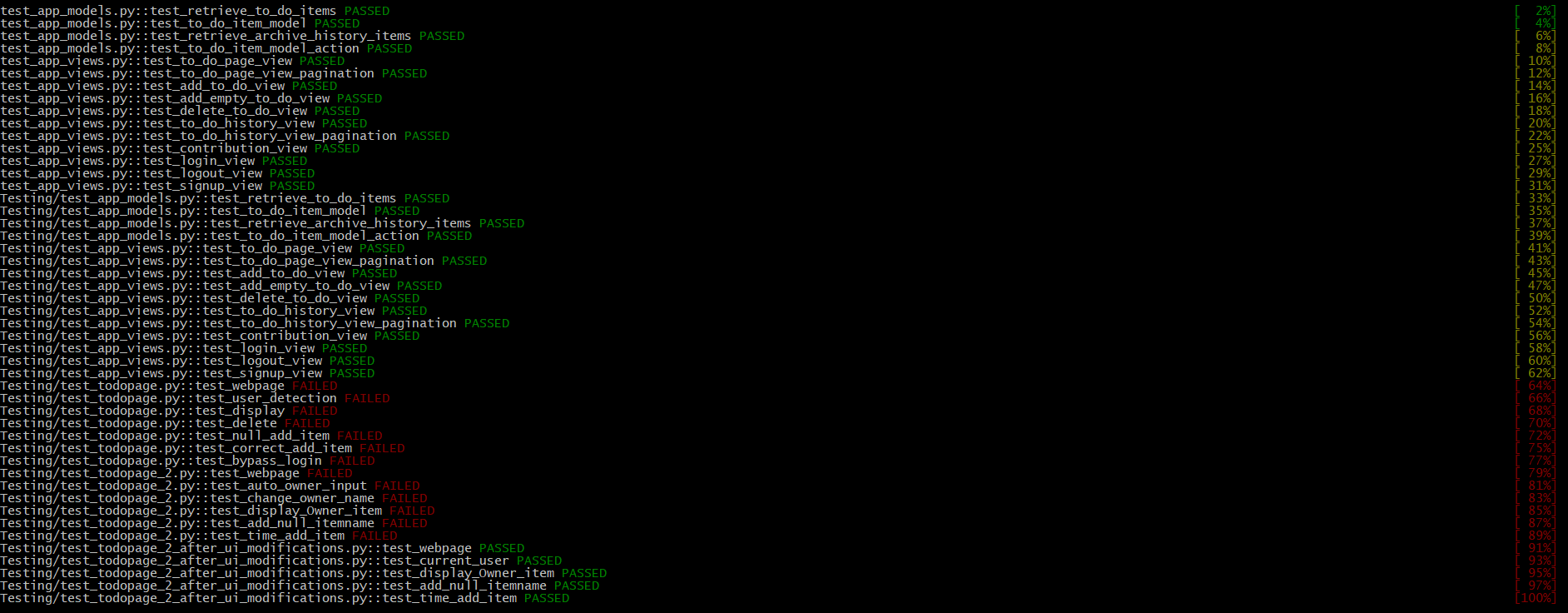
# **Justifications of Results**

## Introduction

This justification applies to users who clone the completed **ETI-To-Do-App** project from the repository and run the pytest -v --cov command on the root folder of the project as some of the tests results seen would be different from the screenshots captured during development and testing of this web application.

## Developer Unit Tests

Under the **To\_do\_page** folder, all developer unit tests inside *test\_todopage.py* and *test\_todopage\_2.py* would fail if a user were to clone this project and run the pytest command inside this folder as seen in **Figure 1**.



**Figure 1**

One of the reasons identified for the failure is due to inclusion of the authentication views of the Django Authentication system for a change of requirements of the User Login Page in Sprint 2 to allow new users to sign up. As a result, the URL path for the login page was changed from <http://127.0.0.1:8000/login/login> to <http://127.0.0.1:8000/accounts/login>.

Since the developer unit tests both *test\_todopage.py* and *test\_todopage\_2.py* were completed and ran before the developer assigned to the issue for change of requirements for the User Login Page had finished working on the changes, the User Login Page URL used in these two files were not changed to the new URL.

With the other reason being the redesign of the To-Do Page and the To-Do History Page to improve the User Experience of the pages, the main HTML elements that used to display each To-Do Item and To-Do Item’s History was changed from an unordered list to a table with rows. This has resulted in the selenium web driver in the two developer unit test files not being able to find the elements which displayed data for the To-Do Item and To-Do Item’s History records.

After these turns of events, it was decided that there was a need of a new unit test file *test\_todopage\_2\_after\_ui\_modifications.py* to be created to consolidate all the changes made so far as well as retain the original unit tests done.

## QA Unit Tests

This is the tests results you would see if you clone the completed **ETI-To-Do-App** project from the repository and run the pytest command with the coverage option --cov on the **QA** folder of the project shown in Figure 2 below.

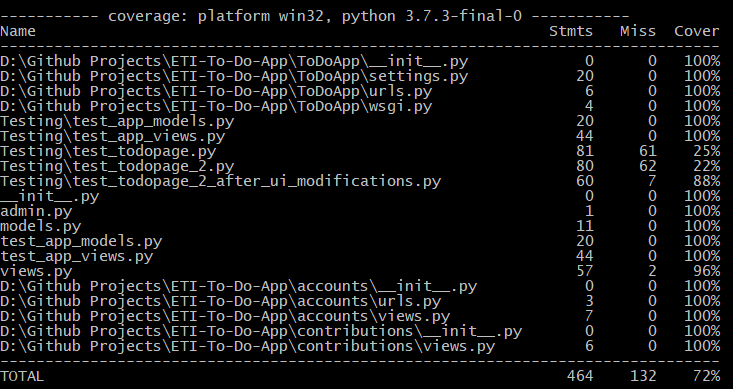


**Figure 2**

As you can see, the Sprint 1 QA unit tests would all fail like the Sprint 1 Developer unit tests due to the same reason where the URL path for the login page was changed after inclusion of the authentication views of the Django Authentication system for a change of requirements of the User Login Page in Sprint 2.

## Coverage

This is the coverage results you would see if you clone the completed **ETI-To-Do-App** project from the repository and run the pytest command with the coverage option --cov on the **To\_do\_page** of the project shown in Figure 3 below.



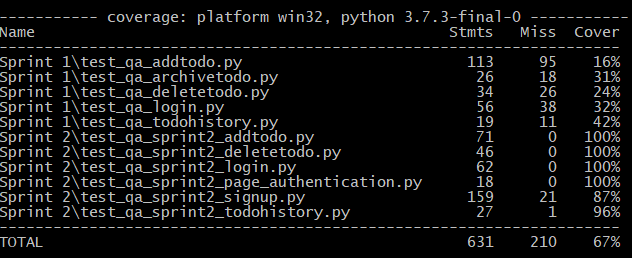
**Figure 3**

Since, the developer unit tests inside *test\_todopage.py* and *test\_todopage\_2.py* will fail due to the URL path for the login page was changed after inclusion of the authentication views of the Django Authentication system, the lines of code after getting the login page which includes block of codes in other unit test functions would not execute, hence explaining the low coverage percentages of these 2 files.

For the coverage of the *test\_todopage\_2\_after\_ui\_modifications.py* file, the reason for the loss of coverage was due to the many if else statements in the **test\_time\_add\_item()** unit test function to convert a datetime variable from the datetime library to the output of Django’s Model’s DateTimeField.

For the coverage of the *views.py* file, the missing 4% is due to the **logout\_view()** function in the *views.py* file that is not used as the authentication views of the Django Authentication system includes a default logout view which is currently used instead.

This is the coverage results you would see if you clone the completed **ETI-To-Do-App** project from the repository and run the pytest command with the coverage option --cov on the **QA** folder of the project shown in Figure 4 below.



**Figure 4**

Most noticeably again, the QA unit test coverage for QA unit test files under Sprint 1 is very low due to the same reasons that affect the low coverage percentage of the Developer unit tests inside *test\_todopage.py* and *test\_todopage\_2.py* files. The change in the login page URL path results in most lines of codes in each unit test function not being executed.

The loss in code coverage percentage for the *test\_qa\_sprint2\_signup.py* file is due to the first unit test function **test\_qa\_signup\_no\_input()** which calls the **handle\_if\_test\_user\_exists()** function to delete the test user used in the test case where the user successfully signs up before starting to run each unit test in this file. Taking a deeper look into this function, there is a section where there is a for loop running to check each username column in the Django User Administration page for the test user’s username. For the case where the test user does not exist in the database, a chunk of statements under the if block that checks for the test user’s username would not run which explains the 13% loss of coverage.

Lastly, for the loss in code coverage for the *test\_qa\_sprint2\_todohistory.py* file, it is due to an if-else block which depends on the number of records shown on the To-Do History page.

|  |
| --- |
| if total\_number\_of\_todo\_history\_items >= 5:  assert page\_1\_row\_count == 5  else:  assert page\_1\_row\_count == total\_number\_of\_todo\_history\_items |

As the new user interface for the To-Do History page uses pagination where each time 5 records are shown in each page, this if else block is used in the unit test to assert that if the number of records retrieved from the database is more than or equal to 5 records the first page would show 5 records, else the first page will show the number of records if there are less than 5 records. This would result in a 4% loss of code coverage depending on the number of records of To-Do History Items in the database.