Test Plan

Proof Checker Tool - SE691

Rakhfa Amin, Thoman Andrews, Kersley Jatto, Aiasha Sattar, Colton Shoenberger

Server-Side Test Plan

We are utilizing the Django framework for our project, which comes with a robust suite of built-in testing tools (https://docs.djangoproject.com/en/3.2/topics/testing/). Any filename that matches the pattern `test*.py` is recognized by Django as a test file. Django `TestCase`'s subclass the Python built-in `unittest` module, which is capable of performing both unit and integration testing. Any function in a test file that matches the pattern `test*` is recognized by Django as a test method, and is run when the command `python manage.py test` is run in the terminal. Django test methods support a wide variety of assertions (`assertTrue`, `assertEqual`, etc.) that can be utilized to validate that the system is behaving as expected.

We are complementing our use of Django tests with the Coverage.py tool, which provides code coverage reports when our tests are run. This allows us to identify exactly what lines of code are being executed (and perhaps more importantly, which lines are *not* being executed) when our tests are performed. It also provides us with percentages of lines executed in individual files and the overall system. On the next page, you will find a recent report of our overall code coverage. Our test suite currently includes 133 test cases with over 600 assertions. We are proud to report our code coverage is currently at 90% of server-side lines of code.

Code Coverage Report

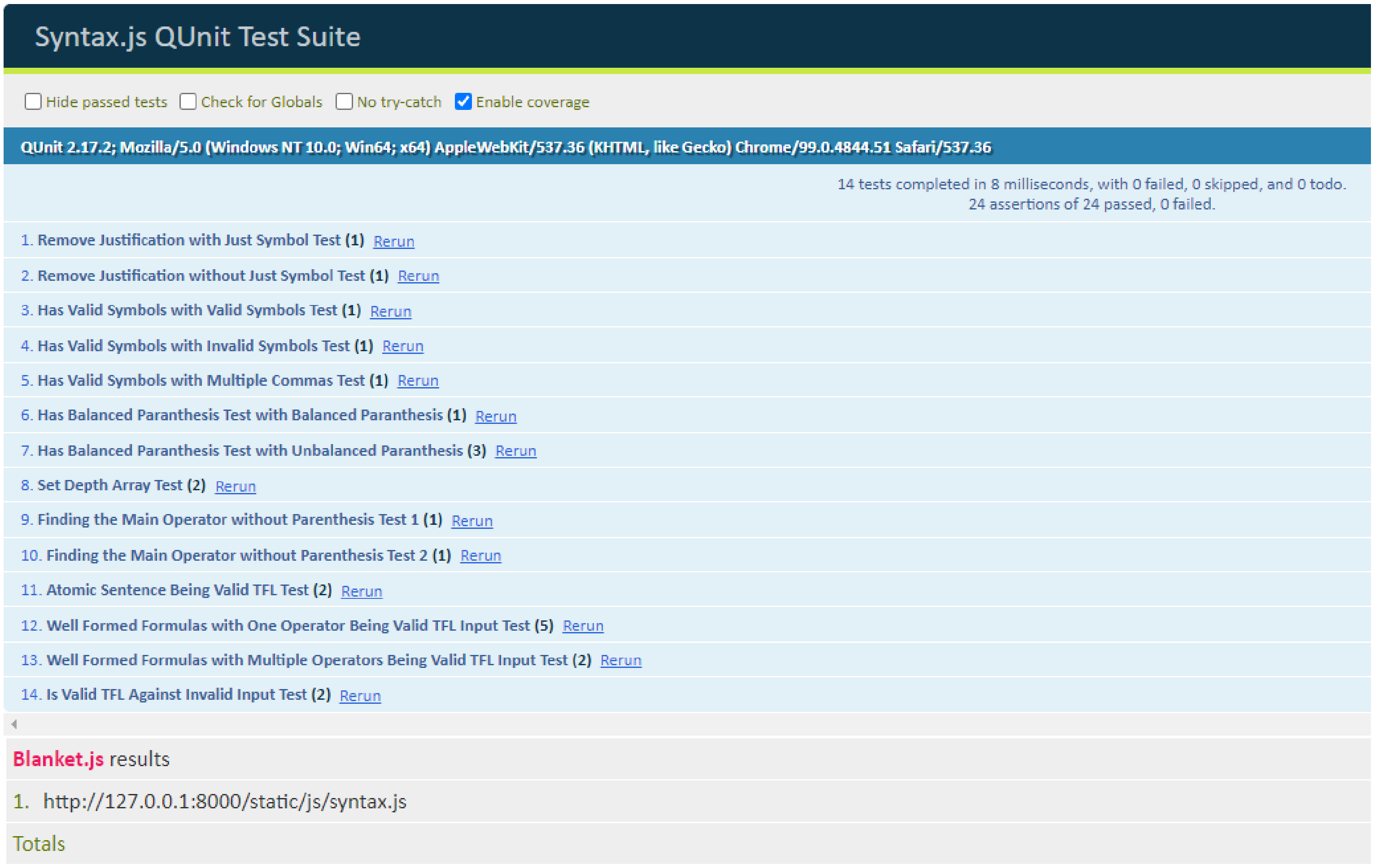
```
Name                                                        Stmts   Miss  Cover
-------------------------------------------------------------------------------
accounts\apps.py                                                4      0   100%
accounts\decorators.py                                         12      2    83%
accounts\forms.py                                              56      2    96%
accounts\tokens.py                                              6      0   100%
accounts\urls.py                                                9      1    89%
accounts\views.py                                             125     26    79%
assignments\admin.py                                            1      0   100%
assignments\apps.py                                             4      0   100%
assignments\forms.py                                           39     15    62%
assignments\models.py                                           1      0   100%
assignments\tests.py                                            1      0   100%
assignments\urls.py                                             5      0   100%
assignments\views.py                                          189    148    22%
courses\admin.py                                                1      0   100%
courses\apps.py                                                 4      0   100%
courses\forms.py                                                6      0   100%
courses\models.py                                               1      0   100%
courses\tests.py                                                1      0   100%
courses\urls.py                                                 3      0   100%
courses\views.py                                               86     59    31%
proofchecker\admin.py                                          26      3    88%
proofchecker\apps.py                                            4      0   100%
proofchecker\forms.py                                          30      3    90%
proofchecker\models.py                                        116     14    88%
proofchecker\proofs\proofchecker.py                            54      0   100%
proofchecker\proofs\proofobjects.py                            25      0   100%
proofchecker\proofs\proofutils.py                             389     45    88%
proofchecker\rules\assumption.py                               18      3    83%
proofchecker\rules\biconditionalelim.py                        38      7    82%
proofchecker\rules\biconditionalintro.py                       54      7    87%
proofchecker\rules\conditionalelim.py                          33      4    88%
proofchecker\rules\conditionalintro.py                         31      6    81%
proofchecker\rules\conjunctionelim.py                          33      7    79%
proofchecker\rules\conjunctionintro.py                         36      4    89%
proofchecker\rules\conversionofquantifiers.py                  91      9    90%
proofchecker\rules\demorgan.py                                 96     21    78%
proofchecker\rules\disjunctionelim.py                          47      7    85%
```

```
proofchecker\rules\disjunctionintro.py                             31      4    87%
proofchecker\rules\disjunctivesyllogism.py                         47     10    79%
proofchecker\rules\doublenegationelim.py                           35      7    80%
proofchecker\rules\equalityelim.py                                 60     11    82%
proofchecker\rules\equalityintro.py                                21      0   100%
proofchecker\rules\excludedmiddle.py                               45      7    84%
proofchecker\rules\existentialelim.py                              56      7    88%
proofchecker\rules\existentialintro.py                             38      7    82%
proofchecker\rules\explosion.py                                    28      4    86%
proofchecker\rules\indirectproof.py                                34      7    79%
proofchecker\rules\modustollens.py                                 51     12    76%
proofchecker\rules\negationelim.py                                 34      4    88%
proofchecker\rules\negationintro.py                                34      7    79%
proofchecker\rules\premise.py                                      26      0   100%
proofchecker\rules\reiteration.py                                  29      4    86%
proofchecker\rules\rule.py                                          6      1    83%
proofchecker\rules\rulechecker.py                                  51      2    96%
proofchecker\rules\universalelim.py                                37      7    81%
proofchecker\rules\universalintro.py                               48      7    85%
proofchecker\urls.py                                                8      0   100%
proofchecker\utils\binarytree.py                                  104      8    92%
proofchecker\utils\constants.py                                    15      0   100%
proofchecker\utils\follexer.py                                     37      7    81%
proofchecker\utils\folparser.py                                    66      8    88%
proofchecker\utils\numlexer.py                                     16      1    94%
proofchecker\utils\numparser.py                                     9      0   100%
proofchecker\utils\syntax.py                                       96      5    95%
proofchecker\utils\tfllexer.py                                     23      1    96%
proofchecker\utils\tflparser.py                                    36      0   100%
proofchecker\views.py                                             201     36    82%
prooftool\settings.py                                              35      0   100%
prooftool\urls.py                                                   4      0   100%
------------------------------------------------------------------------------
TOTAL                                                            5681    569    90%
```

Client-Side Test Plan

While Python accounts for over 50% of the code in our system, JavaScript accounts for about 30% (according to GitHub). Because JavaScript is not a compile-based language, conventional testing frameworks and strategies could not be used. Fortunately, there were frameworks and tools available to test the client side JavaScript code and we ultimately decided to also utilize the QUnit testing framework to test the client-side functionality of our system.

The QUnit framework is recommended for use by the main Django website for running tests against Django applications’ Javascript. By utilizing the test methods provided by an included qunit.js file, we are able to have tests that run asserts against Javascript methods in a manner similar to more prevalent testing frameworks like JUnit. Our application runs its QUnit tests through the Proof Checker website in the HTML which runs the qunit.js. And in order to obtain coverage, it works in tandem with the BlanketJS framework which is specialized in obtaining code coverage for QUnit. The QUnit is organized into an interactable GUI that shows existing tests while clicking file names listed below shows BlanketJS’s measure of code coverage. The following pages showcase the tests being run from the site and how coverage is displayed.



Syntax.js QUnit Test Suite

☐ Hide passed tests ☐ Check for Globals ☐ No try-catch ☑ Enable coverage

QUnit 2.17.2; Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36

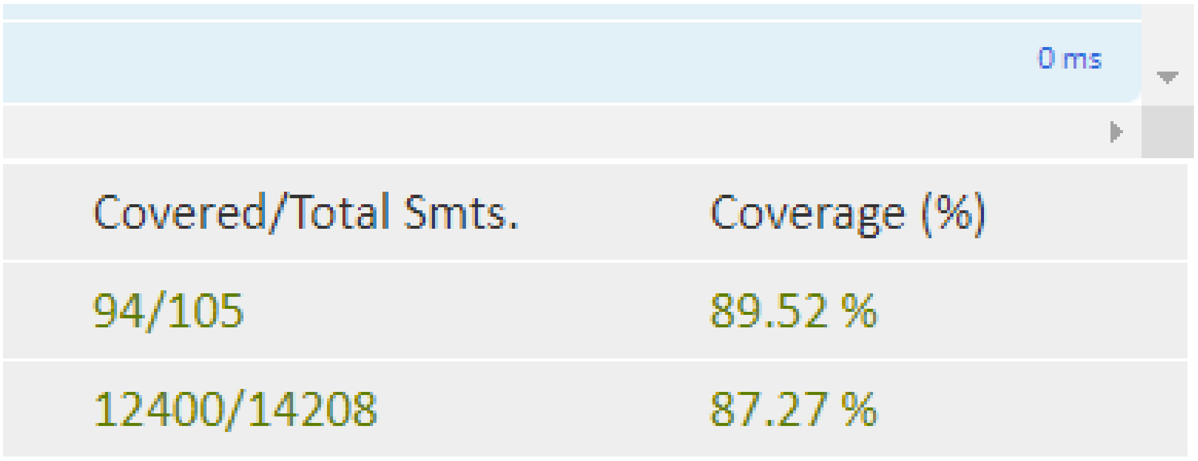
14 tests completed in 8 milliseconds, with 0 failed, 0 skipped, and 0 todo.
24 assertions of 24 passed, 0 failed.

1. Remove Justification with Just Symbol Test (1) Rerun
2. Remove Justification without Just Symbol Test (1) Rerun
3. Has Valid Symbols with Valid Symbols Test (1) Rerun
4. Has Valid Symbols with Invalid Symbols Test (1) Rerun
5. Has Valid Symbols with Multiple Commas Test (1) Rerun
6. Has Balanced Paranthesis Test with Balanced Paranthesis (1) Rerun
7. Has Balanced Paranthesis Test with Unbalanced Paranthesis (3) Rerun
8. Set Depth Array Test (2) Rerun
9. Finding the Main Operator without Parenthesis Test 1 (1) Rerun
10. Finding the Main Operator without Parenthesis Test 2 (1) Rerun
11. Atomic Sentence Being Valid TFL Test (2) Rerun
12. Well Formed Formulas with One Operator Being Valid TFL Input Test (5) Rerun
13. Well Formed Formulas with Multiple Operators Being Valid TFL Input Test (2) Rerun
14. Is Valid TFL Against Invalid Input Test (2) Rerun

Blanket.js results

1. http://127.0.0.1:8000/static/js/syntax.js

Totals



0 ms

Covered/Total Smts.	Coverage (%)
94/105	89.52 %
12400/14208	87.27 %

These images show the syntax.js file's coverage approximately 90%.