

Software Engineering CSC 648/848

The MovAI Project - Milestone 4

Section 4, Team 2

Team

Team Lead	Ashmitha Pais
Scrum Master	Steve Betts
Github Lead / Back-End	Preet Dhaliwal
Back-End Lead	Chris Farnsworth
Front-End Lead	Abdul Barrie
Product Owner / Front-End	Nathan Loo

History

Initial Draft	03/05/2023
----------------------	------------

Unit Testing -

Description of 5 P1 features and the list of corresponding source files in the Github:

1. Login

- a. This priority feature allows the user to login to the MovAI application.
- b. The user enters their username and password
- c. The database checks if the values are already populated
- d. If so, it logs the user into the application and sets them a token

File Names

Login.test.js
tests.py

2. Registration

- a. This priority feature allows the user to register into the MovAI application
- b. The user enters an email ID, username, password and reconfirms their password
- c. The database populates the values in the database
- d. It leads the user to the main page

File Names

Signup.test.js
tests.py

3. Generating movie recommendations

- a. The user chooses genre, year, age and time limit
- b. Based on the preferences, MovAI will recommend 10 movies

File Names

Recommend.test.js
tests.py

4. Logout

- a. The user clicks on logout
- b. The user's session cookie and CSRF token is wiped out

File Names

Logout.test.js
tests.py

5. Users choosing preferences

- a. The user has options like genre, year, age and time limit
- b. The user selects two or three from each category
- c. The preferences are added and sent to open AI to generate movie recommendations

File Names

tests.py

Github link of test cases directory:

https://github.com/CSC-648-SFSU/csc648-spring23-04-team02/blob/main/Application_SE02/back-end/team02app/tests.py

Description of functional and statement coverage:

1. Login

The login functionality is tested on both the front end and back end.

On the front end, tests were written to check:

- The page is rendered correctly by detecting whether certain html elements were displayed.
- Check username input:
 - Check if input takes in text.
 - Check if logging in fails if an invalid (i.e. doesn't match requirements) username is inputted.
- Check if password input takes in text.
- Check if, upon clicking 'submit,' the appropriate fetch statement is called.

On the back end, tests were written to check:

- If the api URL for login was connected to the correct function in models.py.
- Check if invalid users (i.e. doesn't exist) were rejected from the database

2. Registration

Registration functionality is tested on both front end and back end.

On the front end, tests were written to check:

- The page is rendered correctly by detecting whether certain html elements were displayed.
- Check if fetch is called on validated submission.
- Check if fetch is not called on an invalidated submission.

On the back end, tests were written to check:

- If the api URL for registration was connected to the correct function in models.py.

- If a response code of 200 (OK) was returned upon a successful post.
- If the data was correctly returned to the front end upon a successful post.

3. Recommending movies

The recommendation functionality is tested on both the front end and back end.

On the front end, tests were written to check:

- The page(s) rendered correctly by detecting whether certain html elements were displayed.
- Check user's input:
 - Check if genre is taken as input.
 - Check if the year is taken as input.
 - Check if age is taken as input.
 - Check if the time limit is taken as input.
 - Check if recommendations fail if these values aren't parsed in.
- Check if, upon clicking 'submit,' the appropriate fetch statement is called.

On the back end, tests were written to check:

- If the api URL for recommendations was connected to the correct function in models.py.
- Check if movie values generated are present within our database or not

4. Logout

The logout functionality is tested on both the front end and back end.

On the front end, tests were written to check:

- The page is rendered correctly by detecting whether certain html elements were displayed.
- On rendering logout:
 - Check if the user gets logged out.
 - Check if the user is redirected.
- Check if, upon clicking 'submit,' the appropriate fetch statement is called.

On the back end, tests were written to check:

- If the api URL for logout was connected to the correct function in models.py.
- Check if the response code of logout once logged in is correct or fails

5. User provides preferences

This functionality allows the users to select their preferences so the recommendations are tailored to meet their needs. This test is checked both at front-end and back-end.

- On rendering the home page after logging in:
 - Check if the user is logged in.
 - Check if the user has selected values for genre, age, time limit and year.
- Check if, upon clicking 'submit,' the appropriate fetch statement is called.

On the back end, tests were written to check:

- If the api URL for the next page after every click was connected to the correct function in models.py.
- Check if the response code of recommendations once logged in is correct or fails

Front - End Tests -

```
Test Suites: 4 passed, 4 total
Tests:       16 passed, 16 total
Snapshots:   0 total
Time:        3.656 s
Ran all test suites.
```

Watch Usage: Press w to show more.

Mode	LastWriteTime	Length	Name
-----	-----	-----	-----
d-----	5/3/2023 9:25 AM		csc648-spring23-04-team02

```
PASS src/__test__/Logout.test.js
PASS src/__test__/Login.test.js
  • Console
```

```
console.log
  invalid json response body at reason: Unexpected end of JSON input
  at src/Login.js:45:19
```

```
PASS src/__test__/Signup.test.js
  • Console
```

```
console.log
  invalid json response body at reason: Unexpected end of JSON input
  at src/Signup.js:56:19
```

```
PASS src/__test__/recommend.test.js
  • Console
```

```
console.log
{
  genre: [ 'Action' ],
  year: [ '2000 to 2020' ],
  runtime: [ 'Under 90' ],
  age: [ 'PG-13' ]
}

  at generate_results (src/recommend.js:21:13)

console.log
  invalid json response body at reason: Unexpected end of JSON input
  at src/recommend.js:43:17
```

Back-End Tests -

```
Found 8 test(s).
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (mysql.W002) MySQL Strict Mode is not set for database connection 'default'
  HINT: MySQL's Strict Mode fixes many data integrity problems in MySQL, such as data truncation upon insertion, by escalating warning
s into errors. It is strongly recommended you activate it. See: https://docs.djangoproject.com/en/4.1/ref/databases/#mysql-sql-mode

System check identified 1 issue (0 silenced).
Movie 'Mission: Impossible - Ghost Protocol 2011' not found in the database.
Movie 'The Mummy Returns 2001' not found in the database.
Movie 'The Bourne Ultimatum 2007' not found in the database.
Movie 'Pirates of the Caribbean: The Curse of the Black Pearl 2003' not found in the database.
Movie 'Skyfall 2012' not found in the database.
Movie 'The Great Wall 2016' not found in the database.
Movie 'The Lost World: Jurassic Park 1997' not found in the database.
Movie 'The Chronicles of Narnia: The Lion, the Witch and the Wardrobe 2005' not found in the database.
Movie 'Indiana Jones and The Last Crusade 1989' not found in the database.
.....
Ran 8 tests in 15.994s

OK
Destroying test database for alias 'default'...
```

Description of your integrated CI workflow:

During this milestone we integrated CI through github actions in order to run our jest tests when merging into main. This was done by creating a YAML file that specifies the instructions needed for github to run these tests. This YAML file was also altered to handle future tests that are added for the front end.

Integration Testing

Test Case ID	RE_1	Test Case Description	Test recommendation component		
Created By	Ashmitha Pais	Reviewed By	Steve Betts	Version	1
<u>QA Testers' Log</u>	Check if recommendations are displayed				
Tester's Name	Ashmitha Pais	Date Tested	29-Apr-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to chrome browser		1	Adventure	
			2	1960 to 1980	
			3	150 to 180	
			4	R	
Test Scenario	Verify if recommendations are received				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open chrome browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	

2	Login with credentials	Credentials can be entered	As expected	Pass	
3	Click on the Log In button	User is logged in	As expected	Pass	
4	Click on get started button	Preferences page opens up	As expected	Pass	
5	Select a genre	Genre is selected	As expected	Pass	
6	Select a year range	Year is selected	As expected	Pass	
7	Select a runtime range	Runtime is selected	As expected	Pass	
8	Select an age range	Age is selected	As expected	Pass	
9	Wait for results in the /recommend page that opens	10 movie titles are displayed	As expected	Pass	

Test Case ID	RE_2	Test Case Description	Test recommendation component		
Created By	Ashmitha Pais	Reviewed By	Steve Betts	Version	1
<u>QA Testers' Log</u>	Check if recommendations are displayed				
Tester's Name	Ashmitha Pais	Date Tested	29-Apr-23	Test Case Pass/Fail/Not	Fail
S #	Prerequisites:		S #	Test Data	
1	Access to chrome		1	Adventure	

	browser				
			2	1940 to 1960, 1920 to 1940	
			3	120 to 150, 150 to 180	
			4	PG-13PG	
Test Scenario	Verify if recommendations are received				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open chrome browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Login with credentials	Credentials can be entered	As expected	Pass	
3	Click on the Log In button	User is logged in	As expected	Pass	
4	Click on get started button	Preferences page opens up	As expected	Pass	
5	Select a genre	Genre is selected	As expected	Pass	
6	Select a year range	Year is selected	As expected	Pass	
7	Select a runtime range	Runtime is selected	As expected	Pass	
8	Select an age range	Age is selected	As expected	Pass	
9	Wait for results in the /recommend page that	10 movie titles are displayed, leads to a	No recommendations	Fail	

	opens	blank page			
--	-------	------------	--	--	--

Test Case	RE_3	Test Case Description	Test recommendation component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if recommendations are displayed				
Tester's Name	Ashmitha Pais	Date Tested	3-May-23	Test Case Pass/Fail/Not	Fail
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox browser		1	Fantasy, Comedy, Horror	
			2	1980 to 2000, 2000 to 2020	
			3	90 to 120 Minutes	
			4	PG-13, R	
Test Scenario	Verify if recommendations are received				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	

1	Open chrome browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Login with credentials	Credentials can be entered	As expected	Pass	
3	Click on the Log In button	User is logged in	As expected	Pass	
4	Click on get started button	Preferences page opens up	As expected	Pass	
5	Select a genre	Genre is selected	As expected	Pass	
6	Select a year range	Year is selected	As expected	Pass	
7	Select a runtime range	Runtime is selected	As expected	Pass	
8	Select an age range	Age is selected	As expected	Pass	
9	Wait for results in the /recommend page that opens	10 movie titles are displayed, leads to a blank page	No recommendations	Fail	

Test Case ID	SU_1	Test Case Description	Test the sign up component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if registration component signs user into the website on Firefox.				

Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Email = 'test@email.com'	
			2	Username = 'testuser1234'	
			3	Password = 'Password1!'	
Test Scenario	Verify if the user is logged in upon registration				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Register with credentials	Credentials can be entered	As expected	Pass	
3	Click on the 'Sign Up' button	User is logged in	As expected	Pass	
4	Redirect to	Website	No Response	Not Executed	

	home page	should automatically redirect to the home page			
--	-----------	------------------------------------------------	--	--	--

Test Case ID	SU_2	Test Case Description	Test the sign up component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if registration component allows duplicate usernames to be registered				
Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Email = 'test@email.com'	
			2	Username = 'testuser1234'	
			3	Password = 'Password1!'	
Test Scenario	Verify that no duplicate usernames				

	can be registered.				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Register with credentials	Credentials can be entered	As expected	Pass	
3	Click on the 'Sign Up' button	Website should not log in user.	As expected	Pass	
4	Prevent sign up	Error message displayed	No Response	Not Executed	

Test Case ID	SU_3	Test Case Description	Test the sign up component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if registration allows users to register with insecure passwords				
Tester's Name	Nathan Loo	Date Tested	3-May-23	Test Case Pass/Fail/No	Fail

				t	
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Email = 'test@email.com'	
			2	Username = 'testuser5790'	
			3	Password = 'password'	
Test Scenario	Verify that registration will not allow the user to sign up with an insecure password				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Register with credentials	Credentials can be entered	As expected	Pass	
3	Click on the 'Sign Up' button	Website should not log in user.	User is logged in	Fail	
4	Prevent sign up	Error message	No Response	Fail	

		displayed			
--	--	-----------	--	--	--

Test Case ID	LI_1	Test Case Description	Test the login component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
QA Testers' Log	Check if logging into a user on the database works.				
Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Username = 'testuser1234'	
			2	Password = 'Password1!'	
Test Scenario	Verify that logging in works correctly.				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	

1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Enter credentials	Credentials can be entered	As expected	Pass	
3	Click on the 'Log in' button	User should be logged in	As expected	Pass	
4	Redirect to home page	Website should automatically redirect to the home page	As expected	Pass	

Test Case ID	LI_2	Test Case Description	Test the login component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if logging into a nonexistent account causes errors				
Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Username = 'Orange'	

			2	Password = 'Password451!'	
Test Scenario	Verify that if the user presents credentials that don't match the database, they will not be logged in.				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Enter credentials	Credentials can be entered into the login form	As expected	Pass	
3	Click on the 'Log in' button	Website should not log in user	As expected	Pass	
4	Prevent log in	Error message displayed	No Response	Not Executed	

Test Case ID	LO_1	Test Case Description	Test the logout component		
---------------------	------	------------------------------	---------------------------	--	--

Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if logout works.				
Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Username = 'testuser1234'	
			2	Password = 'Password1!'	
Test Scenario	Verify that if a user is logged in, the logout button logs out the user from the website.				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Enter	Credentials can	As expected	Pass	

	credentials	be entered			
3	Click on the 'Log in' button	User should be logged in	As expected	Pass	
4	Redirect to home page	Website should automatically redirect to the home page	As expected	Pass	
5	Click on the 'Logout' button	User should be logged out	As expected	Pass	

Test Case ID	LO_2	Test Case Description	Test the logout component		
Created By	Nathan Loo	Reviewed By	Ashmitha Pais	Version	1
<u>QA Testers' Log</u>	Check if logout works.				
Tester's Name	Nathan Loo	Date Tested	2-May-23	Test Case Pass/Fail/Not	Pass
S #	Prerequisites:		S #	Test Data	
1	Access to Firefox Browser		1	Email = 'test@email.com'	
			2	Username = 'testuser6903'	
			3	Password = 'Password1!'	

Test Scenario	Verify that if a user is logged in via 'sign up', the logout button logs out the user from the website.				
Step #	Step Details	Expected Results	Actual Results	Pass/Fail/Not Executed/Suspended	
1	Open browser and navigate to http://54.241.156.230/	Site should open	As expected	Pass	
2	Enter credentials	Credentials can be entered	As expected	Pass	
3	Click on the 'Log in' button	User should be logged in	As expected	Pass	
4	Redirect to home page	Website should automatically redirect to the home page	As expected	Pass	
5	Click on the 'Logout' button	User should be logged out	As expected	Pass	

URL - Bug Tracker

<https://github.com/CSC-648-SFSU/csc648-spring23-04-team02/issues>

3. Coding Practices -

1. Coding Style

In order to practice good coding habits, we implemented the 'Prettier' tool to format our front end code. This tool will help keep code formatted for future front end changes and runs each time the code is built in order to avoid in line formatting issues being pushed into production.

Formatted files:

csc648-spring23-04-team02\Application_SE02\front-end\src\Login.js
csc648-spring23-04-team02\Application_SE02\front-end\src\Member.js
csc648-spring23-04-team02\Application_SE02\front-end\src\recommend.js
csc648-spring23-04-team02\Application_SE02\front-end\src\results.js
csc648-spring23-04-team02\Application_SE02\front-end\src\select_genre.js
csc648-spring23-04-team02\Application_SE02\front-end\src\select_runtime.js
csc648-spring23-04-team02\Application_SE02\front-end\src\select_year.js
csc648-spring23-04-team02\Application_SE02\front-end\src\Signup.js

2. Documentation Generation

Code comments were added to all of the front end files relating to the 5 P1 features. These code comments were used to generate HTML documentation using JSDoc. The documentation can be found by navigating to the project, and opening the index.html file through google chrome. The index.html file is located at path
(csc648-spring23-04-team02\Application_SE02\front-end\src\out\index.html).

URL-

https://github.com/CSC-648-SFSU/csc648-spring23-04-team02/tree/main/Application_SE02/front-end/src/out