Who: Maryan Mazadi, Sigrunn Sky, Matthew Uhlar and Nicolas Carson
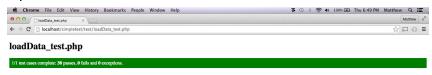
Title: Project Netflix

Vision: Choices are hard to make, and sometimes you want simple choices made for you. This is a Web App designed to help users choose movies based on what is playing on Netflix in the genre they specify.

Automated Tests: We used tellurium to verify homepage html elements. Here is a test verifying that the results work for each genre selection; https://app.te52.com/sharing/GhZ2ZLyI5RG83rP09Q7-8g . For some reason tellurium thinks there are two action & adventure check boxes and it's not recognizing the international checkbox even though the screenshots verify that these checkboxes are on the home page. We also used tellurium to verify the homepage contents for some reason the tester doesn't recognize a few of the checkboxes names but their existence can be verified by looking at the screenshot. Here is a link to the test results of verifying the homepage contents including all checkboxes: https://app.te52.com/sharing/dFSS0g2I5RG83rP09Q7-8g . We also can usebrowsershots.org to test our website on a range of browsers. Here is the link to browsershots: http://browsershots.org/https://csci-3308-project-netflix-nicolas9101.c9.io/# . We also used ping tools (http://tools.pingdom.com/fpt/#!/edgYOe/https://csci-3308-project-netflix-nicolas9101.c9.io/results.php) to generate a speed test of our website.

Simpletest (http://www.simpletest.org/) was used to test the PHP code which loads all film information into the database. The automated tests ensure that all information is stored in the correct table inside the database, can be fetched from the database, and that nonsensical or duplicate films are not inserted into the database. Test were run through local host. Results:

User Acceptance Tests:

Project Name: Project Netflix

Test Case Template

Test Case ID: TEST_1   **Test Designed by:** <Nicolas Carson>

Test Priority (Low/Medium/High): Med   **Test Designed date:** <11/3/2015>

Module Name: Project Netflix Homepage   **Test Executed by:** <Nicolas Carson>

Test Title: Verify that homepage shows   **Test Execution date:** <11/10/2015>
correctly with proper genre options
Description: Test Project Netflix Homepage

Pre-conditions: User has the website URL or a local host of index.html
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to Project Netflix Page	https://csci-3308-project-netflix-nicolas9101.c9.io/	User should be able to access and view Homepage	User is navigated to Project Netflix homepage	Pass	
2	Select Movie Genre	Action & Adventure	User should be able to see genre options and select a desired genre	User is able to select checkboxes corresponding to their genre selection	Pass	
3	Click Find Movie Button		User should be able to click a button to generate a random movie suggestion of a movie in the user's selected genre.	The Submit Button works and returns the results from the database	Pass	

Post-conditions:
        The user's genre selection is used to query the project netflix movie database for a movie with a matching genre. The randomly selected movie's data is then returned from the database and displayed on a results page for the user to view.

Project Name: Project Netflix

Test Case Template

Test Case ID: TEST_2   **Test Designed by:** <Nicolas Carson>

Test Priority (Low/Medium/High): Med **Test Designed date:** <11/10/2015>

Module Name: Project Netflix Homepage with 2 genre selections **Test Executed by:** <Nicolas Carson>

Test Title: Verify that homepage shows correctly with proper genre options and that submitting selection returns results with 2 genre selections **Test Execution date:** <11/10/2015>

Description: Test Project Netflix Homepage and function

Pre-conditions: User has the website URL or a local host of index.html
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to Project Netflix Page	https://csci-3308-project-netflix-nicolas9101.c9.io/	User should be able to access and view Homepage	User is navigated to Project Netflix homepage	Pass	
2	Select multiple Movie Genres	Documentary + Action & Adventure	User should be able to see genre options and select a desired genre	User is able to select checkboxes corresponding to their genre selection	Pass	
3	Click Find Movie Button		User should be able to click a button to generate a random movie suggestion of a movie in the user's selected genre.	The Submit Button works and returns the results from the database, but currently it returns two results instead of one result with two matching genres	Pass	The submit button returns the randomized results but currently it returns two

						movies, one for each genre instead of a movie whose genre is both selections.

Post-conditions:
   The user's genre selection is used to query the project netflix movie database for a movie with a matching genre. The randomly selected movie's data is then returned from the database and displayed on a results page for the user to view but currently it returns two results from two selections instead of the one result like it should.

Project Name: Project Netflix

Test Case Template

Test Case ID: TEST_3	**Test Designed by:** <Nicolas Carson>
Test Priority (Low/Medium/High): Med	**Test Designed date:** <11/10/2015>
Module Name: Project Netflix Homepage 3 Genres	**Test Executed by:** <Nicolas Carson>
Test Title: Verify that homepage shows correctly with proper genre options and that submitting selection returns results with 3 genre selections	**Test Execution date:** <11/10/2015>
Description: Test Project Netflix Homepage and function	

Pre-conditions: User has the website URL or a local host of index.html
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to Project Netflix Page	https://csci-3308-project-netflix-nicolas9101.c9.io/	User should be able to access and view Homepage	User is navigated to Project Netflix homepage	Pass	
2	Select multiple Movie Genres	Comedy + Classic + Drama	User should be able to see genre options and select a desired genre	User is able to select checkboxes corresponding to their genre selection	Pass	
3	Click Find Movie Button		User should be able to click a button to generate a random movie suggestion of a movie in the user's selected genre.	The Submit Button works and returns the results from the database but not fully the way intended because it returns a result for each genre selection instead of one result that matches all three selections. But each genre is at least returning movie results.	Pass	The submit button returns the randomized results but currently it returns three movies, one for each genre instead of a movie whose genre is all three selections.

Post-conditions:
The user's genre selection is used to query the project netflix movie database for a movie with a matching genre. The randomly selected movie's data is then returned from the database and displayed on a results page for the user to view but currently it returns two results from two selections instead of the one result like it should.

Project Name: Project Netflix

Test Case Template

Test Case ID: TEST_4

Test Priority (Low/Medium/High): Med

Module Name: Project Netflix Homepage

Test Title: Test the load time of the page, analyze it and find bottlenecks.
Description: Test Project Netflix Homepage

Test Designed by: <Pingdom.com>

Test Designed date: <11/10/15>

Test Executed by: <Sigrunn Sky>

Test Execution date: <11/10/15>

Pre-conditions: User has the website URL that we can enter into the site.
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Enter main website URL into tester site: http://tools.pingdom.com/fpt/#!/cQU9Mz/https://csci-3308-project-netflix-nicolas9101.c9.io/results.php	https://csci-3308-project-netflix-nicolas9101.c9.io/	We expect more insight into the performance grade, page size, and run time of our website.. We expect results above 80/100 for the performance grade and below one second of runtime.	Our performance grade was an 88/100 and our load time is 348 milliseconds. Our HTML code is 80% while the CSS is 20% of time spent per content.	Pass
2	Enter results page URL into tester site	results.php	For this site, we expect it to be a little bit slower than the homepage, but are still hoping for results above 80/100 for the performance grade and below one second of runtime.	This page was a little bit slower than the homepage with a performance grade was an 83/100 and a load time of 436 milliseconds. The page size was 270B with the majority of this being the web browser waiting for data from the server. This wait accounts for 53% of the "time spent per state" load time analysis. The second in time was the connecting state, which accounts for approximately 20% if "time spent per state" during the load time analysis.	Pass

Post-conditions:
    The website has a reasonable runtime and there aren't any bottlenecks in our code. The results of this test match the end-user experience exactly, and we expected our results.php page to load slightly slower than the homepage because the homepage is static data while the

results page is dynamic.

Version Control Software: [Project Netflix Github](https://github.com/Clacious/CSCI_3308_Project_Netflix/tree/master)
(https://github.com/Clacious/CSCI_3308_Project_Netflix/tree/master)