

Code Coverage Report, Group 30-TA

Code Coverage Analysis Report for login.test.js

1. Initial Code Coverage Measurement

- Tool Used: Jest
- Initial Code Coverage Percentage: 29.16% (Statements), 18.42% (Branch), 27.27% (Functions), 29.47% (Lines)
- Screenshot: Below is the initial code coverage report for login.test.js.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	28.81	31.42	32	28.93	
LeaseAgreement.js	20.63	42.3	25	20.63	4-152,175-187,205,227-238
Login.js	29.16	18.42	27.27	29.47	6,51-57,63,71-180
RentPayment.js	100	66.66	100	100	9-22

Test Suites: 3 passed, 3 total
Tests: 9 passed, 9 total
Snapshots: 0 total

2. Analysis of Missed Statements

- Missed Statements:
 - Branching for Network and Server Errors: Initial tests did not fully cover cases like network failures, server 404 errors, or 500 internal server errors, resulting in missed branches in error handling logic.
 - Sign-Up Form Validations: Password complexity rules (e.g., requiring a digit, special characters, uppercase letters) and input validations were partially covered, leaving some statements untested.
 - API URL Custom Handling: There was no test to handle potential variations in the API base URL, leading to missing branches related to custom configuration logic.

3. Additional Test Scripts to Improve Coverage

- New Tests Written:
 - Network Error Handling: test('should handle network error during login') was added to simulate and handle network connectivity issues during login attempts.
 - Server-Side Error Responses: Additional tests were added for handling 404 (test('should display error for 404 response during login')) and 500 (test('should display

generic error if fetch fails with unknown status code during login')) server responses, covering previously missed scenarios.

- Password Complexity Validations: Tests were introduced to check password complexity during sign-up, including checks for non-alphanumeric characters, digits, and uppercase letters (e.g., test('should display an error if password does not contain a non-alphanumeric character during sign-up')).
- Malformed JSON Response Handling: The test test('should handle malformed JSON response gracefully during login') was added to ensure the application can handle unexpected response formats.
- Custom API Base URL Handling: test('should return custom base URL if __API_BASE_URL__ is set') covered logic to handle custom API configurations.

4. Updated Code Coverage Measurement

- Updated Coverage Percentage: 89.58% (Statements), 92.1% (Branch), 72.72% (Functions), 90.52% (Lines)
- Screenshot: Below is the updated code coverage report for login.test.js.

at ValidateUser (rental_project/wwwroot/js/RentPayment.js:28:5)

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	53.38	71.42	52	53.61	
LeaseAgreement.js	20.63	42.3	25	20.63	4-152,175-187,205,227-238
Login.js	89.58	92.1	72.72	90.52	6,88-90,151,167-171
RentPayment.js	100	66.66	100	100	9-22

Test Suites: 3 passed, 3 total
Tests: 27 passed, 27 total
Snapshots: 0 total
Time: 0.855 s, estimated 1 s
Ran all test suites.
○ suhaibskhan@Suhaibs-MacBook-Pro-2 Group-30-TA %

Code Coverage Analysis Report for RentPayment.test.js

Initial Code Coverage Measurement:

- Tool Used: Jest
- Initial Code Coverage Percentage: 100% (Statements), 66.66% (Branch), 100% (Functions), 100% (Lines)
- Screenshot: Below is the initial code coverage report for RentPayment.test.js.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	85.29	87.93	81.81	85.79	
LeaseAgreement.js	75	85.71	88.88	75	4-16,76-77,80-81,104,163
Login.js	89.58	92.1	72.72	90.52	6,88-90,151,167-171
RentPayment.js	100	66.66	100	100	9-22

As the line coverage was already 100%, we did not need to write any additional tests for the Rentpayment.js file.

Code Coverage Analysis Report for LeaseAgreement.test.js

1. Initial Code Coverage Measurement

- Tool Used: Jest
- Initial Code Coverage Percentage: 75% (Statements), 85.71% (Branch), 88.88% (Functions), 75% (Lines)
- Screenshot: Below is the initial code coverage report for login.test.js.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	85.29	87.93	81.81	85.79	
LeaseAgreement.js	75	85.71	88.88	75	4-16,76-77,80-81,104,163
Login.js	89.58	92.1	72.72	90.52	6,88-90,151,167-171
RentPayment.js	100	66.66	100	100	9-22

2. Analysis of Missed Statements

- Missed Statements:
 - Add Tenants Verification: There was no test to confirm that the Add Tenants button added UI elements.
 - Fetch response ok: No test to verify an alert being called if the Fetch Response returned ok.
 - Fetch response not ok: No test to verify an alert being called if the Fetch Response did no return ok.

3. Additional Test Scripts to Improve Coverage

- New Tests Written:
 - Add button adds UI: Clicks the add button which creates new UI elements. Check if UI elements with the class “input-container” exist.
 - Fetch response ok: Mock an ok fetch response and expect the alert to be called.
 - Fetch response not ok: Mock a not ok fetch response and expect the alert to be called.

4. Updated Code Coverage Measurement

- Updated Coverage Percentage: 95% (Statements), 92.85% (Branch), 100% (Functions), 95% (Lines)
- Screenshot: Below is the updated code coverage report for login.test.js.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	92.35	89.65	86.36	92.89	
LeaseAgreement.js	95	92.85	100	95	77,104,163
Login.js	89.58	92.1	72.72	90.52	6,88-90,151,167-171
RentPayment.js	100	66.66	100	100	9-22