**Defect Report for Budget Application**

**Summary**

Three sets of tests (run\_tests\_models.py, run\_tests\_system.py, and run\_tests\_endpoints.py) were executed to validate the application's functionality across models, system workflows, and API endpoints. All tests passed successfully, except for two failures observed in run\_tests\_system.py. This indicates a mostly robust application with minor areas requiring attention.

**Results Overview**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Suite** | **Total Tests** | **Passed** | **Failed** | **Errors** | **Execution Time** |
| Models Tests | 41 | 41 | 0 | 0 | 0.019s |
| System Tests | 2 | 0 | 1 | 1 | 18.221s |
| Endpoints Tests | 7 | 7 | 0 | 0 | 0.228s |
| Overall | 50 | 48 | 1 | 1 | 18.468s |

A graph with orange and pink bars

Description automatically generated with medium confidence

**Defects Identified**

**1. Timeout in test\_create\_budget (System Tests)**

* **Description**: The TimeoutException occurred during the test, indicating that Selenium could not find the specified element within the timeout period.
* **Likely Cause**:
  + The element was not rendered on the page within the given timeframe.
  + An incorrect locator was used in the test script.
* **Recommendations**:
  + Double-check the element's locator and ensure it matches the page's structure.
  + Increase the timeout in the WebDriverWait.
  + Debug by printing the page source or capturing a screenshot during the failure.

**2. Assertion Failure in test\_login (System Tests)**

* **Description**: After attempting a login, the test remained on /auth/login instead of redirecting to /budget/dashboard.
* **Likely Cause**:
  + Invalid login credentials used in the test.
  + Backend login validation or form submission issues.
* **Recommendations**:
  + Validate the credentials used in the test.
  + Add debug logs to confirm whether the login form was submitted correctly.
  + Review the backend logs to determine if the login request was processed.

**Testing Observations**

**Models Tests (run\_tests\_models.py)**

* **Passed All 41 Tests**:
  + Verified functionality for expense and income management, budget creation, savings calculation, and report generation.
  + Successfully handled edge cases like negative or zero values and duplicate entries.
* **Execution Time**: Fast execution (0.019s), indicating efficient unit testing.

**System Tests (run\_tests\_system.py)**

* **2 Tests Executed**:
  + Verified core workflows for budget creation and login.
  + Encountered 1 error (test\_create\_budget) and 1 failure (test\_login).
* **Execution Time**: 18.221s, expected for Selenium-based tests.

**Endpoint Tests (run\_tests\_endpoints.py)**

* **7 Tests Executed**:
  + Covered login, registration, and report generation endpoints.
  + Successfully validated responses for authenticated/unauthenticated access, duplicate registrations, and missing resources.
* **Execution Time**: 0.228s, demonstrating efficient endpoint response handling.

**Recommendations**

**For System Tests**

1. **Enhance Debugging**:
   * Print the current page URL and source during test execution to pinpoint issues.
   * Capture screenshots upon failures for better analysis.
2. **Optimize Selenium Tests**:
   * Use headless mode to reduce execution time.
   * Increase timeout thresholds for element waits.
3. **Expand Coverage**:
   * Test additional workflows such as adding expenses, generating reports, and deleting budgets.

**Conclusion**

The Budget application demonstrates strong functionality across most tested areas. However, the two defects in the system tests highlight areas requiring immediate attention. Resolving these issues and expanding test coverage will further enhance the application's reliability and robustness.