

Testing Document: Movie Review App

Overview

This document outlines the unit tests and system tests conducted to validate the core functionality of the **CommunityReviews** App, including expected vs. actual results and how they align with system and functional requirements.

Test Instructions

Make sure all dependencies have been installed and the virtual environment activated. Then, in the “**backend**” directory of the application, run:

```
python -m unittest discover tests
```

This command will automatically discover and run all test files in the tests/ directory.

Test Summary

Test Suite	Test Name	Data Set/Action	Purpose	Output	
				Expected	Actual
Main Features	Exploratory	On the home page, sign up and login with the same credentials	Verify user registration and login	Redirected to movie list after login	Redirected to movie list after login
Main Features	Exploratory	Click “Continue as Guest” on home page	Ensure application is accessible to guests	Redirected to movie list	Redirected to movie list
Main Features	Exploratory	Login using username= “Admin” and password= “123”	Verify admin login	Redirected to admin page	Redirected to admin page
Main	Exploratory	Login as the	Test movie	New movie is	New movie is

Features		admin and enter the new movie's title, description and image URL	addition by admin	added to movie list	added to movie list
Main Features	Exploratory	Login as the admin, add a new movie, then click on the "Remove" button on the newly added movie	Test movie removal by admin	Newly added movie is removed from movie list	Newly added movie is removed from movie list
Main Features	Exploratory	Click "Continue as guest" or login with valid credentials on home page	View movies list	show movie list	shows movie list
Main Features	Exploratory	Login and select a movie. Then write and submit a review	Submit and view a review	Review added to list of reviews with the username attached	Review added to list of reviews with the username attached
Main Features	Exploratory	Login, select a movie and submit a review. Then logout, login with another account and select the same movie	Fetch reviews submitted by other users	Review submitted with first account should be listed under "Reviews"	Review submitted with first account should be listed under "Reviews"
Main Features	Exploratory	Login, select a movie and submit a review. Finally, click "Summarize Feedback"	Generate movie review summary	Summarized movie review is generated	Summarized movie review is generated
Main Features	Exploratory	Login, select a movie, submit a review, and click "Summarize Feedback".	Retry movie review summarization	Summarized movie review is regenerated	Summarized movie review is regenerated

		Then, click "Back to movie" and click "Summarize Feedback" again			
Database	test_add_user	username ="testuser", password ="password123"	Add user to database	User inserted and retrievable	User inserted and retrievable
Database	test_add_movie	title ="Test Movie" plot ="Test Plot", poster_url ="http://test.com/poster.jpg"	Add movie to database	Movie saved in database	Movie saved in database
Database	test_add_review	username ="testuser", password ="password123" movie ="Test Movie" review_text ="Great movie!"	Add review to database	Review inserted with correct foreign keys	Review inserted with correct foreign keys
API	test_user_registration	username ="testuser", password ="password123" POST /register	Register via API	201 Created	201 Created
API	test_user_login	username ="testuser", password ="password123" POST /login	Login via API	200 OK	200 OK
API	test_get_movies	GET /movies	Fetch all movies	200 OK	200 OK
API	test_get_reviews	username: reviewer1 password: password123 movie_id: 1 review_text:	Fetch a particular movie's reviews	200 OK and review_text in response	200 OK and review_text in response

		"This was a solid movie." GET /movies/1/reviews			
API	test_view_other_users_reviews	username: userA password: pass movie_id: 1 review_text: "Complex but brilliant!" username: userB password: pass GET /movies/1/reviews	View other users' movie reviews	200 OK with review_text and reviewer's username in response	200 OK with review_text and reviewer's username in response
API	test_review_summary	username: testuser password: test123 movieTitle= "Test Movie" review_text= "Loved it!" POST /summary	Summarize reviews	200 OK	200 OK

Test Validation Plan

Functional Requirements	Validation Tests
Allow users to sign up	test_user_registration, Exploratory - Verify user registration and login
Allow users to login to their accounts	test_user_login, Exploratory - Verify user registration and login
Allow users continue as guest	Exploratory - Ensure application is accessible to guests
Allow admin to see privileges above average	test_add_movie, Exploratory - Verify admin

users	login, Test movie addition by admin, Test movie removal by admin
Display a diverse movie selection	test_get_movies, Exploratory - View movies list
Allow users complete review submission	test_add_review, Exploratory - Submit and view a review
Allow users see and view other user's reviews	test_view_other_users_reviews, Exploratory - Fetch reviews submitted by other users
Save users' review information	test_add_review, test_get_reviews, Exploratory - Submit and view a review, Fetch reviews submitted by other users
Review summarization from LLM	test_review_summarization, Exploratory - Generate movie review summary
Review resummarization from LLM	Exploratory - Retry movie review summarization

Conclusion

All tests passed successfully, indicating that the system meets the requirements specified and that the application is ready for further integration or deployment.