# Grover Web Automation Challenge. 2021

## Approximated Date/Time Log

- Sat 21 August
    - 10:00  to 14:30

## APPROACH

1. Manual exploration and assertion of applications behaviour
    a. Explore movie list, genre, sorting and layout changes.
    b. Formulate websites expected behaviour
2. Define and identify test case workflow
    a. I used this in creating priority and test order
3. Design test structure based on step 2
    a. Identify repeated functions
    b. I used this in identifying possible code abstracting
4. Write tests.
5. Complete other tasks (Documentation & summary)
    a. Commit to Github

## TEST CASES

1. The Top 250 (http://www.imdb.com/chart/top) page returns at least 1 movie,in the list.
2. The previous should be true, for each of the sorting options
3. Assert returned result accordingly
4. Assert each sorting options navigating to the "Western" genre

## EXPECTATIONS

1. When designing your tests, please have in mind the test suite may easily grow to 100's of them. Easy addition and debugging should be taken into account.
2. Granular commits.
3. Clear, maintainable code

Applicant: Enitan Christopher

4. Leverage design patterns

## APPROACH BREAKDOWN

The breakdown explains how each of the approach items addresses some of the task expectations.

### 1. Manual exploration and assertion of applications behaviour

**Result:**

The application is suitable for mobile and desktop view however specifications requested were on found desktop therefore my focus was on desktop view.

1. I noticed there seems to be no correlation between sorted option and movie genre selected according to app behaviour
    a. Either there is a bug: Failed Expected behaviour on Search page header text and result sorting "Feature Film, Rating Count at least 25,000, Western (Sorted by IMDb Rating Descending)" which suggests page is not sorted by selected genre
    b. OR the website's behaviour is as designed to be sorted by IMDb Ratings.
    c. I'm unsure if specification 4 requires assertion for (sorting/genre) feature relationship or page redirects.
2. The purpose of the page fits in the user story: As a user,
    a. when I visit the page https://www.imdb.com/chart/top I should see a list of movies, a sorting and Genre feature
    b. and when I sort the list by an option, I should see the same movie list in related sorting
    c. and when I visit a genre section, I should see a new page with movies listed in my selected genre and sorting filters.
3. On initial load of test page, the sorting is always defaulted to "Ranking"
4. Page response to sorting action is dynamically loaded in view.

**Decisions**

1. For Result 1, I decided to test for failing scenario and pronounce a bug for reasons below
    a. The appended link in the Western genre contains a default sorting parameter similar to chart page and suggests intended changes: *sort=user_rating*
        i. When I changed this manually in the search URL, I had a valid refined result. EG *sort=ranking*. Screenshots are in folder */docs/bug_asset*

## 2. Define and identify test case workflow

**Result:**

1. The Top 250 (http://www.imdb.com/chart/top) page returns at least 1 movie,in the list.
2. The previous should be true, for each of the sorting options
3. Assert returned result accordingly
4. Assert each sorting options navigating to the "Western" genre

Tests are assertions on page elements and functionality, the first 3 tests are user interaction assertions on the chart page while the fourth requires assertions on another page view.

**Decisions**

Test the criteria 1, 2 and 3 collectively and 4 separately, the approach here will allow us to fulfil expectation 3.

1. Should the result list or filter grow we can expand test scope with repeats of similar activity within one test.
2. Test criterion 4 separately for 3 reasons
    a. There seems to be a bug on asserting criterion 4.
    b. The handling/display/layout of page results of "Western genre" is dependent on the search page. Functionality and tests for that would be in a search spec. I could not define test scope limit

c. Actual test for western genre can be done by asserting embedded link url on chart page without page navigation
    i. This allows us to reduce redundant tests and speed up test passing time when search functionality is not tested or needed.

## 3. Design test structure based on step 2: (Define and identify test case workflow)

**Result:**

I based the code structure on functional abstraction methods, this reduces complex dependencies in the code and abstract only functions that are reused across the spec file or test suite.

Leveraging Cypress Commands. This allows me fulfil expectation 1 and 4; where reuse can be easily done if needed and easy to interpret for anyone; both code and results.

**Decisions**

1. All tests in this task are in one spec sheet
    a. Page and sorting assertions
    b. Sorting and Genre assertions
2. Test for criteria 1, 2 and 3 in one test and 4 in another
3. Abstract actions on step 3 into a reusable function command
4. Abstract assertions on criterion 4 to a function command

## BUGS & RESULTS

1. Please note point 1 Result 1c
    a. I'm unsure if spec 4 requests assertion for feature relationship or link and page redirects
2. Under 2 decisions 2 b  I decided to limit the test scope to the chart page.
3. Failing tests: Test 2: *Sorting and Genre assertions*