

LyricRace

Who:

- Danish Mohammed
- Jackson Roberts
- Eric Weng
- Nhan Nguyen
- Ryan Wood
- Noah Cisernos

Test Plan:

The three features we were planning on testing were our login page, our song list, and our lyric database. For our login page to be acceptable we need to make sure that the login page communicates properly with our users and passwords database. To do this we will form tests to check whether a username and password that is in the database allows you to log in, if a username and a wrong password will prevent you from logging in, and if an invalid username will prevent you from logging in. The test for our song database will be very similar. For it to be acceptable, we need to be able to access songs that are on the list. To test this we will test whether to see if a song on the song list. For the lyric database to be acceptable, we need to make sure that the lyrics are all on the correct song. To do this we will test to see whether a lyric that is not in the song pops up in the song database and if it doesn't then the test is passed. We will also test to see if a lyric that is supposed to be in the song pops up, if it does, the test is passed. If these tests are all passed then our program works properly.

Finally, the core gameplay needs to be tested. This can be done by simply playing the game and ensuring that for every word that is entered, the corresponding word is revealed, and that the remaining words counter is reduced by 1.

Automated Tests:

We created our unit tests in Python. They test to ensure all backend API endpoints are functioning correctly. It tests the structure of the song select data, tests that login functions correctly, and tests that POST-only URL's are not accessible via GET. You can see the code at https://github.com/dimohammed328/CSCI3308-Project/blob/master/tests/test_endpoints.py

To run, clone the repository, and cd into the tests folder

You may need to run `pip install requests`

Afterwards, simply run `python3 test_endpoints.py`

And the test output will display:

```
tests — -bash — 80x24
Jacksons-MacBook-Pro:tests jacksonroberts$ python3 test_endpoints.py
Testing song select API for formatting and data
Query type correct
There is at least one song
Song data is structured correctly
All song select tests passed!
=====
Testing to ensure all POST-only endpoints reject GET requests
Route / passed successfully!
Route /login passed successfully!
Route /register passed successfully!
=====
Testing authentication for proper behavior
Testing with registered user "jax" with password "123"
Test passed for registered user!
Testing with unregistered user "nonexistent" with password "badpassword"
Test passed for unregistered user!
Testing for incorrect password on registered user "jax" with password "456"
Test passed for incorrect password
=====
All tests passed!
Jacksons-MacBook-Pro:tests jacksonroberts$ |
```