

Priyanka Karki, Garrett Hempy,
Mattie Yang, Matt Siddle,
Munkhsanaa Otgonbayar, Zachary Teutsch
CSCI 3308: Software Dev.

Milestone 5 - Testing

Account Database testing:

Testing this feature will start with creating dummy accounts in the database that the tester can use to login. If they successfully manage to login with the dummy information, then we know that our account database is functioning how it should. To verify that it is working as intended, we will give the tester a bad login, or a login that is not in the database. If the test does not manage to login with the bad login, then we can be sure that the login process works as intended.

Review population testing:

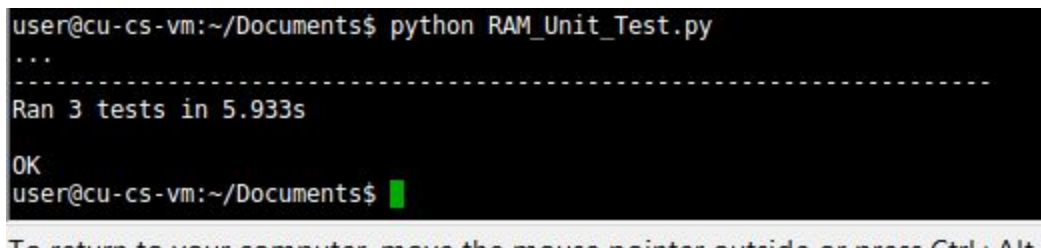
Testing for this feature involves checking to see if a particular review exists in the database, and that the page can correctly populate with data from the review database. To do this, we will give the tester a dummy review page to open, and we will compare the data pulled from the database to constants. If this data matches, we know that the website should be correctly populating the review page with the right information.

Search bar testing:

Testing this feature involves having a constant search field put into the tester in order to pull the songs review page. We will give the tester a song name to input into the search bar, and we will test the resulting data from the search against what it should be outputting in the test. If that succeeds, then we know our search function can bring users to the correct page. The other part of this testing involves testing to see if the search function will route users to the create a new review page if no such review exists. We will give the tester a search term that should not yield anything in the search function, and should route the user to the create a new review page. If that happens, we know our search function is working properly.

Automated Testing:

We wrote some helper functions to help with pulling and updating data from the database. Then, we wrote unit tests for these helper functions to make sure they are working properly before implementing them into our application. In order to run our tests, navigate to the RAM_Unit_Testing.py file. In terminal in the directory containing the unit test file, type “python RAM_Unit_Testing.py” The output will be the results of the unit tests. Below is a screenshot of the results of our test file:



```
user@cu-cs-vm:~/Documents$ python RAM_Unit_Test.py
...
-----
Ran 3 tests in 5.933s

OK
user@cu-cs-vm:~/Documents$
```

To return to your computer, move the mouse pointer outside or press Ctrl+Alt