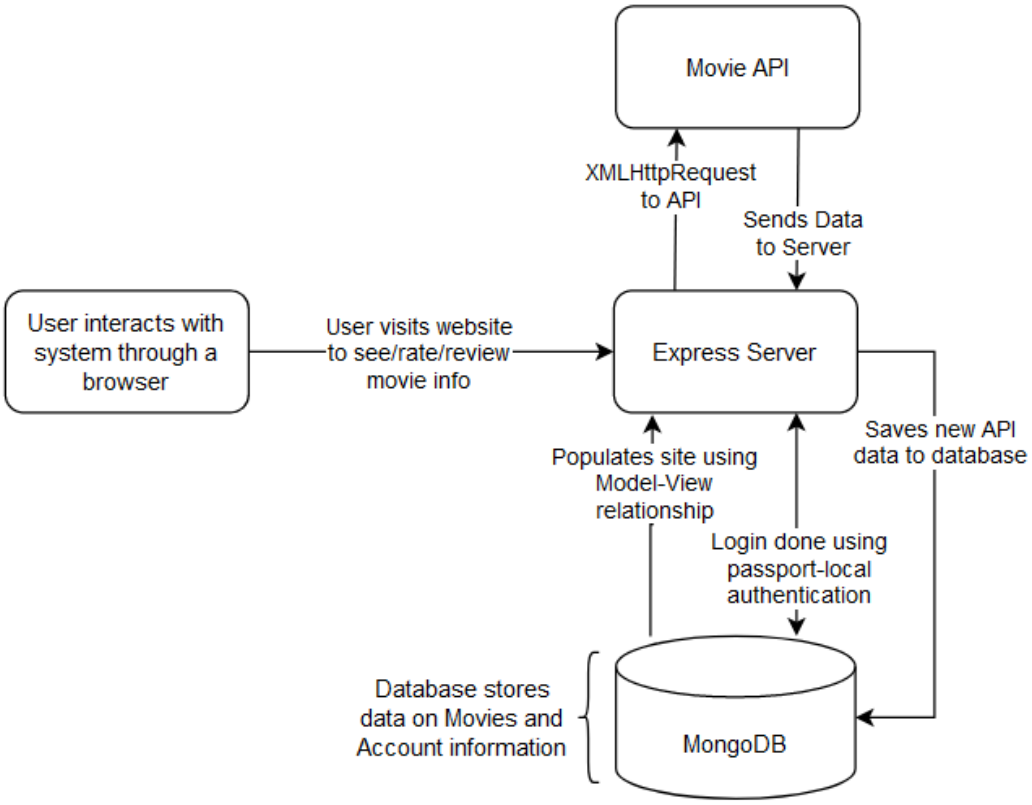


Figure 1: Movie Advisor Architecture Diagram

Supporting documentation for the Architecture Diagram:

Our project is built on the ExpressJS framework and is hosted on the web so it can be accessed through the browser by any user's computer, laptop and/or mobile device. From there the server periodically grabs data from an external movie API via an XMLHttpRequest and then sends the data to a Model tied to the View (using the MVC design pattern) which gets saved to our database that's built using MongoDB. The server then uses the MVC pattern to populate it's display View using the Model that uses the data in the database

The server also uses the MVC pattern for registering and logging in users and saving information to the database. Though security is implemented by using the passport module which saves a user's password as a hash upon registration and securely handles comparisons for logins via passport-local's authentication.

4.0 Conclusions & Recommendations (5 pts.)

Conclusions:

The main agenda of the project was the development of the web application based on Node.js, Express.js and MongoDB which allows a user to find various movies to watch, get to know information about those movies, watch the trailer and rate them.

After the finalization of 2 releases of the project the following functions were implemented: user registration and login/logout, homepage with the various movie sections, movie details page, a rating system of the movie, coming soon movies, top rated movies, basic search, advanced search with filter, administration panel.

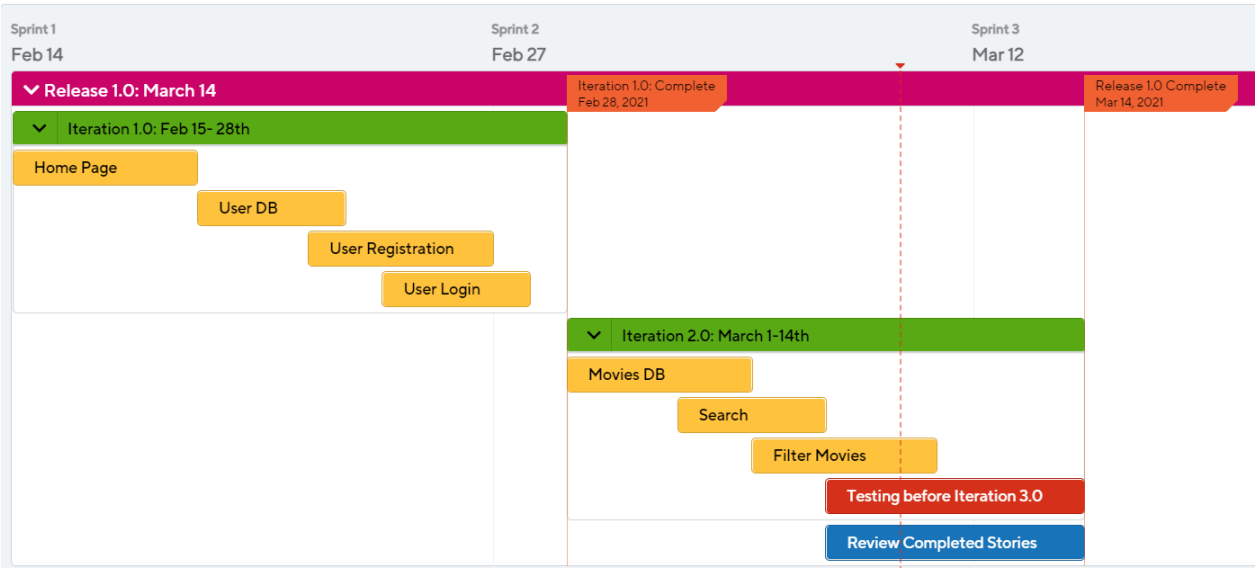
Recommendations:

As the recommendation for the future improvement of the project our team suggests continuing the work on AI recommendations feature that currently wasn't properly set up due to time restrictions. Moreover, we consider adding personal lists for the users to store their favorite movies in one place.

5.0 The Release Plan (20 pts.)

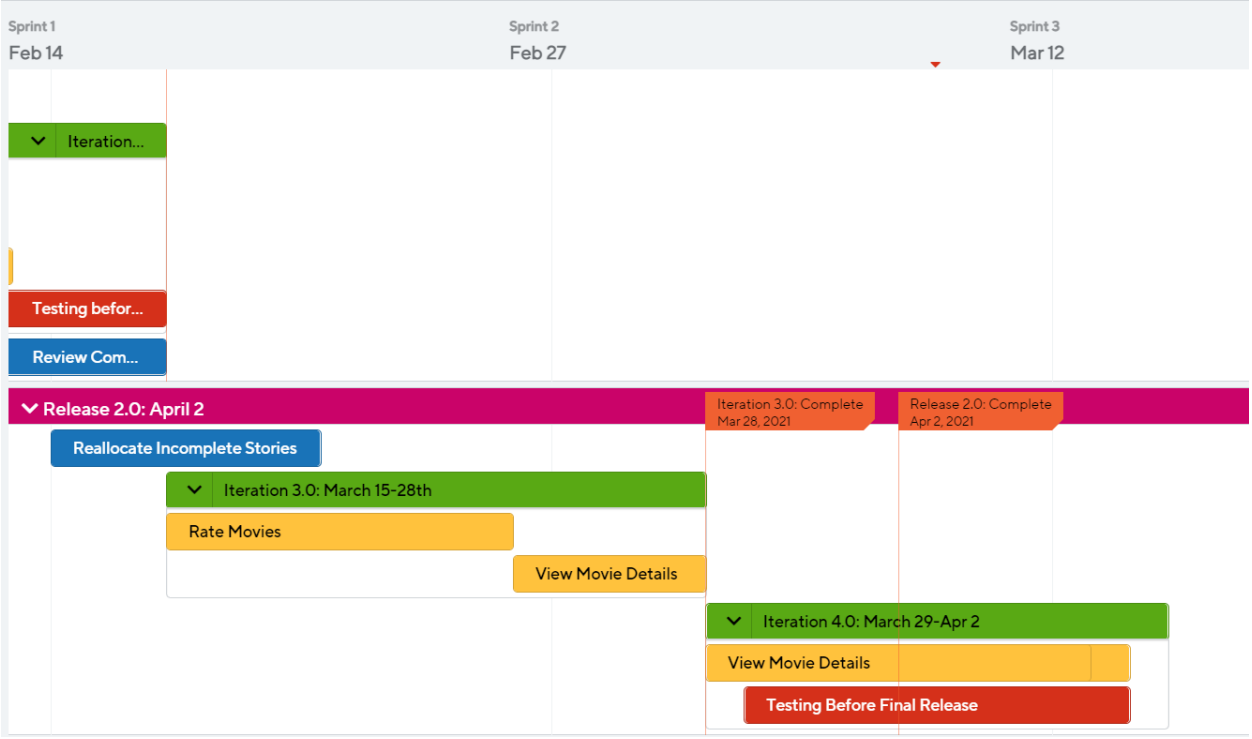
Release 1.0 March 14th:

- Iteration 1.0 - Feb 15th - 28th
- Iteration 2.0- March 1th - 14th
- March 13th: Testing before Release Date
- March 14th: Release 1.0 and Review Completed Stories



Release 2.0: April 2nd

- Iteration 3.0: March 15th - March 28th
- Iteration 4.0: March 29th - April 2nd
- April 1st: Testing Before Final Release
- Final Release (2.0): April 2nd



6.0 Iteration Planning (30 pts.)

Release 1.0:

Iteration 1.0 (Feb 15th - Feb 28th)

A User can Register for an account:

Task	Who	Estimate	Actual
User DB	Vlad	1 hour	0.5 hours
UI Page Registration	Dinara	2 hours	2 hours
Adding user to DB	Dinara + Vlad	1 hour	1 hour

A User can Login to their Account:

Task	Who	Estimate	Actual
Fetch User info from DB to ensure correct User is Logged in	Nikita	1.5 hours	1.5 hours
UI buttons to access Login/Registration Page with controllers and routing	Dinara	1.5 hours	1 hour
UI for Login Page	Dinara	1.5 hours	1 hour
Add UI for Logged In/Logged Out state	Dinara	2 hours	2.5 hours
Add logout: functionality + button for logging out	Dinara	1 hour	0.5 hours

A User can access the Home Page:

Task	Who	Estimate	Actual
UI for Homepage	Rohan	1 hour	2 hours
Mobile Responsive UI	Rohan	2 hours	2.5 hours
UI for Movie Cards on homepage	Rohan	4 hours	4 hours

Iteration 2.0 (March 1 - 14th)

A User can search for specific Movies:

Task	Who	Estimate	Actual
UI with Search Results	Rohan	1 hour	1 hour
Algorithm for showing Search Results	Nikita	3 hours	3 hours
Setting Up MovieDB	Geerthan + Arielle	1 hour	2 hours

Saving Movie Data to DB	Arielle	1 hour	3 hours
Fetching Movie Data from DB	Arielle	3 hours	5 hours
Coming Soon Movies	Hai	1 hour	2 hours
Top Rated Movies	Hai	1 hour	1 hour

A user can Filter when Searching for movies:

Task	**Who**	**Estimate**	**Actual**
Filtering results based on Users choice (Back-end)	Nikita	1 hour	1 hour
UI for displaying Filtered Search Results	Rohan	1 hour	1 hour
Search UI with Filter option	Rohan	1 hour	1 hour

Release 2.0:

Iteration 3.0 (March 14 - April 15th)

Searching and Filtering movies:

Task	**Who**	**Estimate**	**Actual**
Basic Search	Vlad	1 hour	1 hour
Advanced Search	Vlad	2 hours	2.5 hours
Search UI	Rohan	2 hours	2 hours

Movie Detail Page:

Task	**Who**	**Estimate**	**Actual**
Movie Details UI	Dinara + Rohan	1 hours	1 hour
Movie Detail Rating Front-End + Setup	Dinara	1 hours	2 hours

Movie rating implementation	Arielle	2 hours	4 hours

Admin Panel:

Task	Who	Estimate	Actual
Admin Panel UI	Rohan	3 hours	2 hours
Admin Panel: Movie Add	Arielle	1 hour	1 hour
Admin Panel: Movie list	Rohan	1 hour	1 hour

User Logout

Task	Who	Estimate	Actual
User logout: UI	Dinara	1 hour	1 hour
User Logout state + registration updated	Dinara	1 hour	1 hour

7.0 Test Plans (25 pts.)

Test plans and answers for Release 1.0 and Release 2.0

The requirements of the system's test plan include the following components:

1. Introduction
    1. Goals that summarize the testing goals for the project:

       ANSWER: System's tests are being done to ensure that the developing software meets all the criteria highlighted in the technical report and satisfies all the user stories. Those tests are focusing on checking if the delivered software allows the user to perform in a way that is described in user stories.

    2. Any assumptions made that may affect the understanding or execution of the plan:

       ANSWER: To avoid any assumptions that may affect the execution of those plans, the execution of all the tests is focusing on picking a feature of the

delivered software and testing if it is working and allowing a user to perform actions described in the user stories.

3. Describe elements of the software and hardware that are not part of the application but still may impact its correctness and must be checked. Describe the elements that might positively influence testing on the project.

   ANSWER: Internet Connection can influence the speed at which operations are done for the front-end and if processes can be done at all, Browser can affect a web system based on it's compatibility with the software, Web Server Host can affect the overall performance of the system.

2. Scope
   1. Describe the features and functions that will be tested during the project. This should include functional and non-functional requirements.

      ANSWER: Described in the tables "Functional" and "Non-functional testing" below

   2. Describe the features that will not be tested and reason why.
3. Testing Procedures
   1. Describe the testing procedures that project will use. This includes the test lifecycle, types of testing, test objectives, and test criteria.

      ANSWER: For this project we're conducting Unit Test Plan, Integrated Test Plan, Functional Test Plan, Acceptance Test Plan, Non-Functional Test Plan. Testing objectives and criteria are provided in the tables below.

   2. Describe the objectives of the testing process.

      ANSWER: The objective is to ensure that all the aspects of the developed project are working properly.

   3. Describe the types of testing that the project will use.
   4. Describe the strategy for unit testing of the individual subsystems. This includes an indication of the subsystems that will undergo unit tests or the criteria to be used to select subsystems for unit test. Test cases are NOT included here.
   5. Specify the integration testing strategy used. Describe the tests that will be performed in order to verify the interfaces between the subsystems of the software system. This section includes a discussion of the order of integration of subsystems. Test cases are NOT included here.

      ANSWER: The strategy focuses mainly on the interfaces & flow of data/information between the modules. The priority is to be given for the integrating links rather than the unit functions which are already tested.

   6. Specify the strategy for testing the software once it has been deployed. This section includes a discussion of the order of acceptance by software function. Test cases are NOT included here.
   7. Identify the limits under which the program is expected to perform (memory constraints, disk space constraints, etc):

ANSWER: **Not applicable for the project: it's a cloud-based web application where all heavy processes are running on the server and not on the client-side. Any current basic web server will be able to host this project without the need for specific parameters.**

8. Refer to the functional requirements that specify acceptable performance.
9. Describe the tools that you will use for testing.

4. Schedule and Deliverables
    1. Describe the test deliverables that will be created during the project lifecycle. Include two tables, one for the schedule of tasks, another for the list of deliverables:
        1. Acceptance test
        2. Unit test
        3. System/Integration test
        4. Stress test
        5. Performance test
        6. Screen prototypes
        7. Defect reports and summaries
        8. Test logs and reports

        Describe the reports that will be generated by the testing process.

The specific grading criteria are as follows:

- Does the TR include an Unit Test Plan? *[5 pts.]* **[DONE]**
- Does the TR include an Integrated Test Plan? *[5 pts.]* **[DONE]**
- Does the TR include a Functional Test Plan? *[5 pts.]* **[DONE]**
- Does the TR include an Acceptance Test Plan? *[5 pts.]* **[DONE]**
- Does the TR include a Non-Functional Test Plan? *[5 pts.]* **[DONE]**

Release 1.0

UNIT TEST PLAN

Unit Tests for Registration

Test #	Input	Expected Output	Test conditions
1	Username = Lastname Email = Password =	Message: Enter all credentials	Testing null input from user
2	Username = Lastname Email = last Password = 123456	Message: Please enter an email address.	Testing invalid email format

3	Username = Lastname Email = last@name.com Password =	Message: Password Required	Testing null password

Unit Tests for Login

Test #	Input	Expected Output	Test conditions
1	Email = lastname@name.com Password = 1234	Message: Missing credentials	Testing invalid password
3	Email = lastname Password = 123456	Message: Missing credentials	Testing invalid email
5	Email = Password =	Message: Missing credentials	Testing null input from a user
4	Email = lastname@name.com Password = 123456	Log into the system	Testing valid email

Unit Tests for Homepage

Test #	Input	Expected Output	Test conditions
1	Search: Godzilla	Showing search result for query “Godzilla”	Testing search bar
2	Clicking LATEST button	Showing latest movies from database	Testing LATEST button
3	Clicking UPCOMING button	Showing upcoming movies from database	Testing UPCOMING button
4	Clicking TOP RATED button	Showing top rated movies from database	Testing TOP RATED button

Unit Tests for Movie details

Test #	Input	Expected Output	Test conditions
1	Browser request for Godzilla movie details	Showing movie details for Godzilla	Testing details for specific movie
2	Clicking on Godzilla trailer	Playing Godzilla trailer	Testing trailer playing
3	Click star rating and submit	Updated rating for Godzilla	Testing rating system

Unit Tests for Coming soon

Test #	Input	Expected Output	Test conditions
1	Browser request for Coming soon movies	Showing list of upcoming movies	Testing list of coming soon movies

Unit Tests for Top Rated

Test #	Input	Expected Output	Test conditions
1	Browser request for Top Rated movies	Showing list of top rated movies	Testing list of top rated movies

Unit Tests for Filter

Test #	Input	Expected Output	Test conditions
1	Search: God Language: Year: Rating:	Showing list of search results for title with God	Testing searching by title
2	Search: Language: English Year: Rating:	Showing list of movies in English language	Testing searching by language
3	Search: Language: Year: 2020 Rating:	Showing list of movies for 2020 year	Testing searching by year
4	Search: Language: Year: Rating: 8.5	Showing list of movies with rating 8.5	Testing searching by rating

ACCEPTANCE TEST PLAN

Test case	Full description of user story	Acceptance Test	Name(s) of contributing Developer(s)
AT01	A User can Register for an account	**Test with inputs:** User's information is valid	Dinara, Vlad

		Expected outcome: User is being registered **Test with inputs:** User information is invalid **Expected outcome**: User is shown an error message	
AT02	A User can Login in the account	**Test with inputs:** User's combination of email and password is valid **Expected outcome**: User is logged in **Test with inputs:** User information isn't valid **Expected outcome**: User is shown an error message	Nikita
AT03	A User can see the homepage	**Test with inputs:** User requests the home page **Expected outcome**: User's browser renders the homepage successfully	Rohan, Dinara
AT04	A User can search for movies	**Test with inputs:** User types the name of movie that exists in the database **Expected outcome**: User sees a page with all the movies that contain that word **Test with inputs:** User types in the name of movie that doesn't exist in the database **Expected outcome**: User is shown a page with the message of 0 results for the user's request	Nikita
AT05	A User can filter the movies	**Test with inputs:** User enters the parameters that match the movies in the database **Expected outcome**: User sees a page with all the movies that match these filter parameters **Test with inputs:** User enters the parameters that don't match the movies in the database **Expected outcome**: User is shown a page with the message of 0 results for the user's	Hai

		request	
AT06	User can see Coming Soon movies	**Test with inputs:** User requests the “Coming Soon” page **Expected outcome**: User’s browser renders the “Coming Soon” successfully with the proper results fetched from the database	Hai
AT07	User can see Top Rated movies	**Test with inputs:** User requests the “Top Rated” page **Expected outcome**: User’s browser renders the “Top Rated” successfully with the proper results fetched from the database	Hai
AT08	User can see Movie details	**Test with inputs:** User requests the movie details page of a particular movie **Expected outcome**: User’s browser renders the page with the details of a specific movie successfully	Rohan

INTEGRATED TEST PLAN

Test case	Test Case Objective	Test Case Description	Expected Result
IT01	Check the link between Registration and Login modules	Enter login credentials from a newly registered user and click on the Login button	To be successfully logged in
IT02	Check the link between Search and Movie Details modules	Enter the parameters for the movie search and press Search	The search result page will show movies which details match the parameters

FUNCTIONAL TEST PLAN

#	Test Case Objective	Test Case Description	Expected Result

FT01	Registration	Test if a user can register with valid credentials	User is successfully registered
FT02	Login	Test if a registered user can login with valid credentials	Registered user is successfully logged in
FT03	Homepage	Test if any user can access homepage from any state	User can access homepage
FT04	Search Movies	Test if any user can search movies using keywords	User can perform search by keywords
FT05	Filter Movies	Test if any user can filter movies (perform advanced search)	User can filter movies (perform advanced search)
FT06	View Recommendations	Test if a user can see movie recommendations	User is able to see movie recommendations
FT07	View Movie Details	Test if a user can see details of the selected movie	User is able to access and see the details of the selected movie

NON-FUNCTIONAL TEST PLAN

#	Test Case Objective	Test Case Description	Expected Result
NF01	Usability (UX)	Testing different pages from various browsers and sizes of screens	Proper display of all pages
NF02	Scalability	Testing the capacity of the MongoDB database	The applications performs reliably
NF03	Performance	Testing the web application with the internet connection speed	The web application is able to perform even with the slow connection

Release 2.0

ACCEPTANCE TEST PLAN

#	Test Case	Description	Developers and Contributors
TA01	Basic Search	**Test with input:** User types in the title of the movie **Expected Outcome:** The search results are displayed Example: User Input = “God” Results = “Godzilla”, “The Godfather”, etc.	Vlad
TA02	Advanced Search	**Test with input:** User this time enters movie title, release year, and language **Expected Outcome:** the movie with the specific title, release date, and language is displayed. Example: User Input = “Godzilla vs Kong”, “2021”, “English”. Results = “Godzilla vs. Kong”	Vlad
TA03	UI with Search Results	**Test with input:** User clicks on search button to initiate search **Expected Outcome:** User is able to successfully search and see the search results organized Example: User searches “God” Results = the following movies posters are organized and displayed in rows	Rohan
TA04	Movie Details UI + Rating Front-End	**Test with input:** User clicks on movie details button **Expected Outcome:** User is able to view the movie title, date of release, rating, synopsis, and	Dinara + Rohan

		genre. Example: User clicks on Movie info for “Godzilla Vs. Kong” Results = Godzilla Vs Kong, March 31, 2021, Sci-fi/Action, Movie Summary	
TA05	Movie rating implementation	**Test with input:** User clicks on the 5 stars rating slider. **Expected Outcome:** User’s rating based on stars is saved for that movie. Example: User slides rating for The Godfather to 4.5stars Results: the 4.5 star rating is saved and is displayed when the player views that movie details.	Arielle
TA06	UI Logout function/ logged out state	**Test with input:** User clicks the logout button. **Expected Outcome:** User is successfully logged out of the account. Example: User clicks logout button Results = Successfully logs out and login button is displayed back in top right	Dinara

FUNCTIONAL TEST PLAN

#	Test Case Objective	Test Case Description	Expected Result
TF01	Basic Search	Test if the user can search for basic movies' names on the search bar.	Users can see the movies they are looking for.
TF02	Advanced Search	Test if the user can advanced search movies' names on the advanced search page	Users can see the movies and the keywords related to what is being searched.
TF03	Movie Rating	Test if the user can rate a movie.	Users can successfully see the movie is being rated and stored.
TF04	Admin Panel: Movie Add	Test if Admin can add a movie into the database	Admin can properly add movies by UI.
TF05	Admin Panel: Movie Listing	Test if Admin can see all listing movies in the database	Admin can see the listing movies with no trouble.