**Logged Out Home**

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Sat Apr 11 02:38:40 2020 End: Sat Apr 11 02:38:58 2020 Duration: 0:00:18.161000  broken: 0 (0.00%) failed: 1 (33.33%) skipped: 0 (0.00%) passed: 2 (66.67%) |  |

Test Results

Right-click to update field.

# Testing Premium Button - failed

Testing Premium Button

## Details

AssertionError: assert False

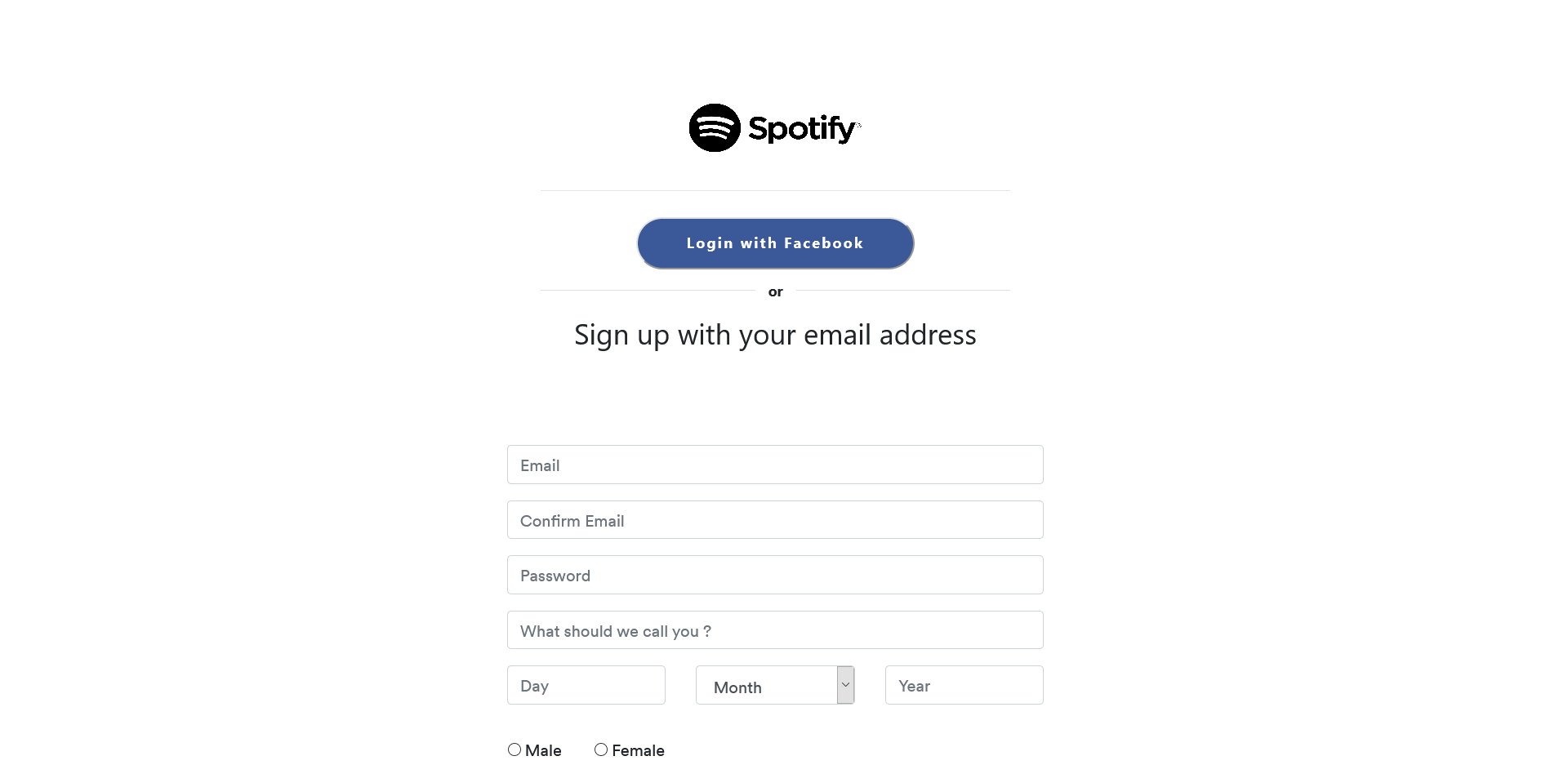
|  |
| --- |
| self = <Web\_Testing.Tests.test\_loggedOutHome.TestLoggedOutHome object at 0x00000261EA6E8A88>, setup\_final = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium Button")  @allure.title("Testing Premium Button")  @allure.description("Testing Premium Button")  @pytest.mark.Do  @pytest.mark.LoggedOutHome  def test\_case\_3(self, setup\_final):  logged\_out\_home = LoggedOutHome(self.driver)  logged\_out\_home.click\_premium()  time.sleep(2)  if str(self.driver.current\_url).find('premium') != -1:  WebHelper().report\_allure("Premium button working", self.driver)  assert True  else:  WebHelper().report\_allure("Premium button failed", self.driver) > assert False E assert False  Web\_Testing\Tests\test\_loggedOutHome.py:95: AssertionError |

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] Premium button failed. Went to Sign up page instead



## Test Teardown

[Fixture] setup\_final::0

# Testing Login Button - passed

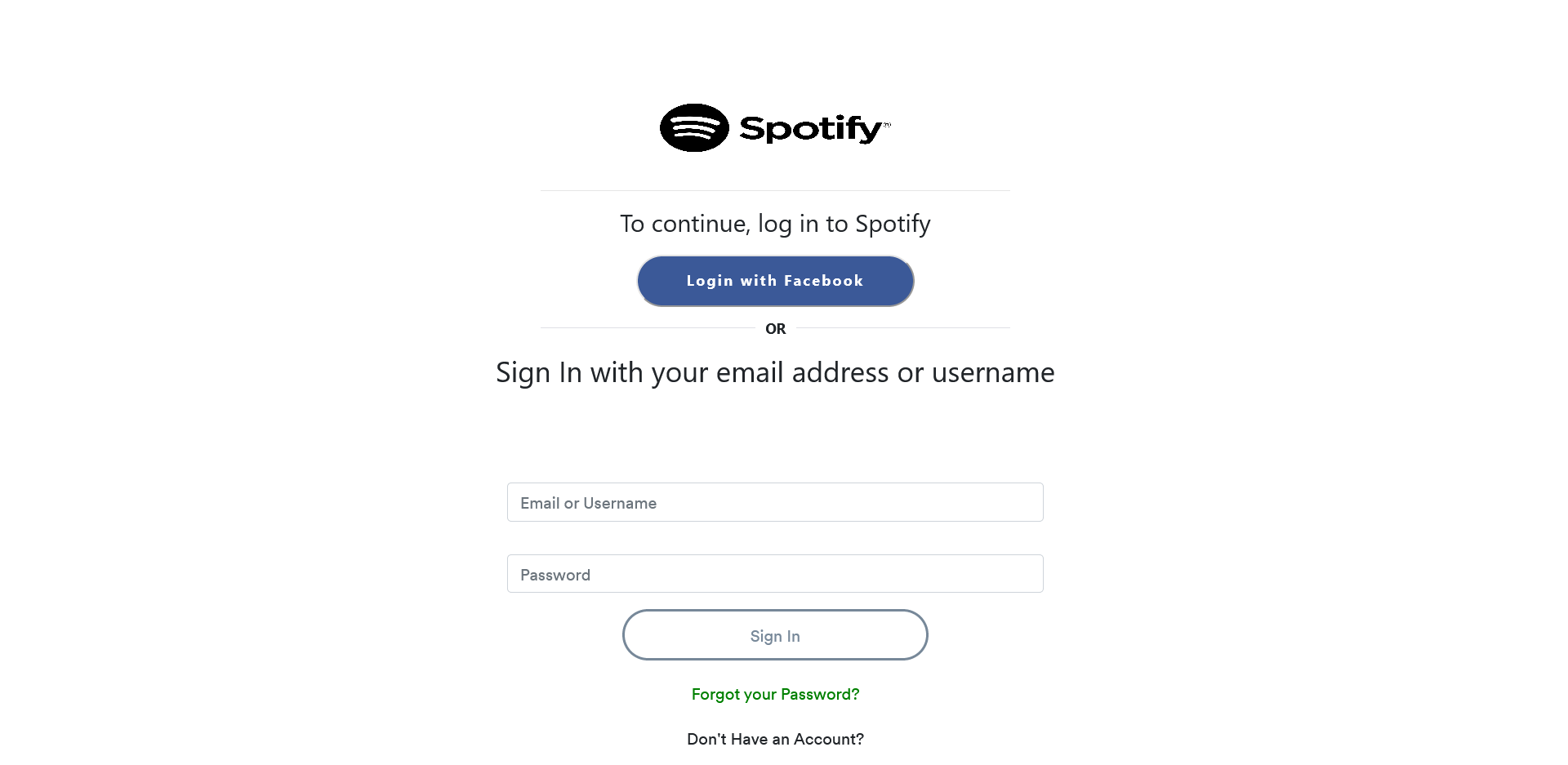
Testing Login Button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Login button working



## Test Teardown

[Fixture] setup::0

# Testing Sign up Button - passed

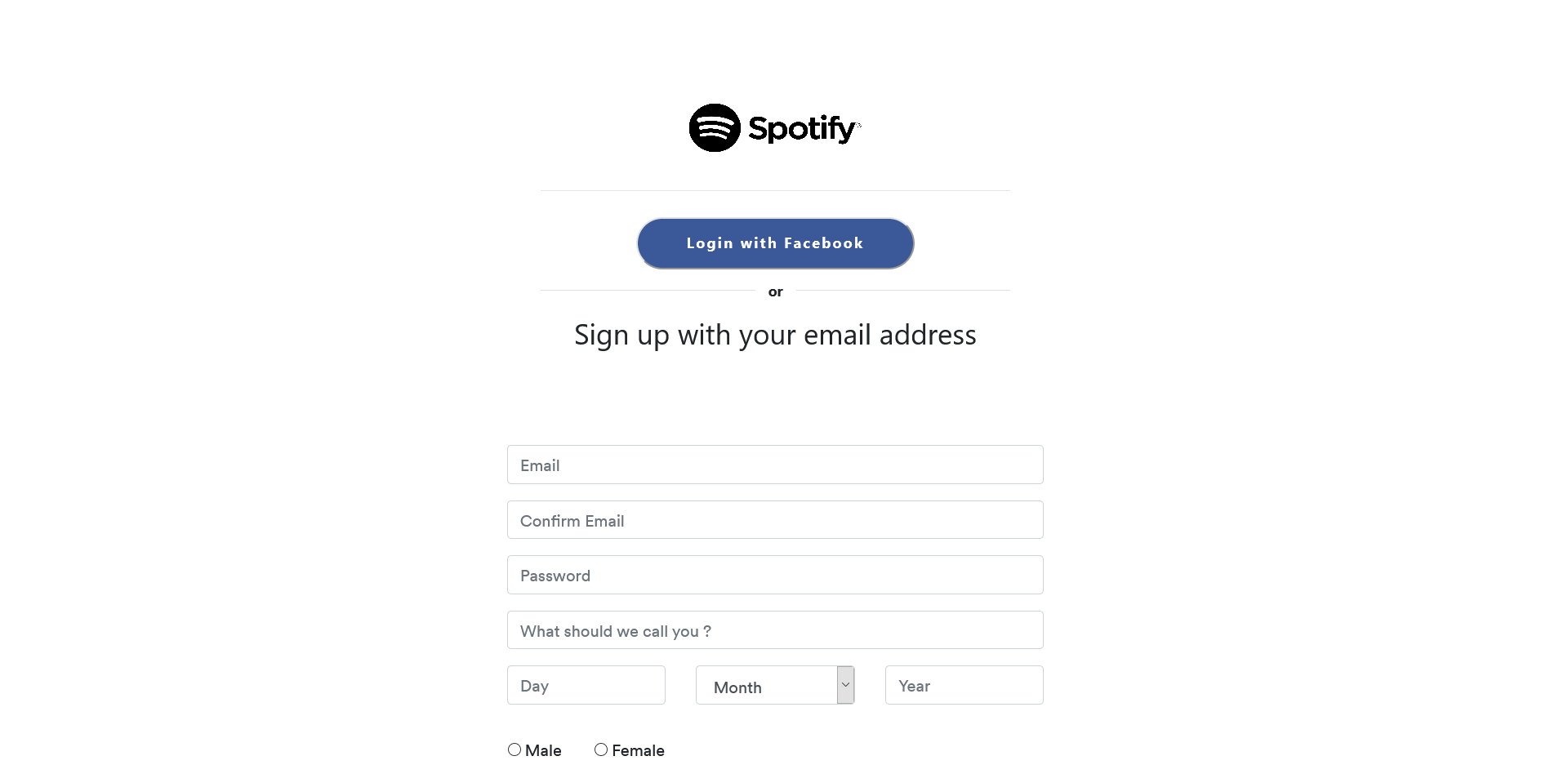
Testing Sign up button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Signup button working



## Test Teardown

[Fixture] setup::0

**Login**

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Sat Apr 11 01:45:51 2020 End: Sat Apr 11 01:47:45 2020 Duration: 0:01:53.799000  broken: 0 (0.00%) failed: 1 (14.29%) skipped: 0 (0.00%) passed: 6 (85.71%) |  |

Test Results

Right-click to update field.

# Refresh Page test - failed

Testing to refresh page that elements are still available

## Details

AssertionError: assert False

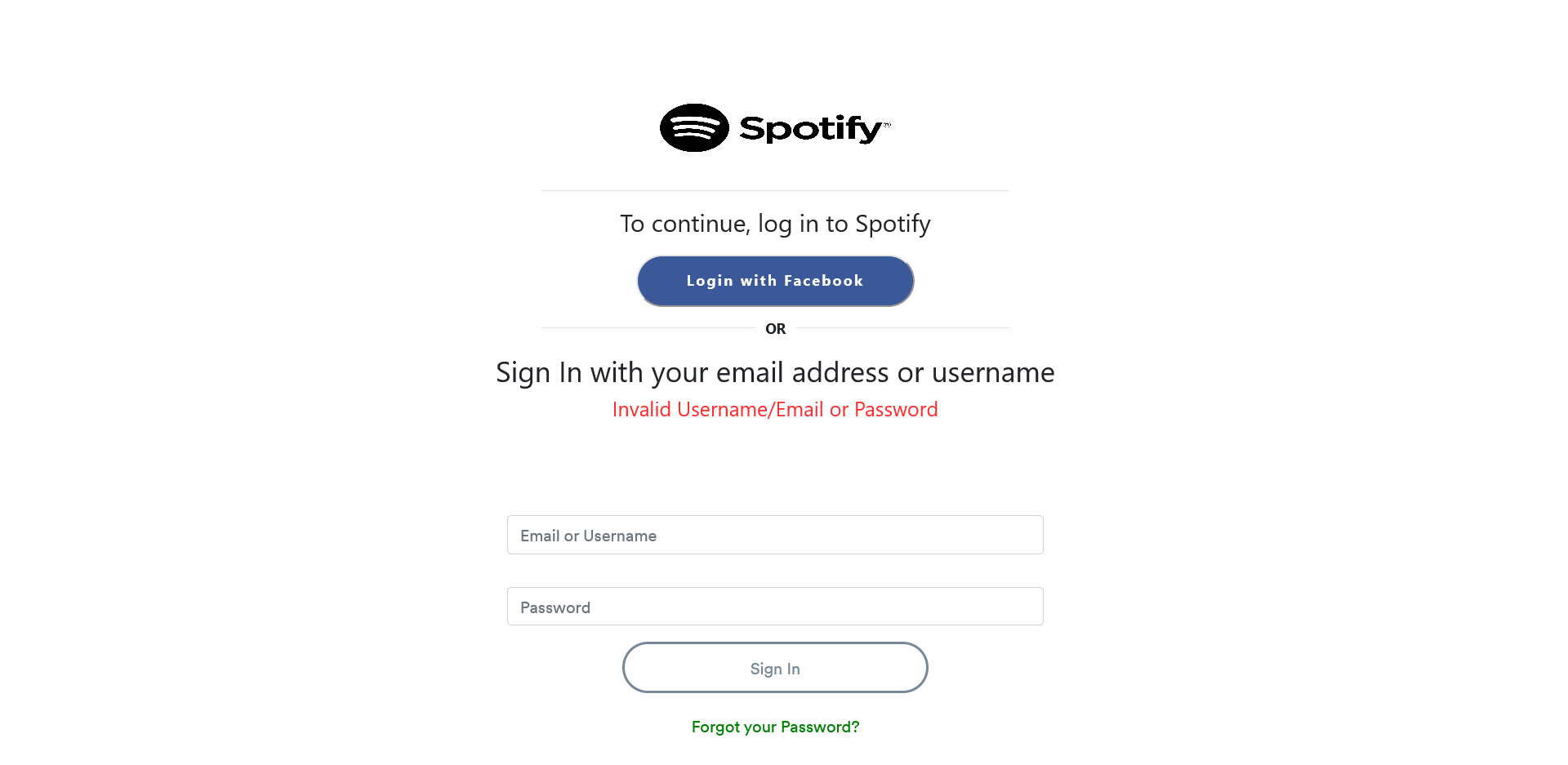
|  |
| --- |
| self = <Web\_Testing.Tests.test\_login.TestLogin object at 0x000001B93FAF66C8>, setup\_final = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Refresh Page test")  @allure.sub\_suite("Refresh Page test")  @allure.title("Refresh Page test")  @allure.description("Testing to refresh page that elements are still available")  @pytest.mark.Do  @pytest.mark.Login  def test\_case\_10(self, setup\_final):  time.sleep(2)  self.driver.refresh()  lp = LoginPage(self.driver)    if (lp.email\_txt is None) or (lp.pass\_txt is None) or (lp.login\_btn is None) or (lp.invalid\_user\_text\_appeared()):  WebHelper().report\_allure(  "ERROR: Elements are not available after refresh OR Invalid text warnings appeared after refresh"  , self.driver) > assert False E assert False  Web\_Testing\Tests\test\_login.py:252: AssertionError |

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] ERROR: Elements are not available after refresh OR Invalid text warnings appeared after refresh



## Test Teardown

[Fixture] setup\_final::0

# Login with correct credentials - passed

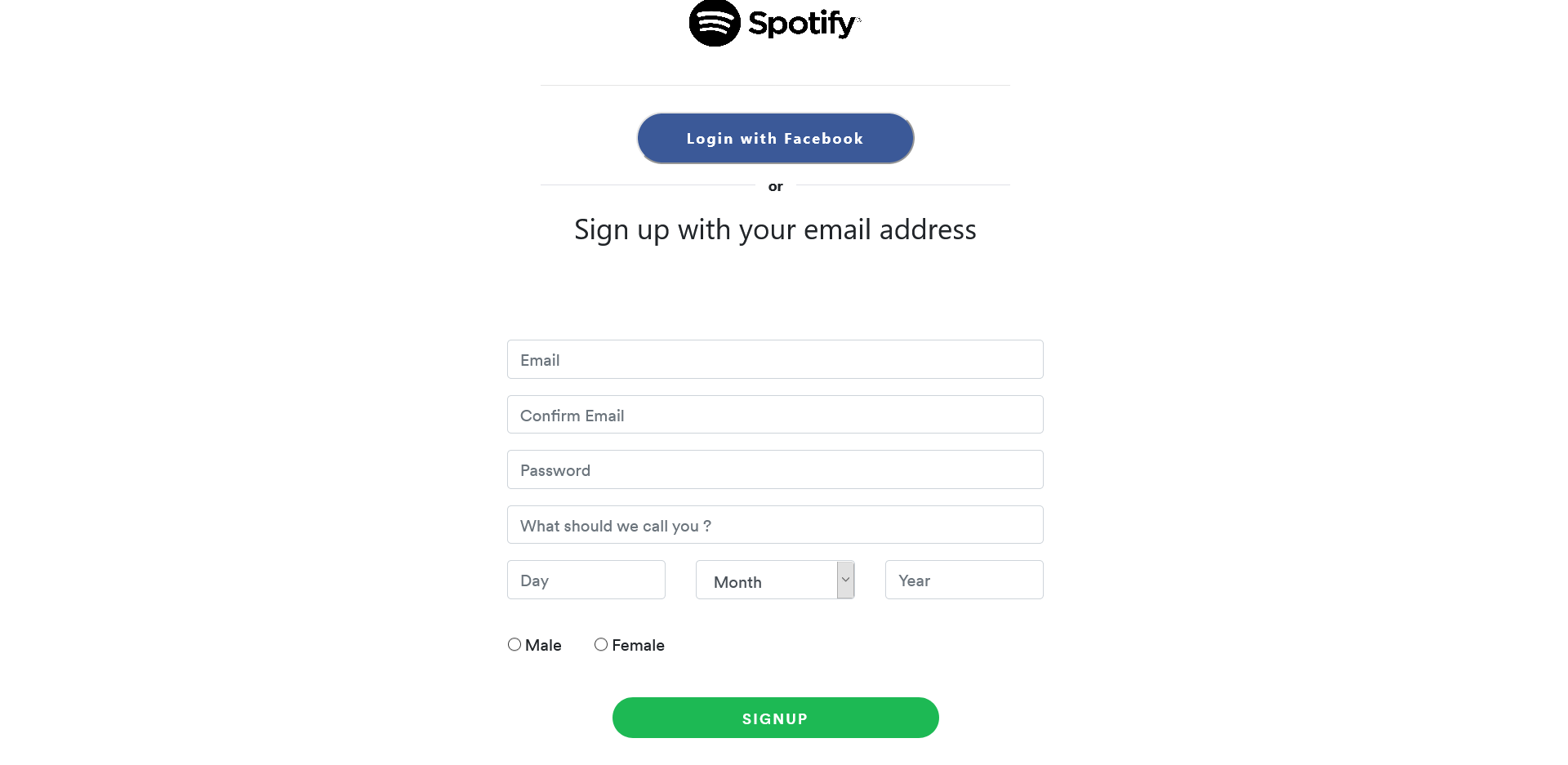
Signing in with the following credentials email : test1@test.com & password: test123

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login succeeded with correct credentials



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Login with empty email - passed

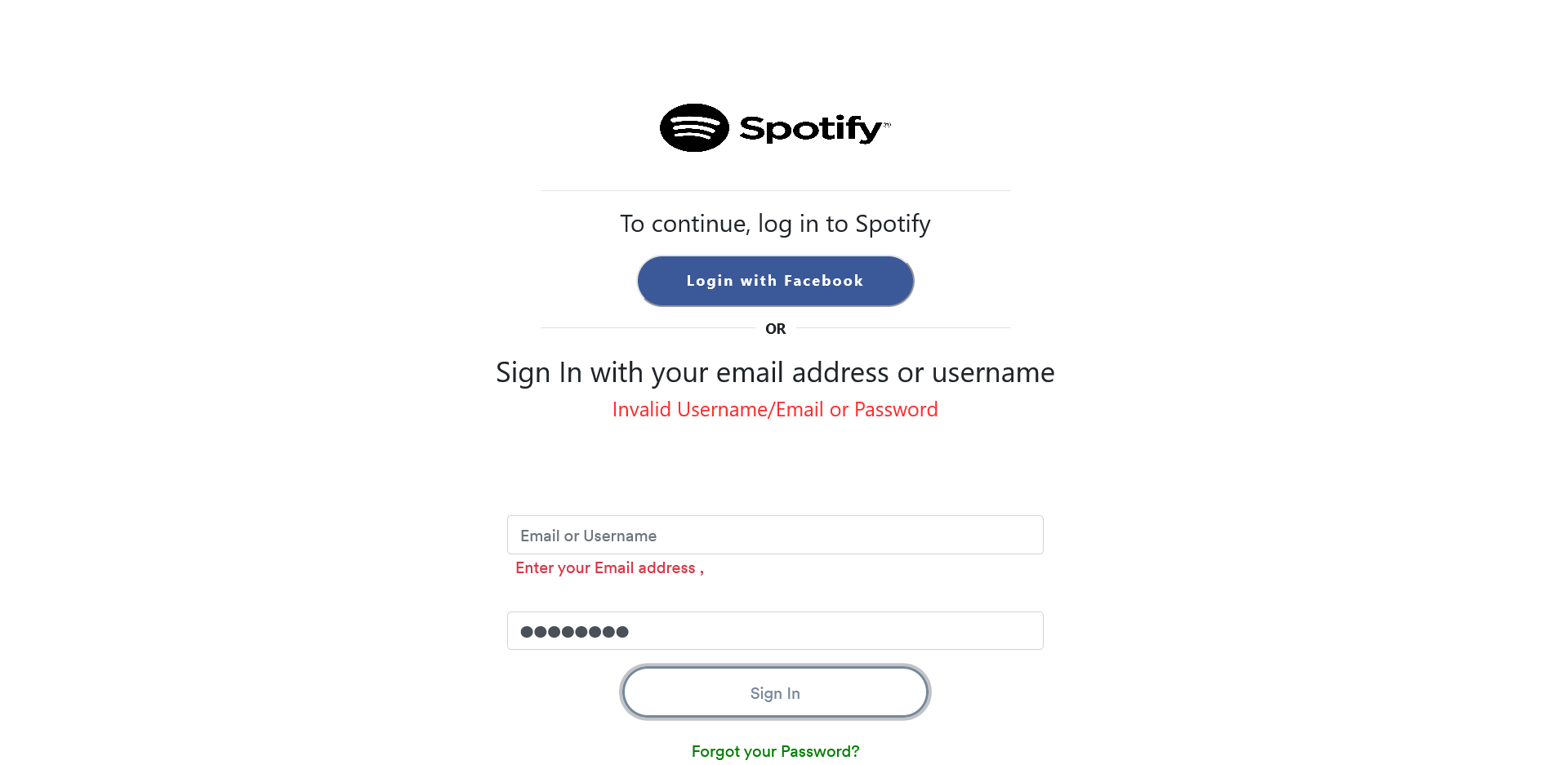
Signing in with empty email

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with empty email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Login with empty email and password - passed

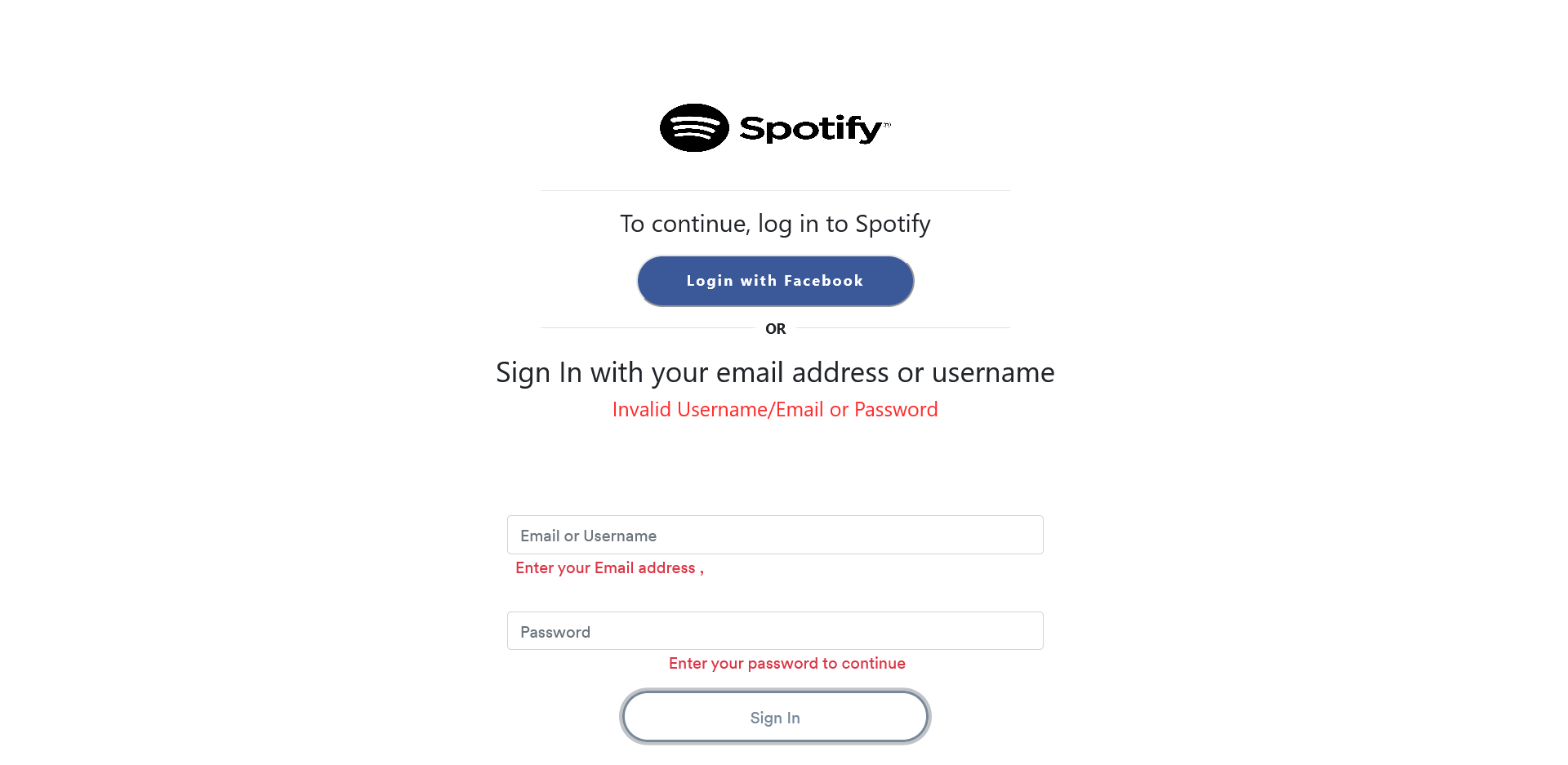
Signing in with empty email and password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with empty email and password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Login with empty password - passed

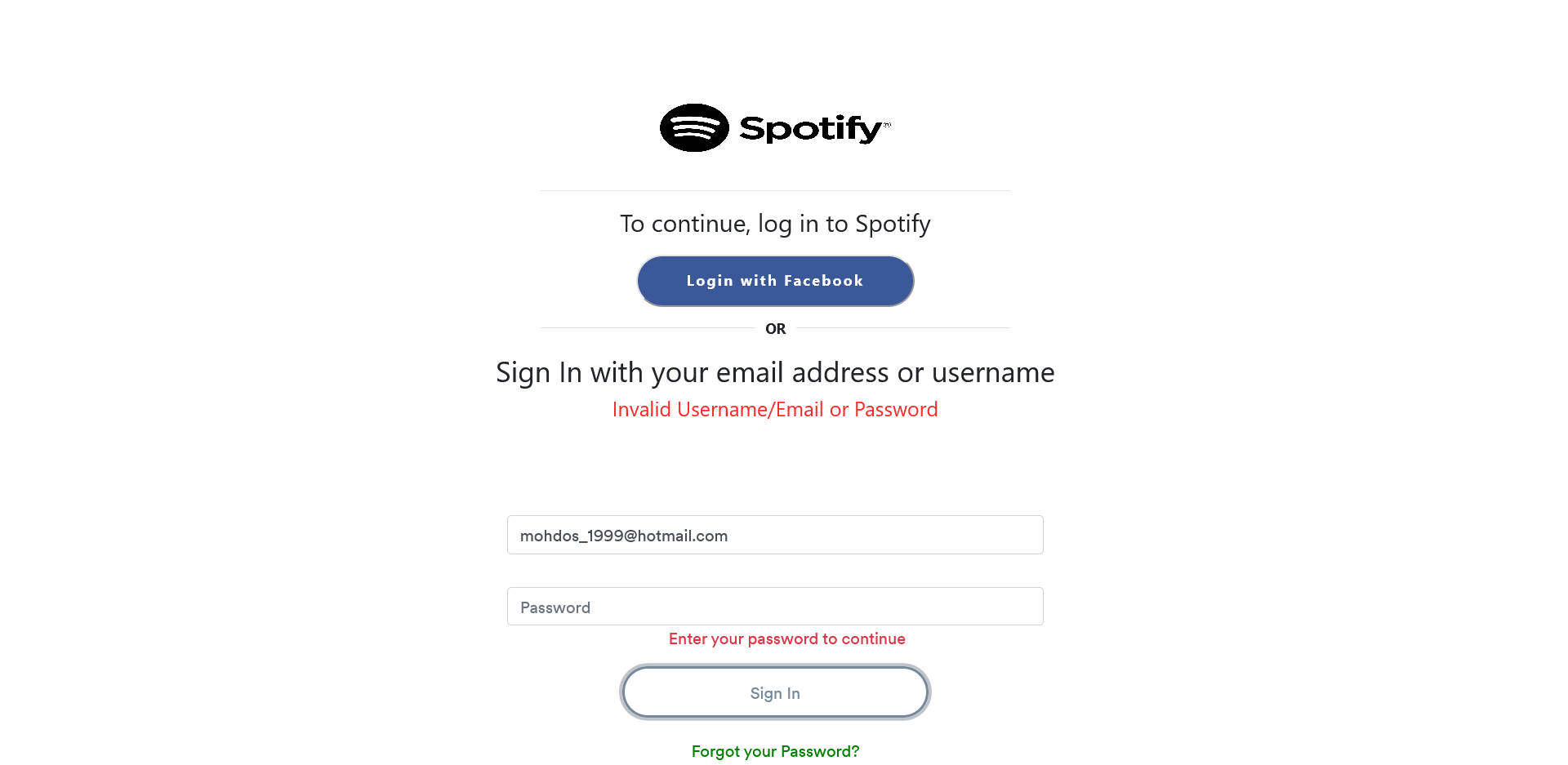
Signing in with empty password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with empty password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Login with wrong email - passed

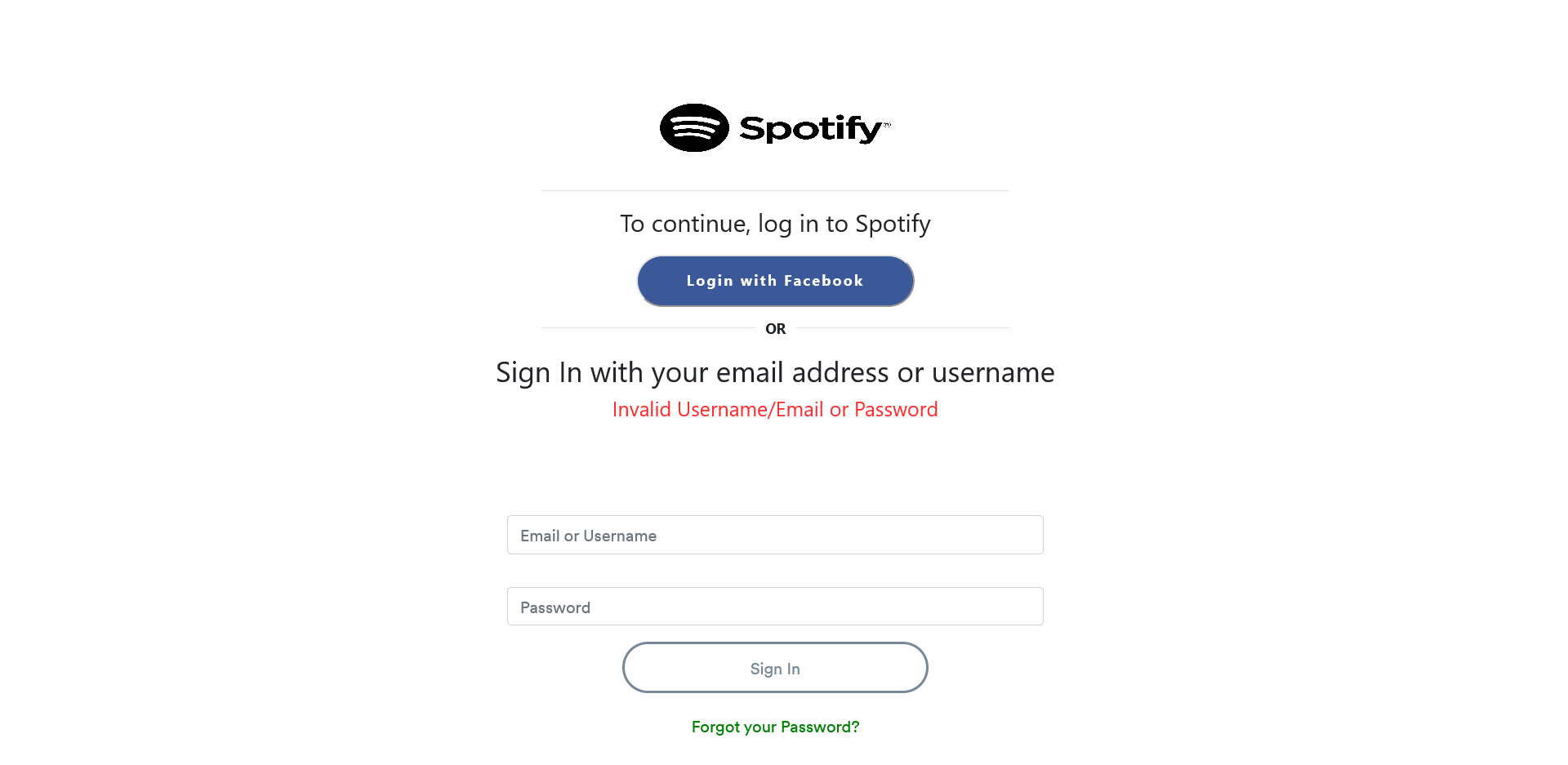
Signing in with the following credentials email : kamelmohsenkamel & password: Kimo2010

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with incorrect credentials



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Login with wrong password - passed

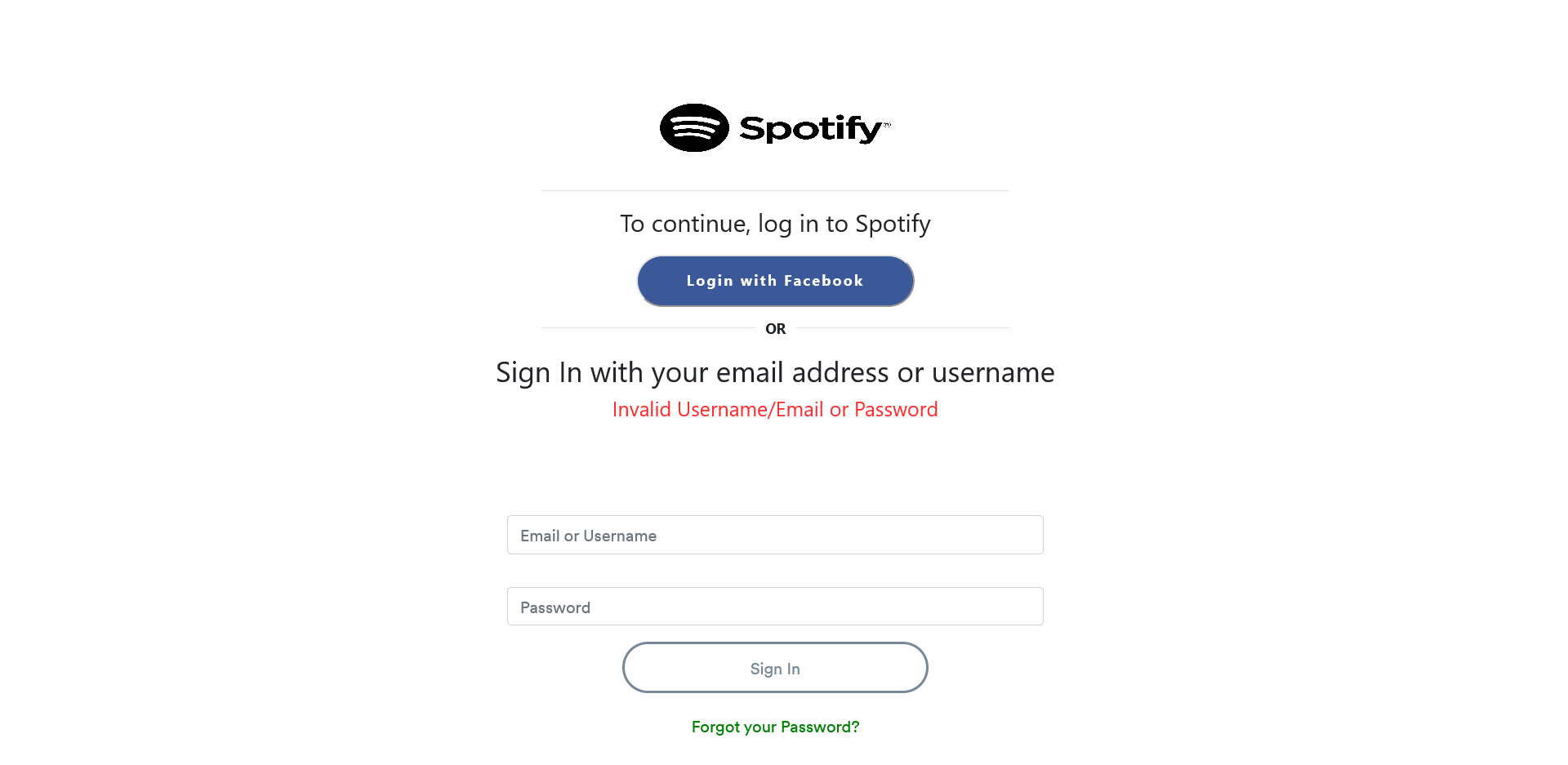
Signing in with the following credentials email : kamelmohsenkamel@gmail.com & password: wrong password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with incorrect credentials



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

**Sign up**

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Fri Apr 10 23:42:57 2020 End: Sat Apr 11 01:40:40 2020 Duration: 1:57:43.673000  broken: 0 (0.00%) failed: 1 (8.33%) skipped: 0 (0.00%) passed: 11 (91.67%) |  |

Test Results

Right-click to update field.

# Refresh Page test - failed

Testing to refresh page that elements are still available

## Details

AssertionError: assert False

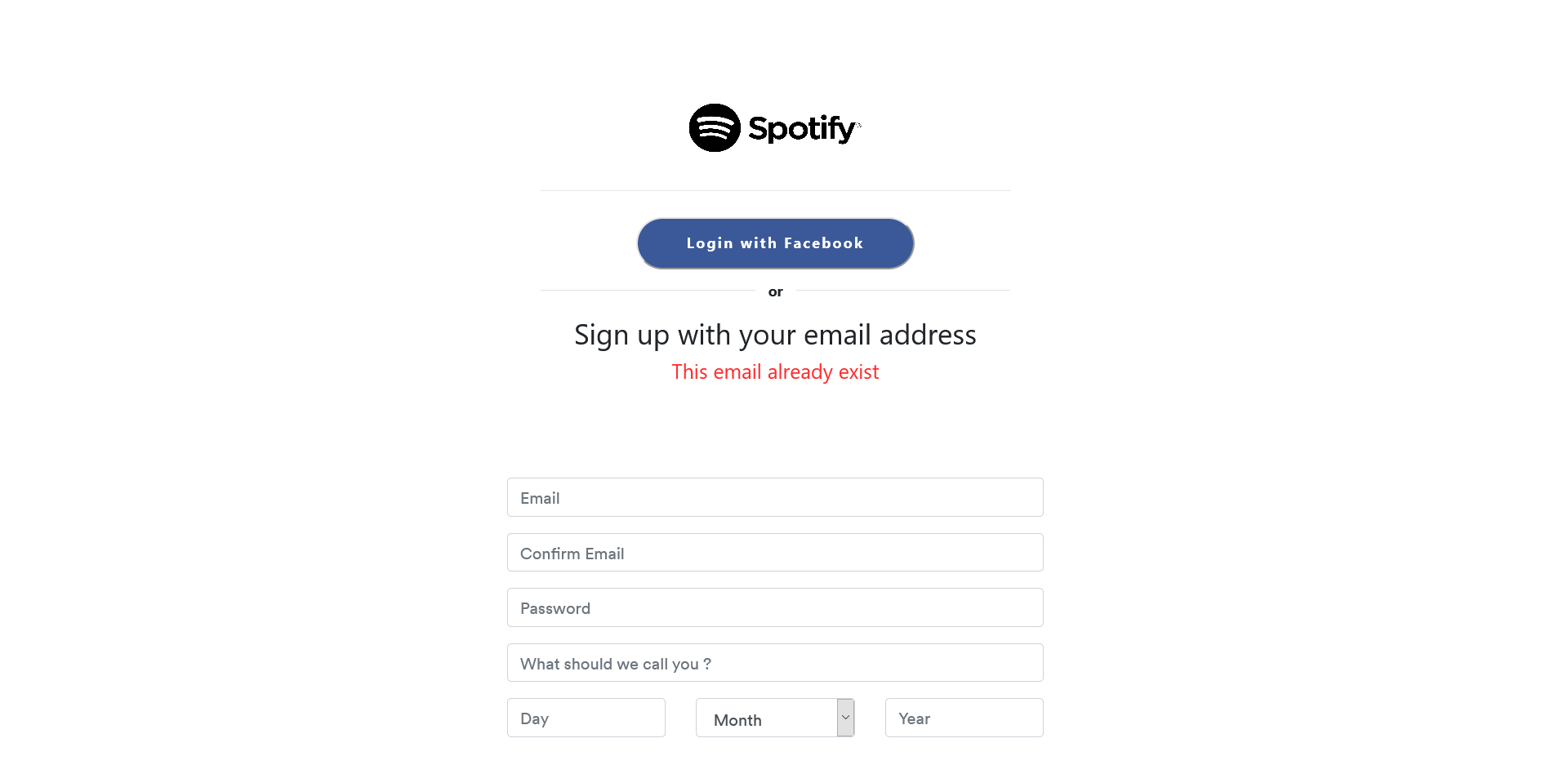
|  |
| --- |
| self = <Web\_Testing.Tests.test\_signup.TestSignup object at 0x00000265679F1F48>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Refresh Page test")  @allure.sub\_suite("Refresh Page test")  @allure.title("Refresh Page test")  @allure.description("Testing to refresh page that elements are still available")  @pytest.mark.Do  @pytest.mark.Signup  def test\_case\_11(self, setup):  time.sleep(2)  self.driver.refresh()  sp = SignupPage(self.driver)    if (sp.email\_txt is None) or (sp.password\_txt is None) or (sp.confirm\_email\_txt is None) \  or (sp.display\_name\_txt is None) or (sp.dob\_day\_txt is None) or (sp.dob\_month\_txt is None)\  or (sp.dob\_year\_txt is None) or (sp.gender\_male is None) or (sp.gender\_female is None)\  or (sp.signup is None) or (sp.is\_text\_dangers\_visible()):  WebHelper().report\_allure(  "ERROR: Some elements are not available after refresh OR Invalid text warnings appeared after refresh"  , self.driver) > assert False E assert False  Web\_Testing\Tests\test\_signup.py:296: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] ERROR: Some elements are not available after refresh OR Invalid text warnings appeared after refresh



## Test Teardown

[Fixture] setup::0

# Invalid age test - passed

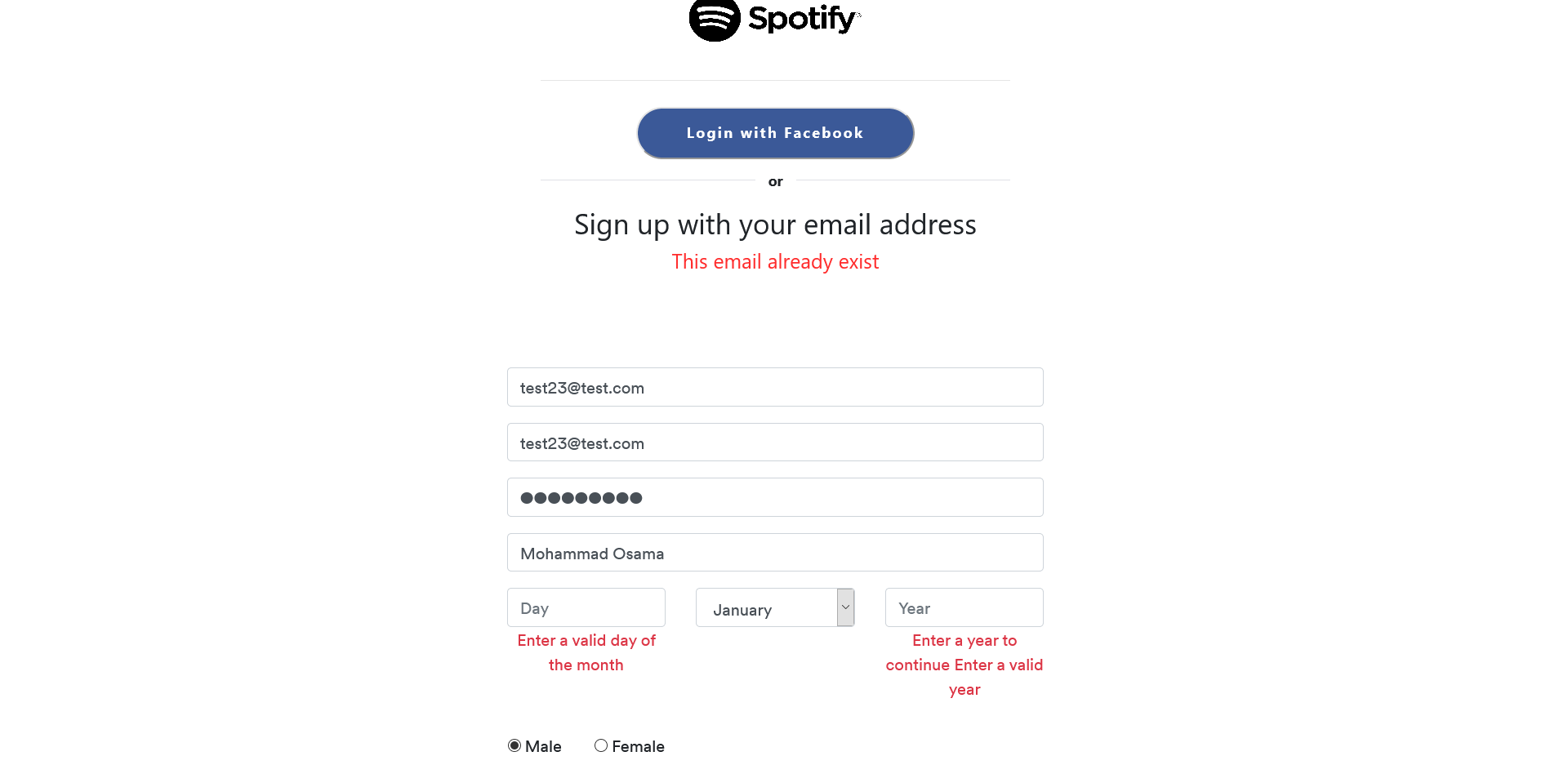
Testing with entering invalid age values

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] SUCCESS: Sign up stopped with invalid age values



[Attachment] stdout

## Test Teardown

[Fixture] setup\_final::0

# Sign up with correct credentials and values - passed

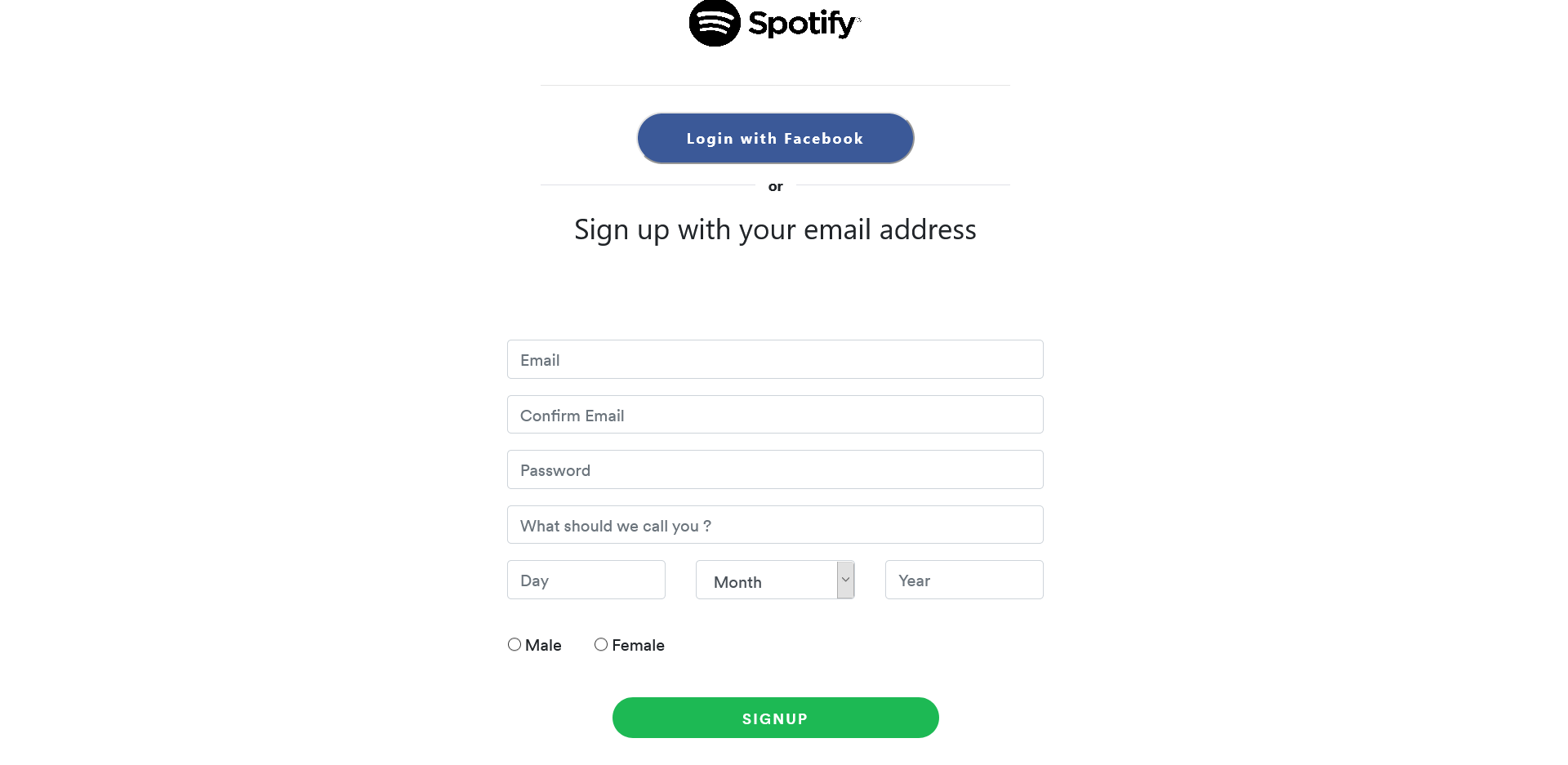
Signing up with the following credentials and values -> Email: test1@test.com, Pass: test123

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up with correct credentials and values proceeded



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Sign up with different confirmation email - passed

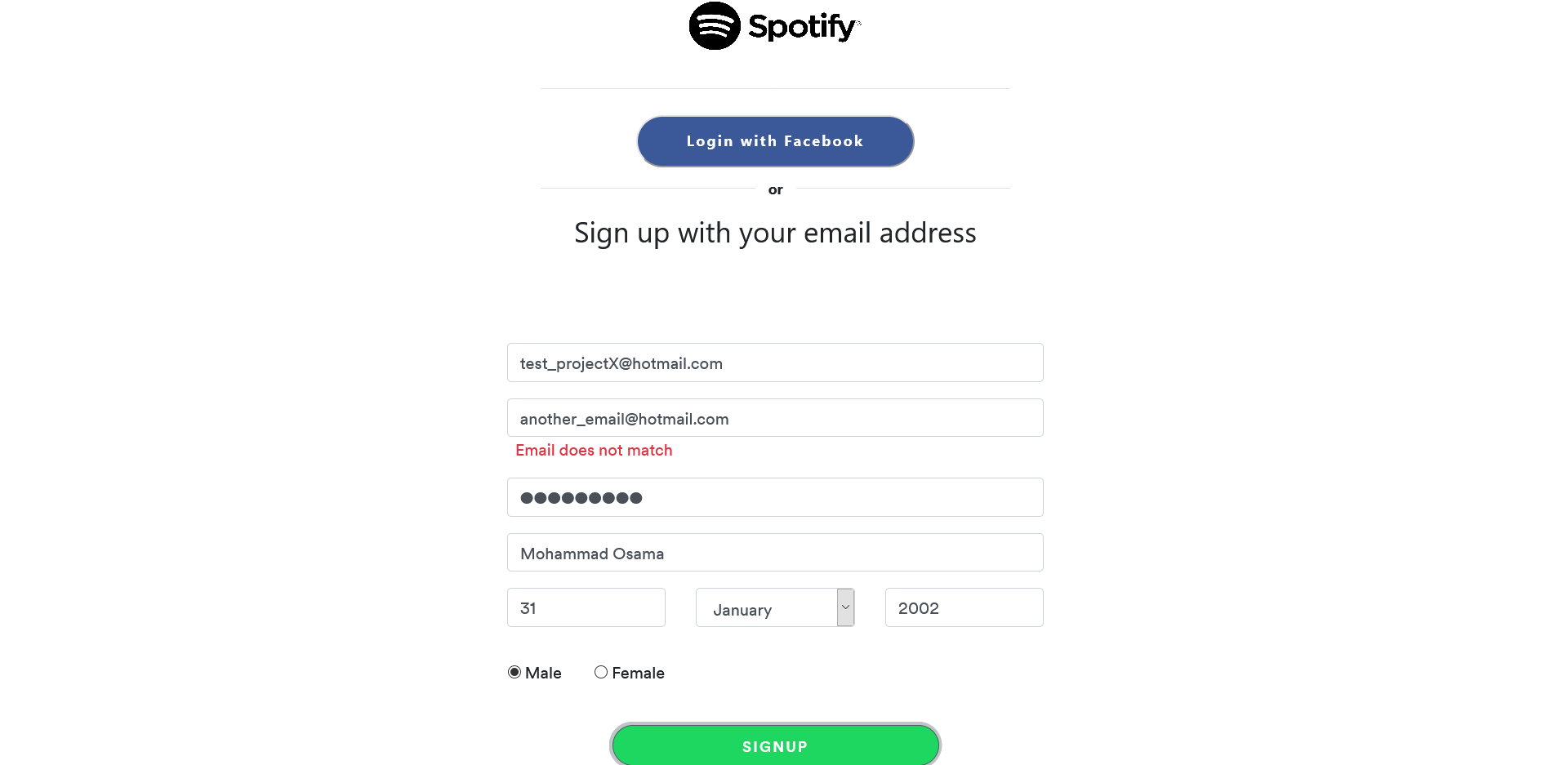
Signing up with different confirmation email from the original email field

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up unsuccessful with different confirmation email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with already registered email - passed

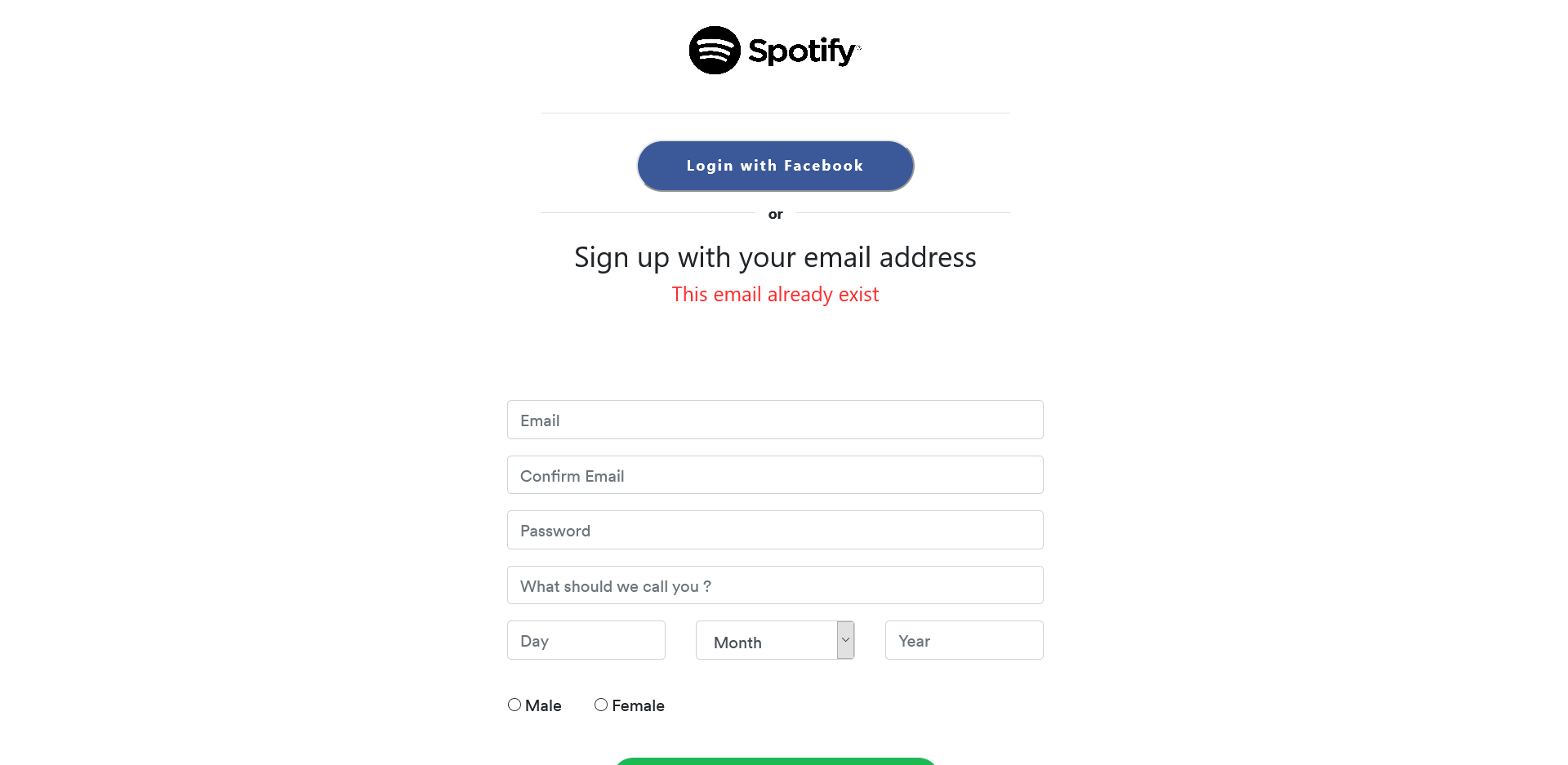
Signing up with the following credentials email: test\_projectX@hotmail.com

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with already registered email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with empty confirmation email - passed

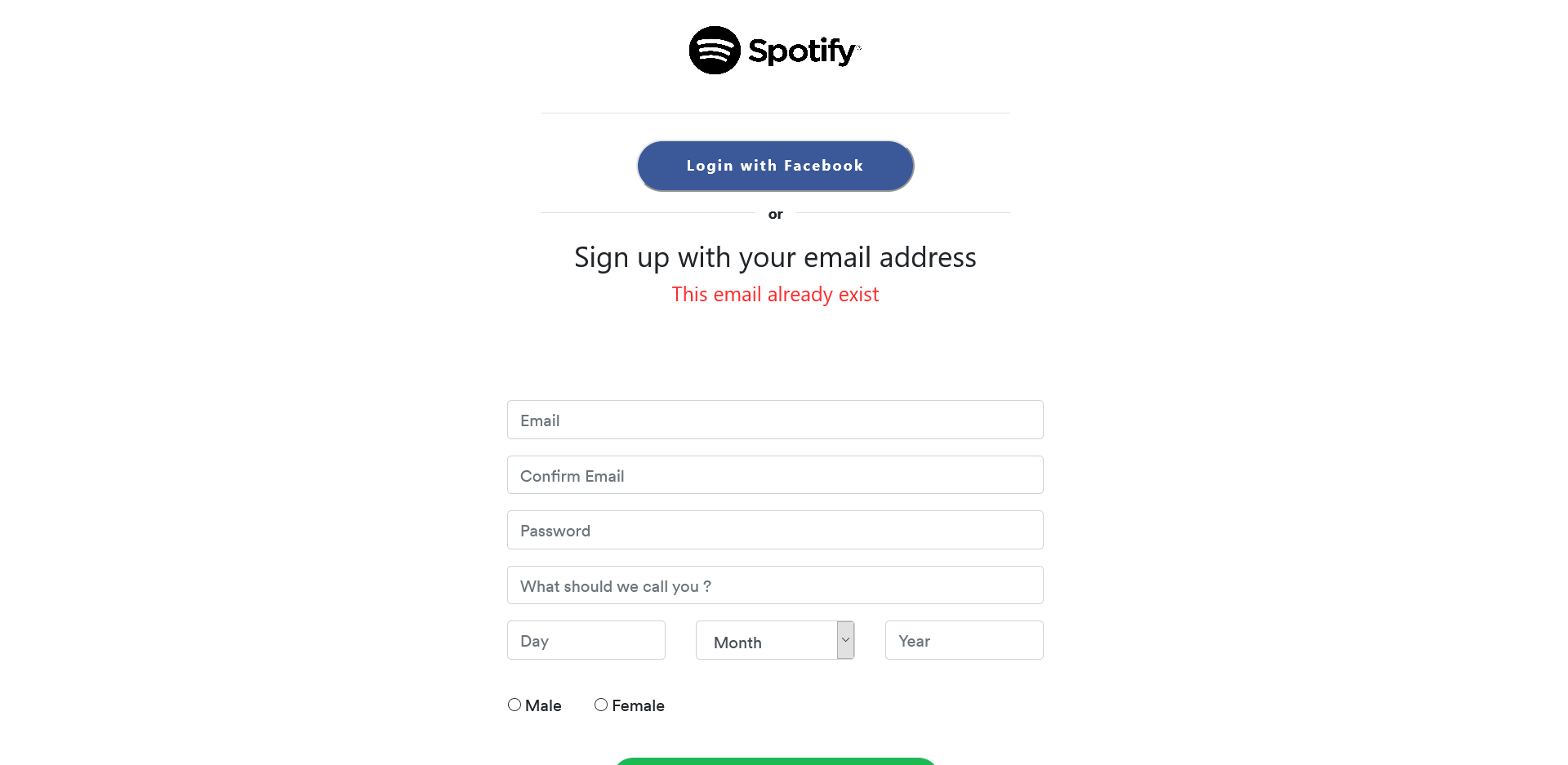
Signing up with empty confirmation email

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with empty confirmation email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with empty email - passed

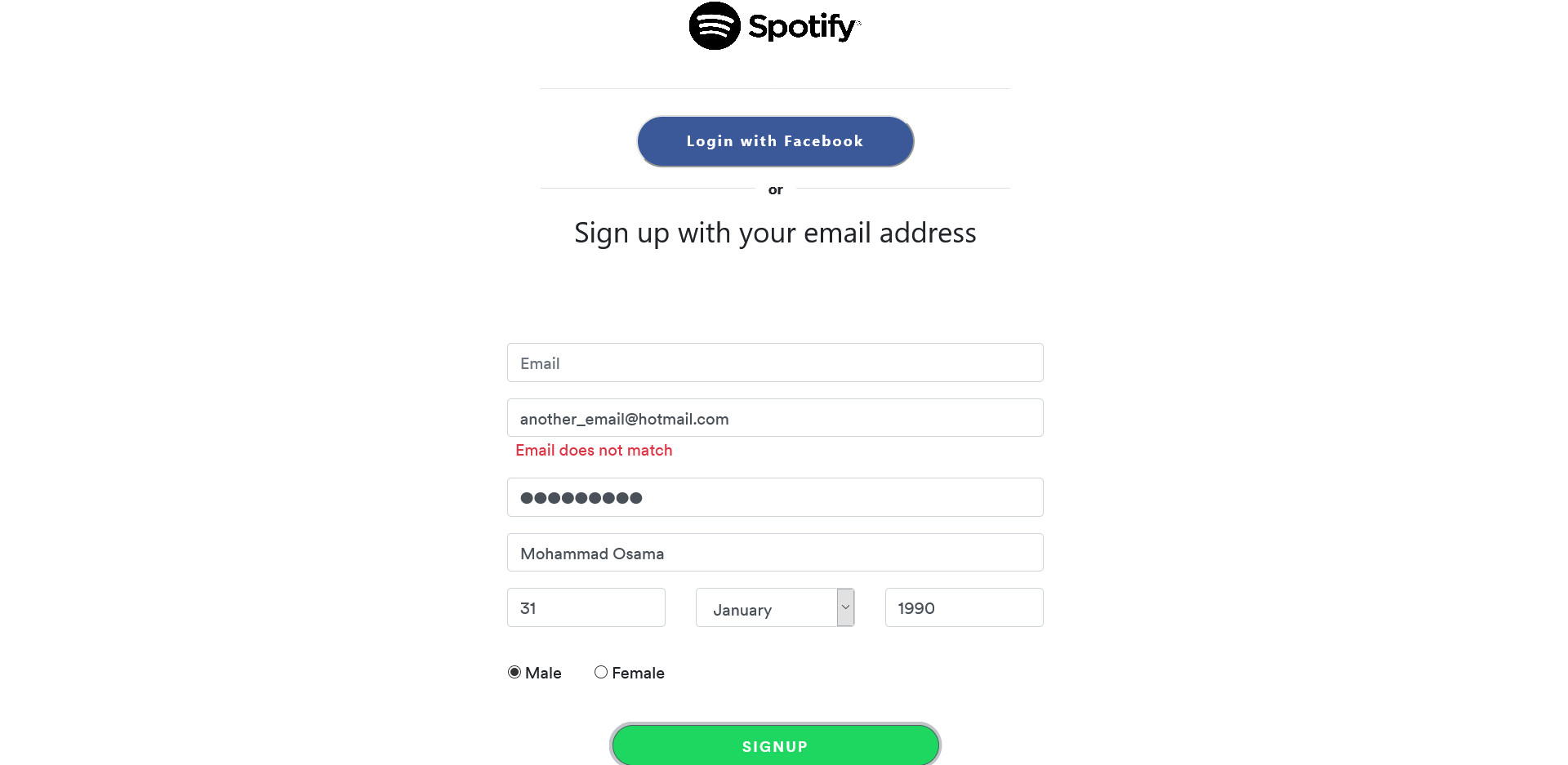
Signing up with empty email

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with empty email



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with empty password - passed

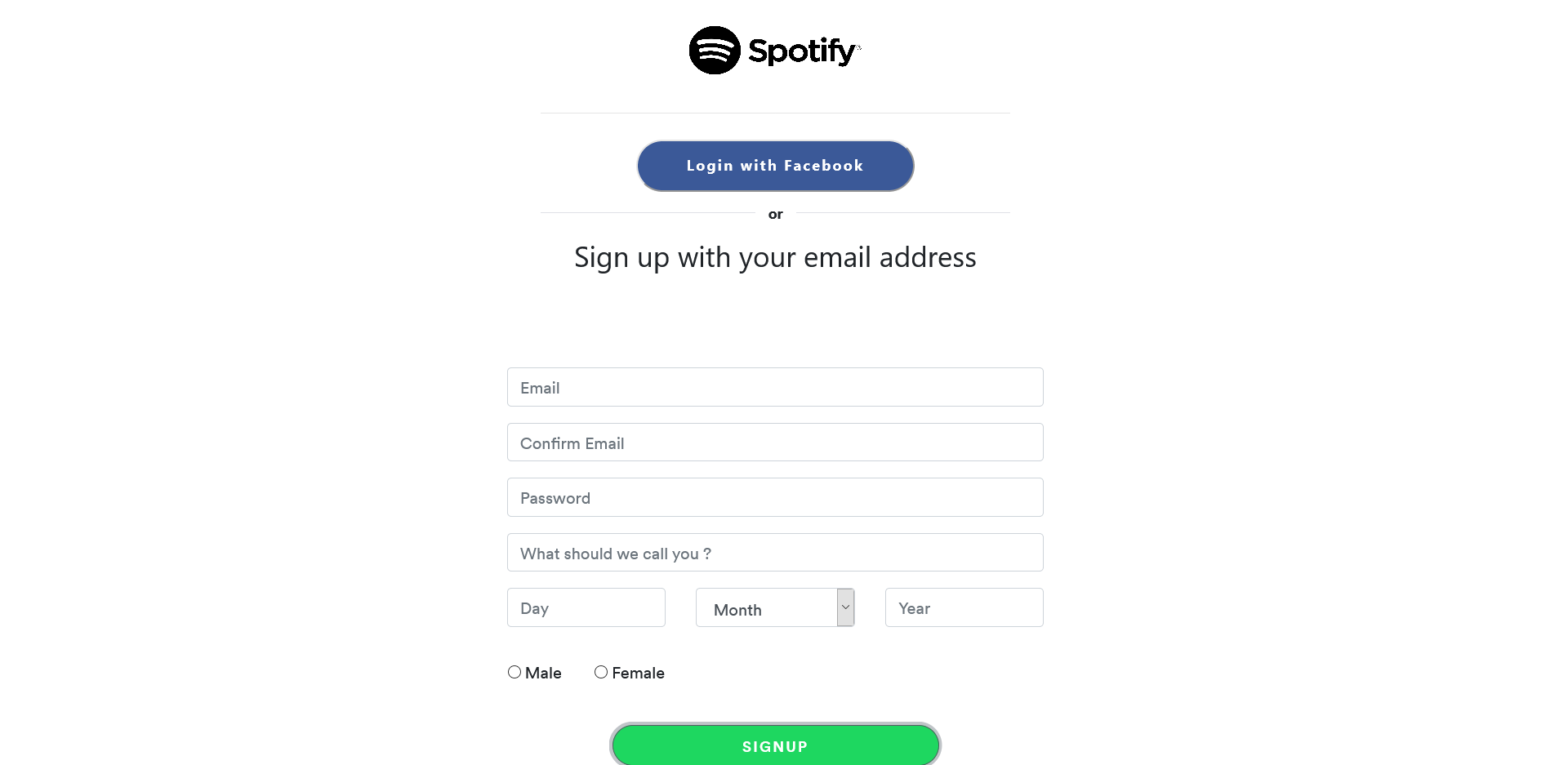
Signing up with empty password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with empty password



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with unselected day - passed

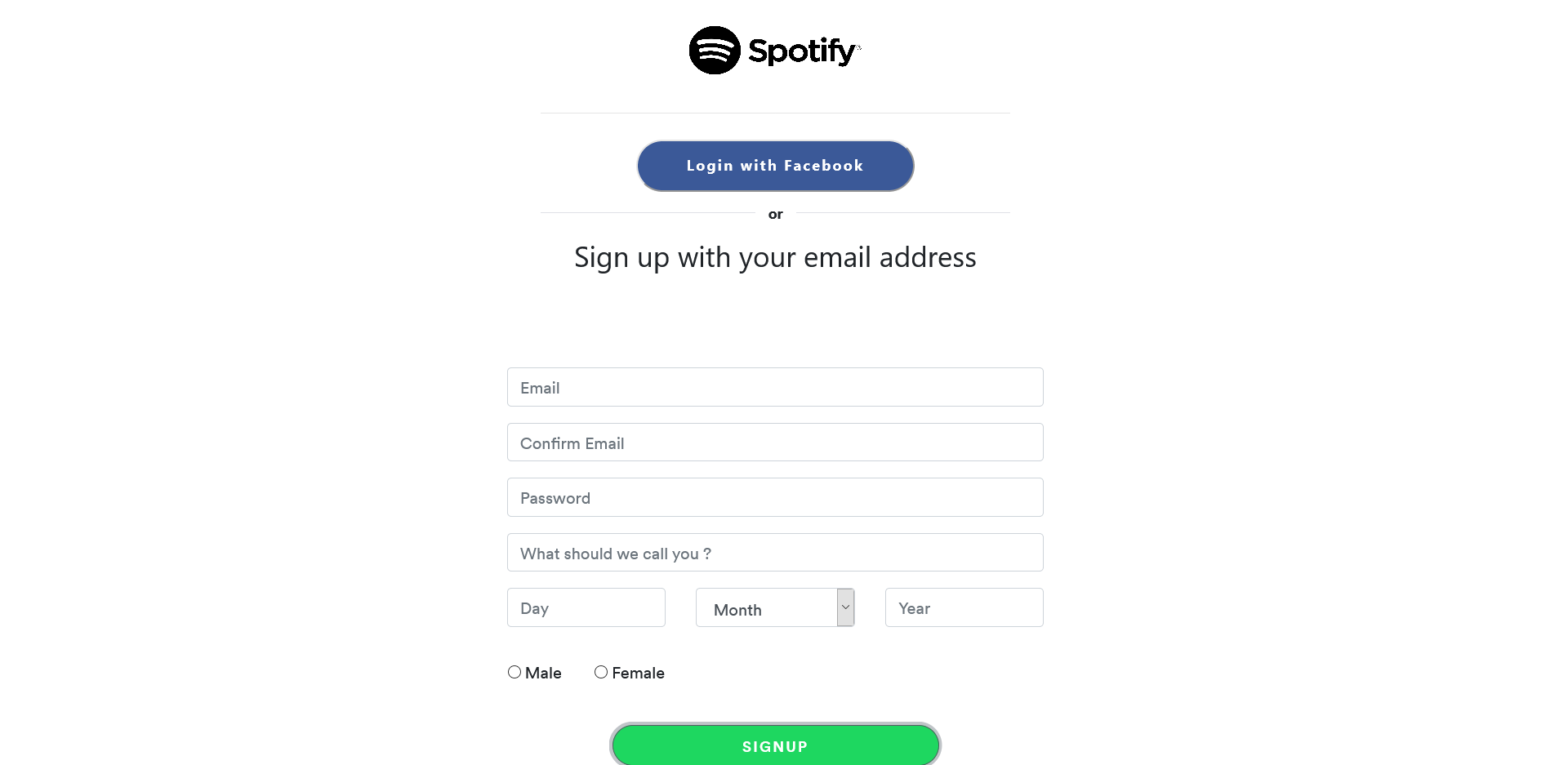
Signing up with unselected day

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected day



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with unselected gender - passed

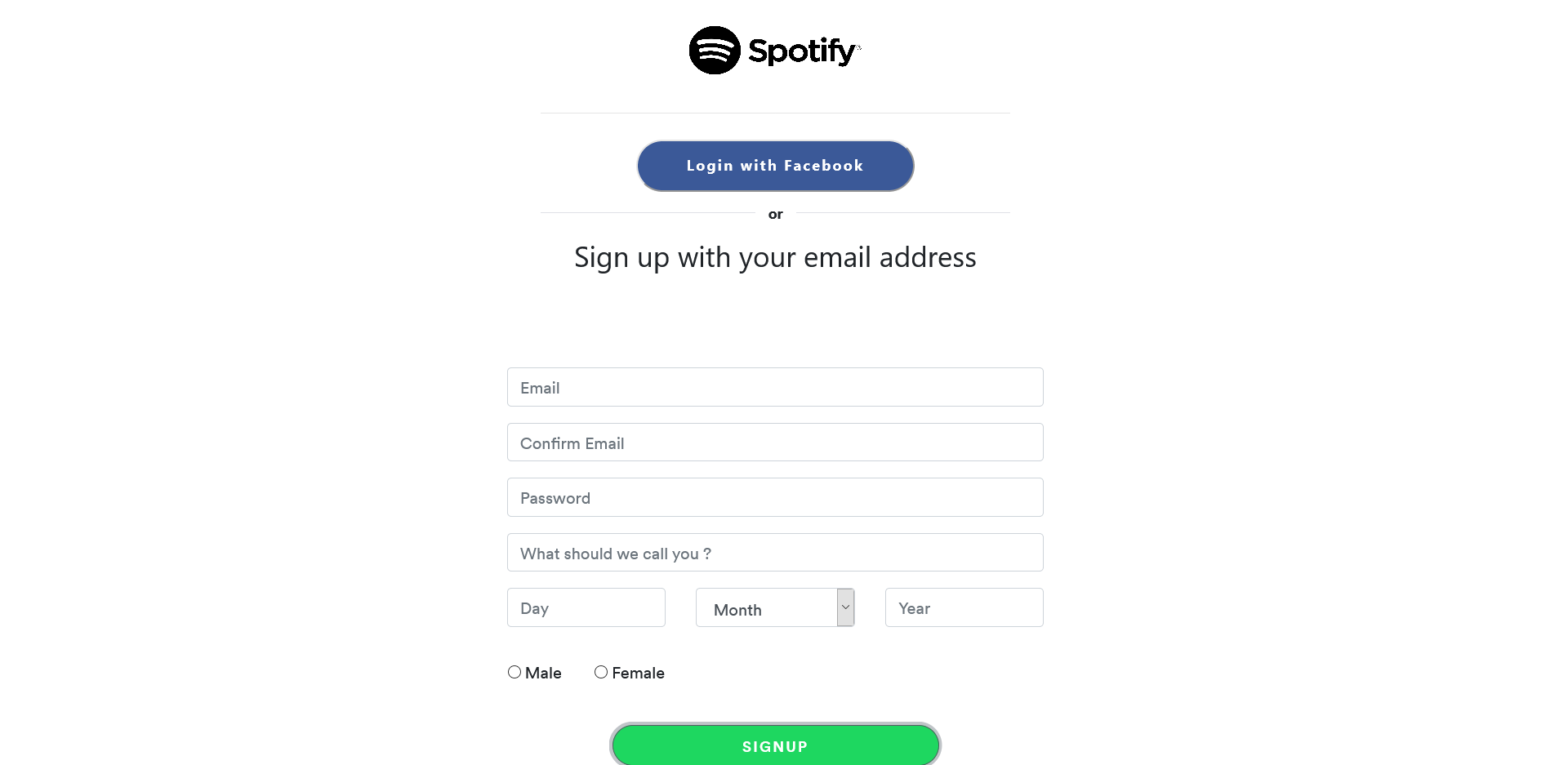
Signing up with unselected gender

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected gender



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with unselected month - passed

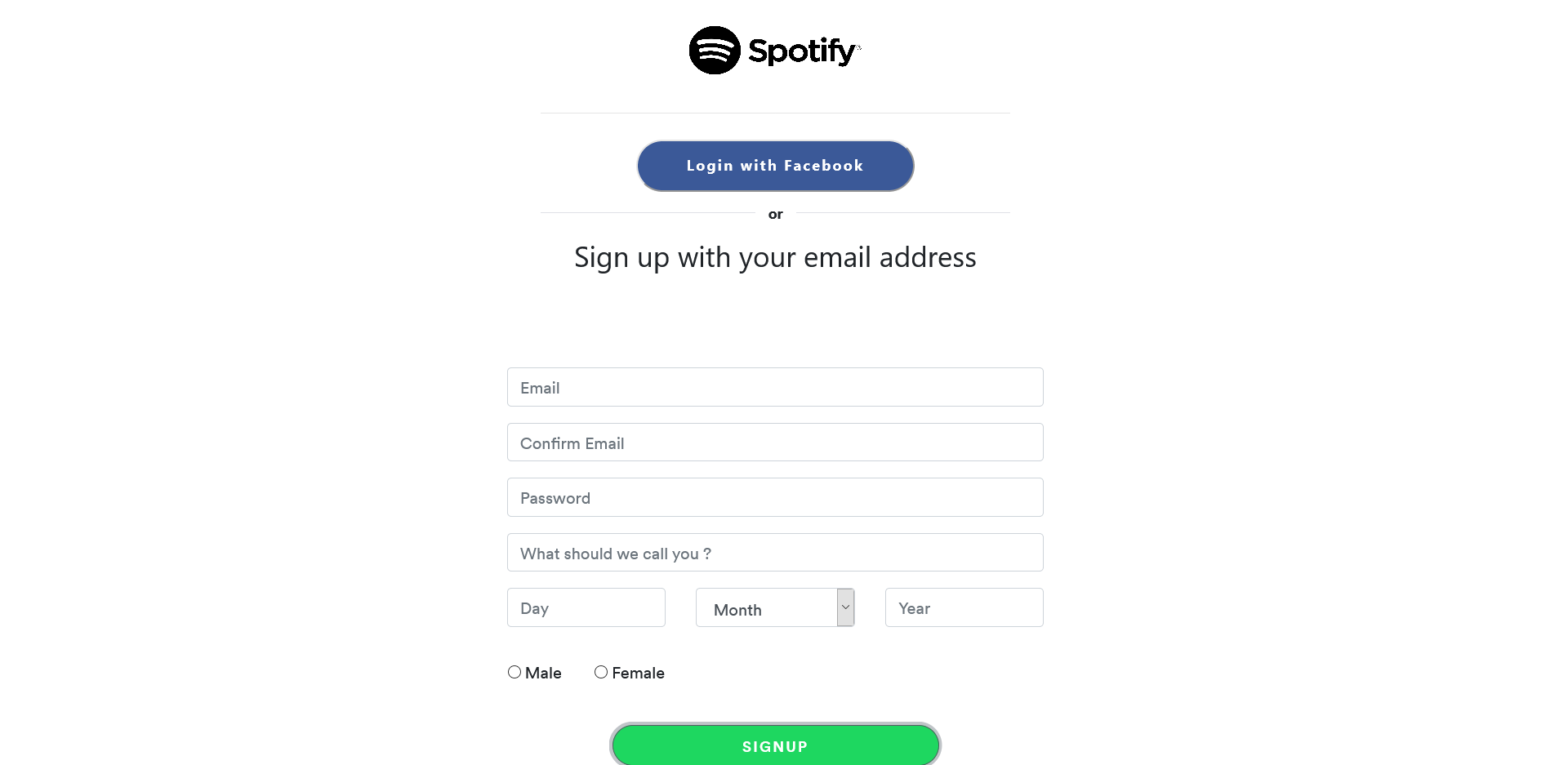
Signing up with unselected month

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected month



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Signup with unselected year - passed

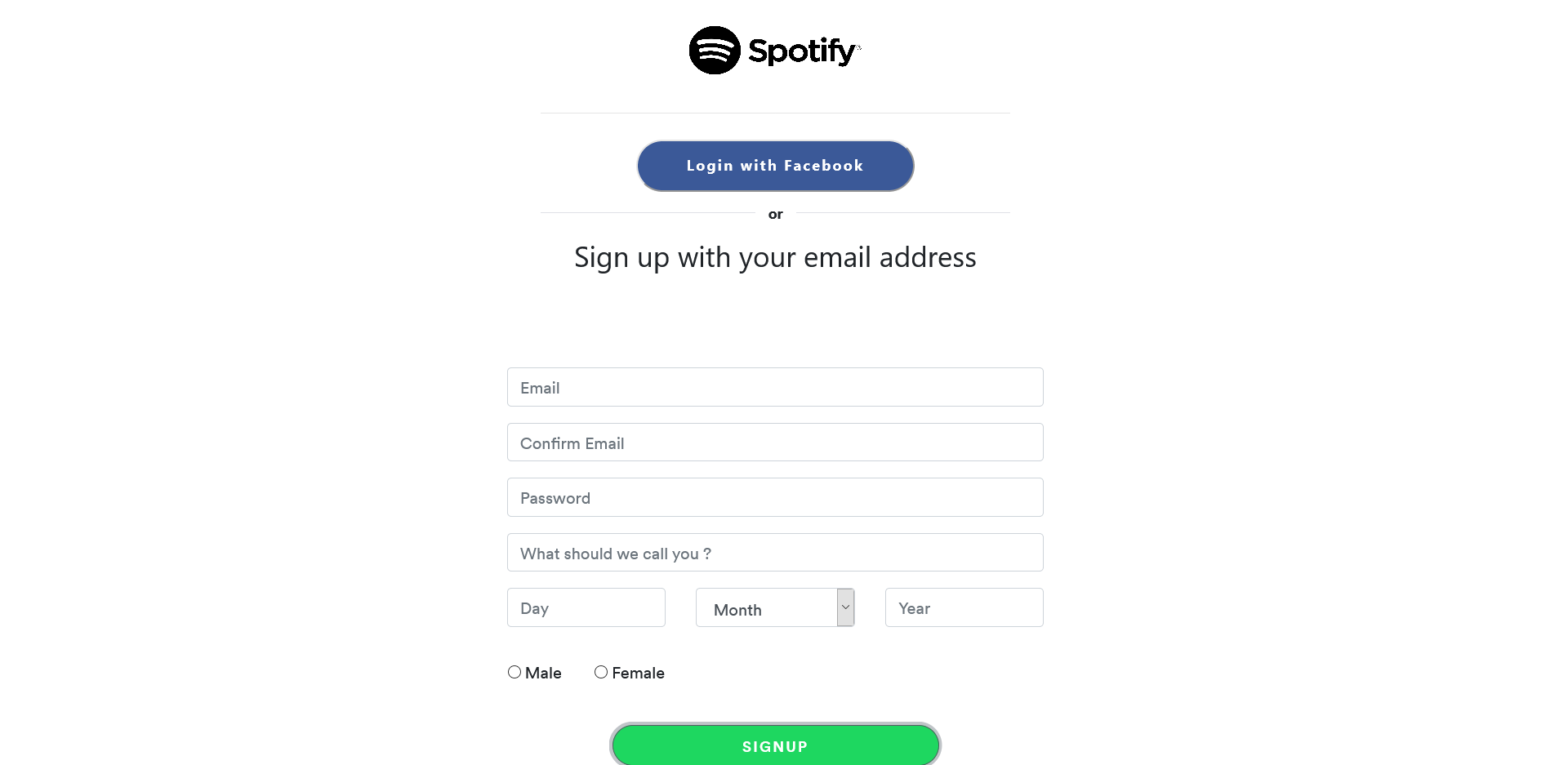
Signing up with unselected year

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected year



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

**Web Player Home**

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Sat Apr 11 02:06:50 2020 End: Sat Apr 11 02:08:22 2020 Duration: 0:01:32.225000  broken: 0 (0.00%) failed: 0 (0.00%) skipped: 0 (0.00%) passed: 9 (100.00%) |  |

Test Results

Right-click to update field.

# Albums cards - passed

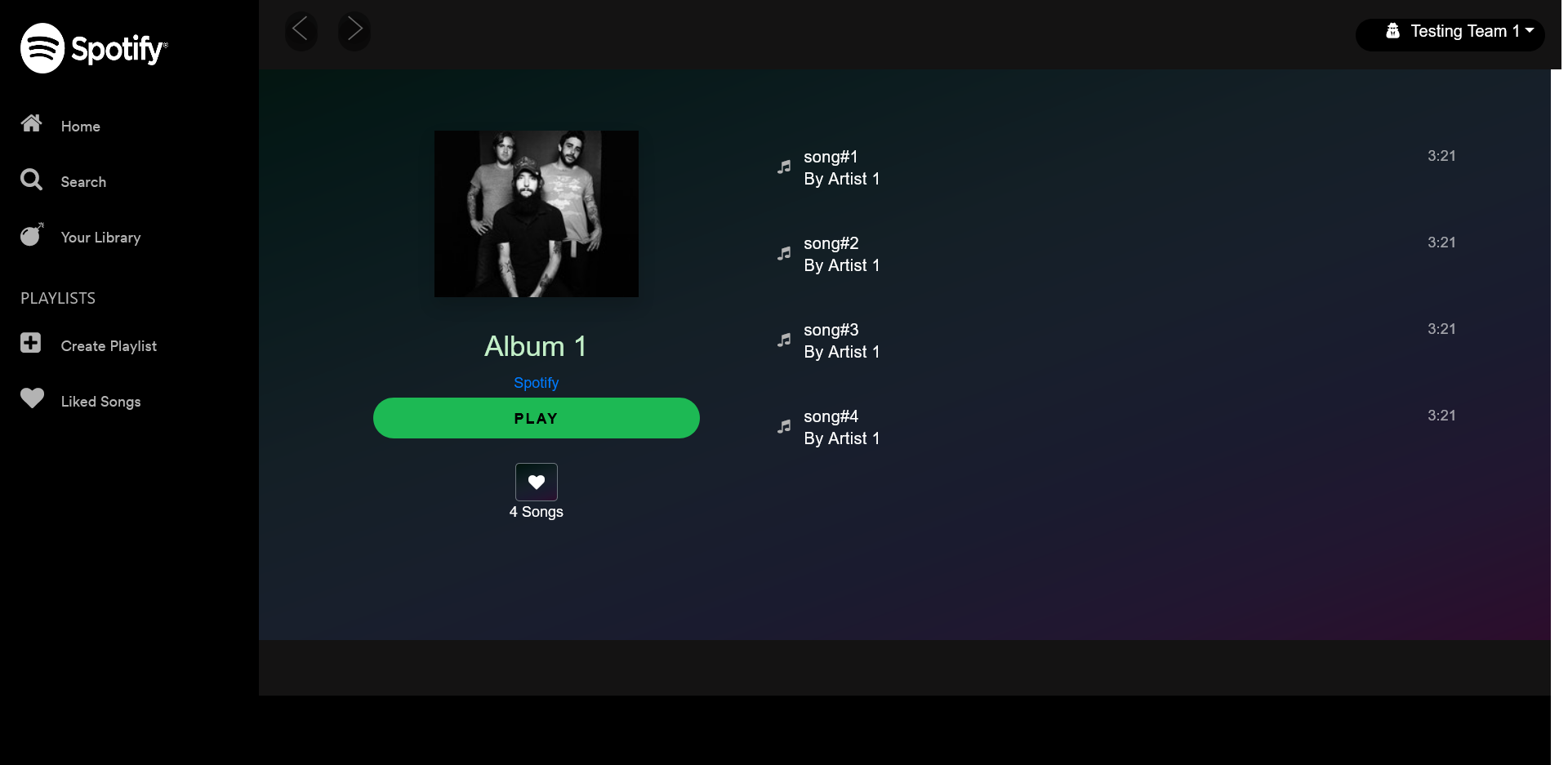
Testing Albums cards

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Albums cards are functional



## Test Teardown

[Fixture] setup::0

# Artist cards button - passed

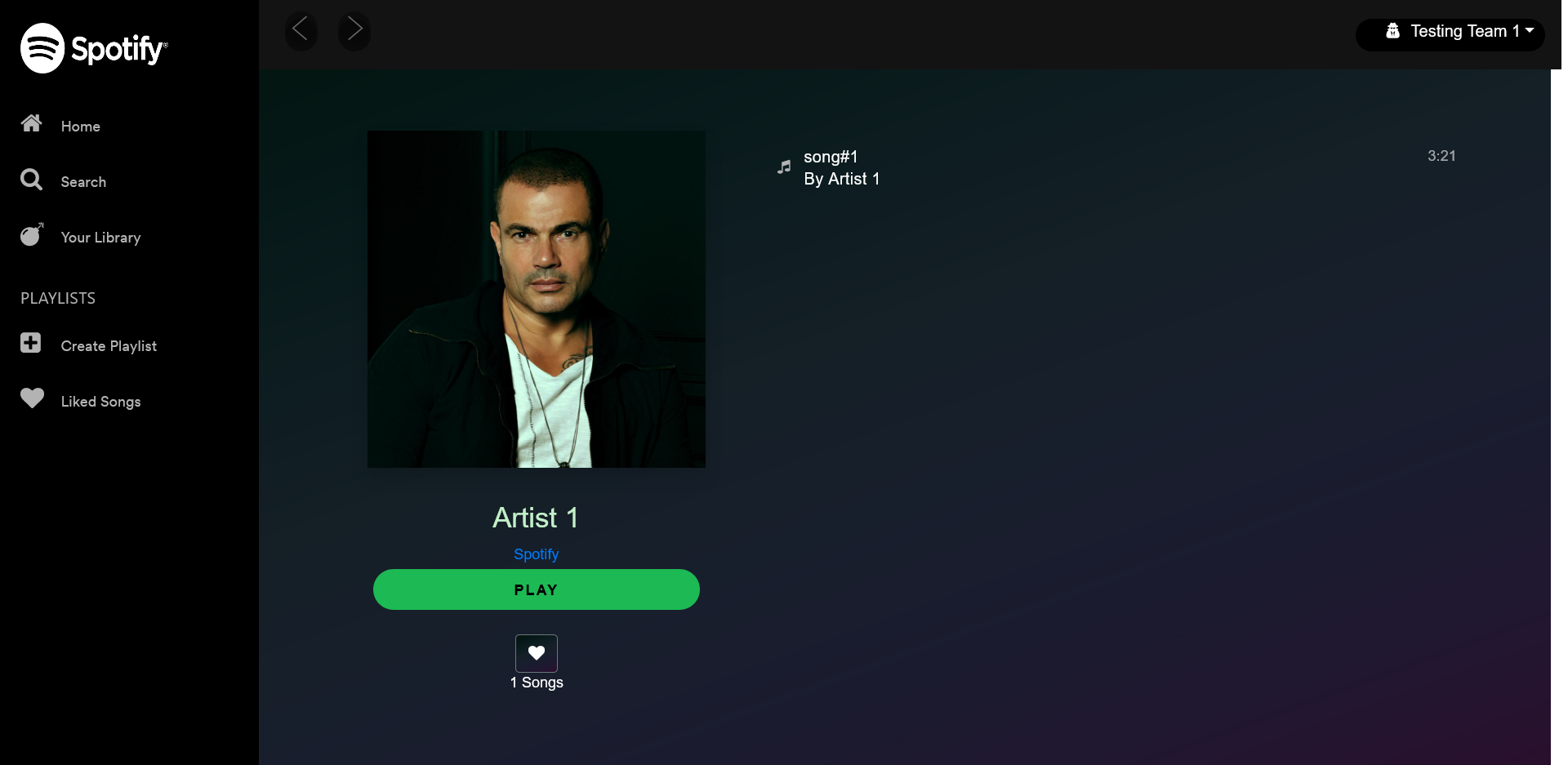
Testing Artists cards Button

## Test Setup

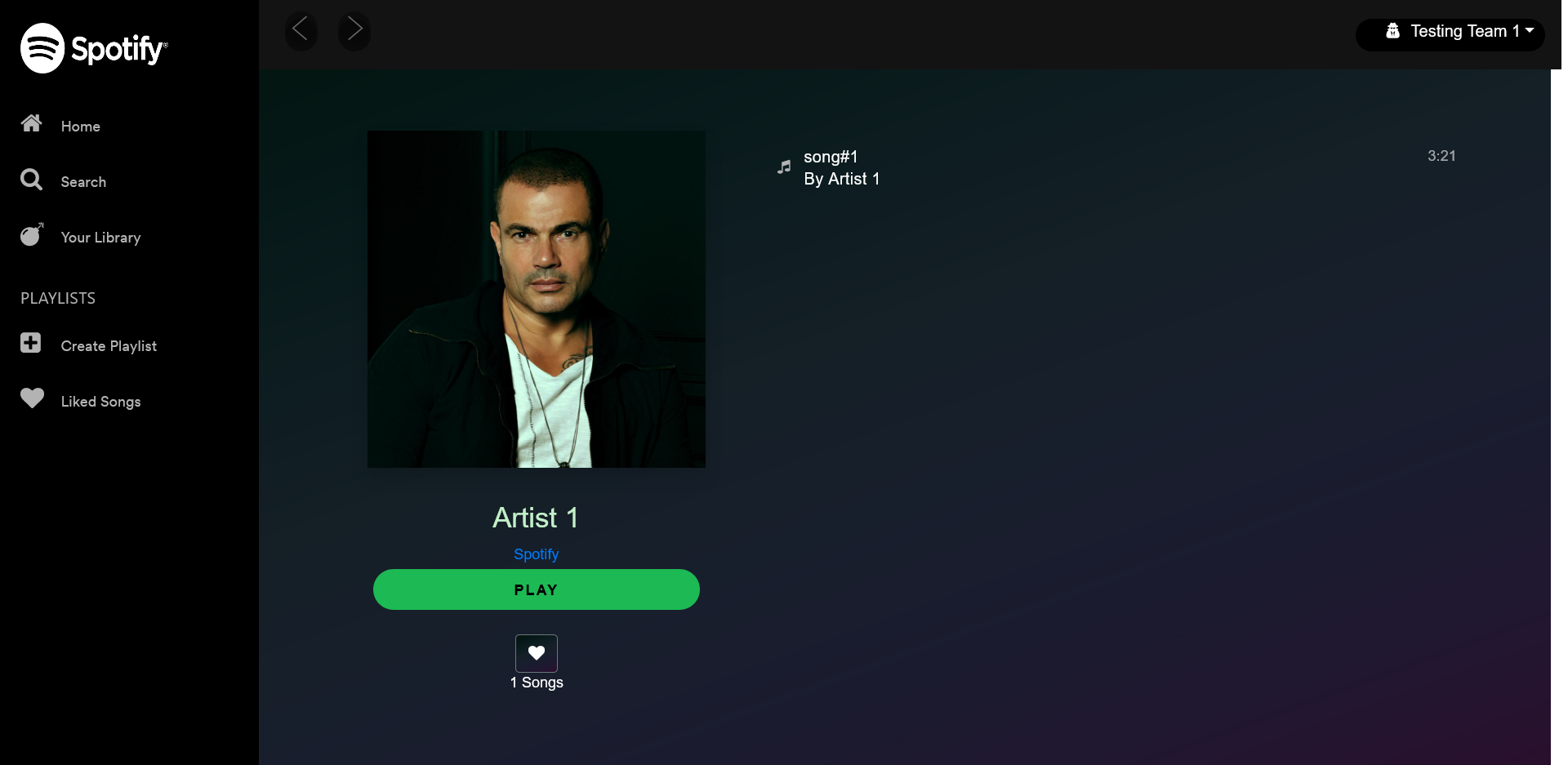
[Fixture] setup

## Test Body

[Attachment] ERROR: Playlist name is not the same as the card name. Card name = Artist 1  
Artist, Playlist name = Artist 1



[Attachment] SUCCESS: Artists cards are functional



## Test Teardown

[Fixture] setup::0

# Liked Songs Button - passed

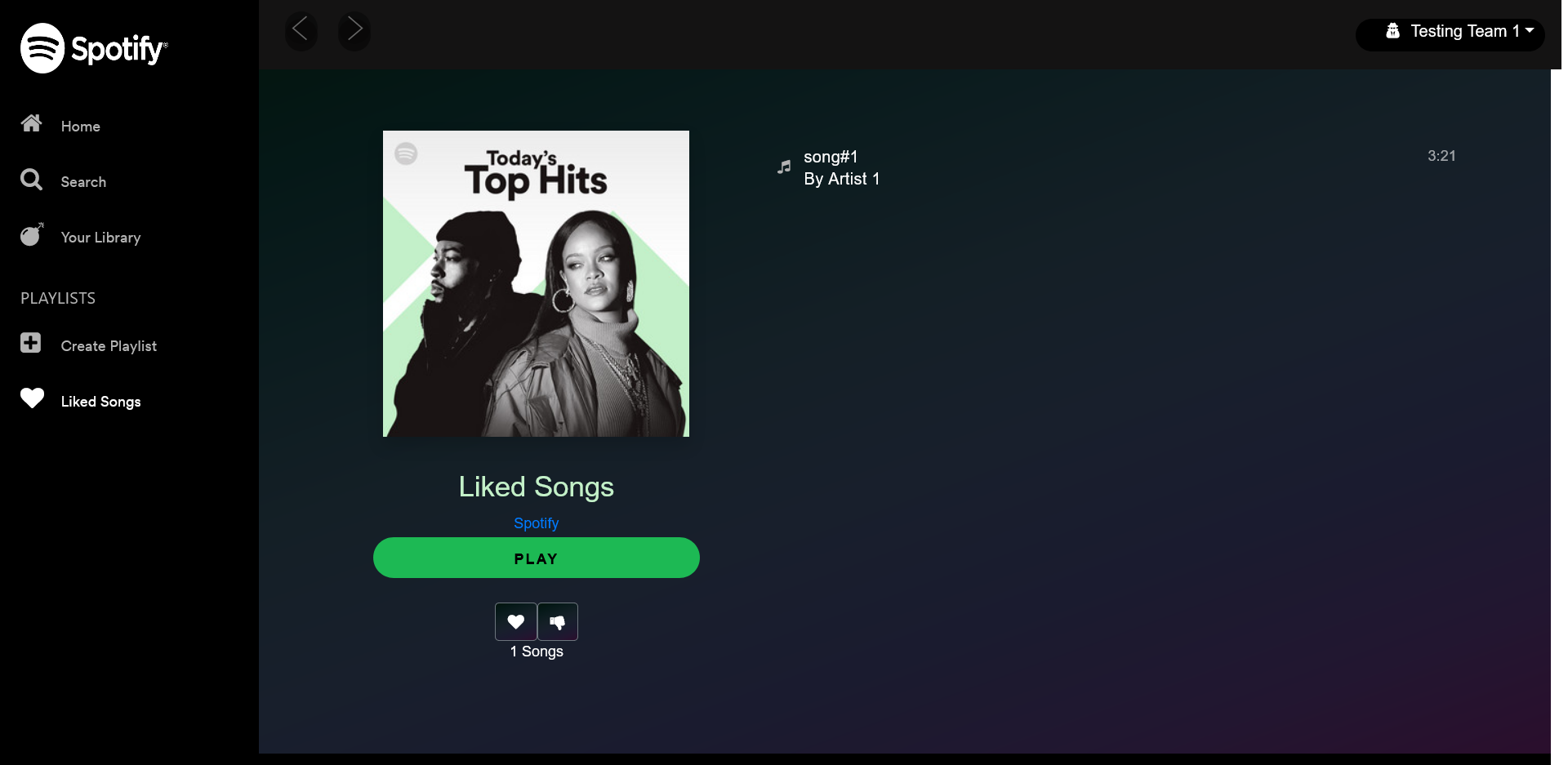
Testing Liked Songs Button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Liked Songs button is functional



## Test Teardown

[Fixture] setup::0

# Logout button - passed

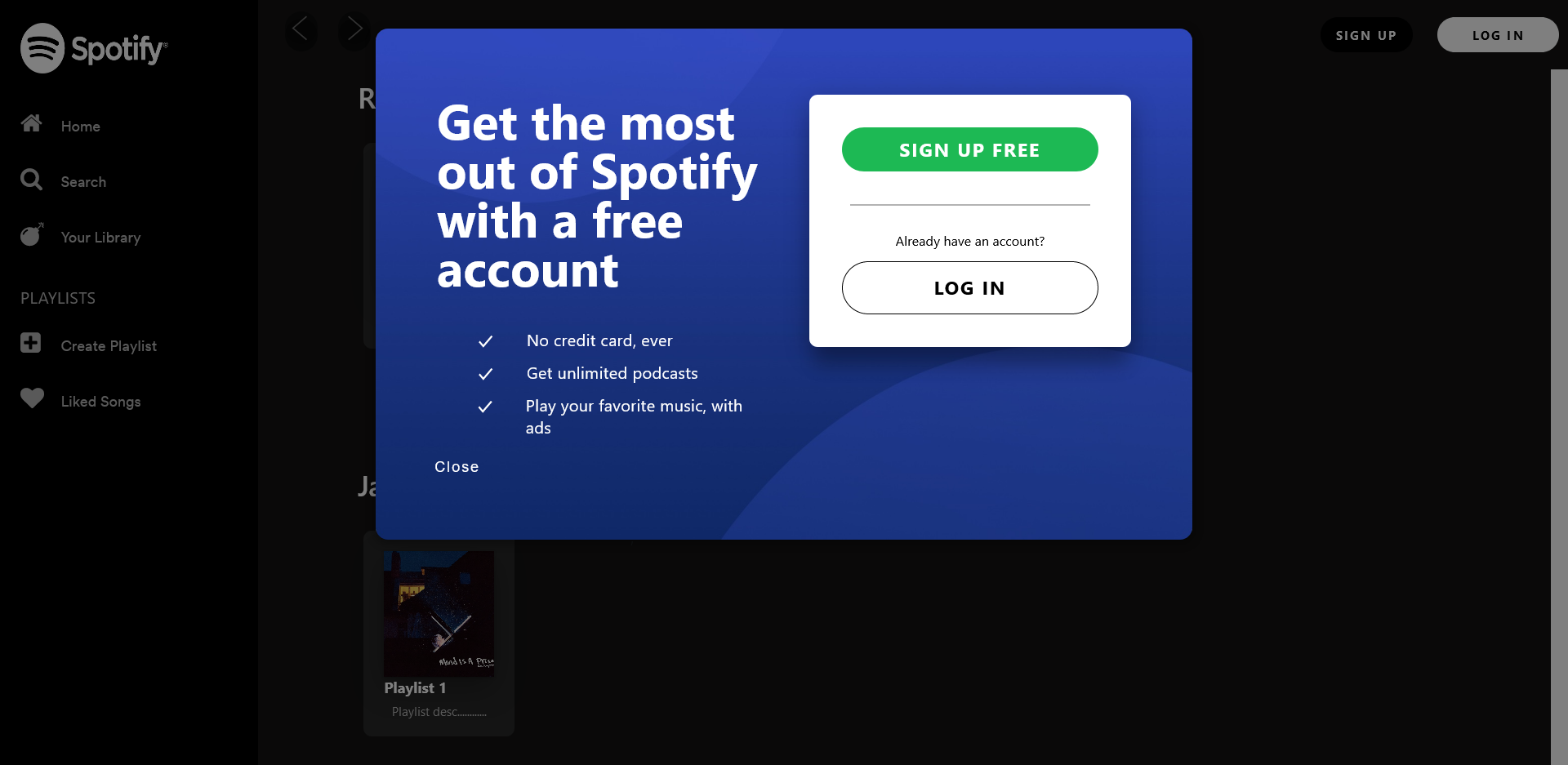
Testing Logout button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign out button is functional



## Test Teardown

[Fixture] setup::0

# Playlists cards - passed

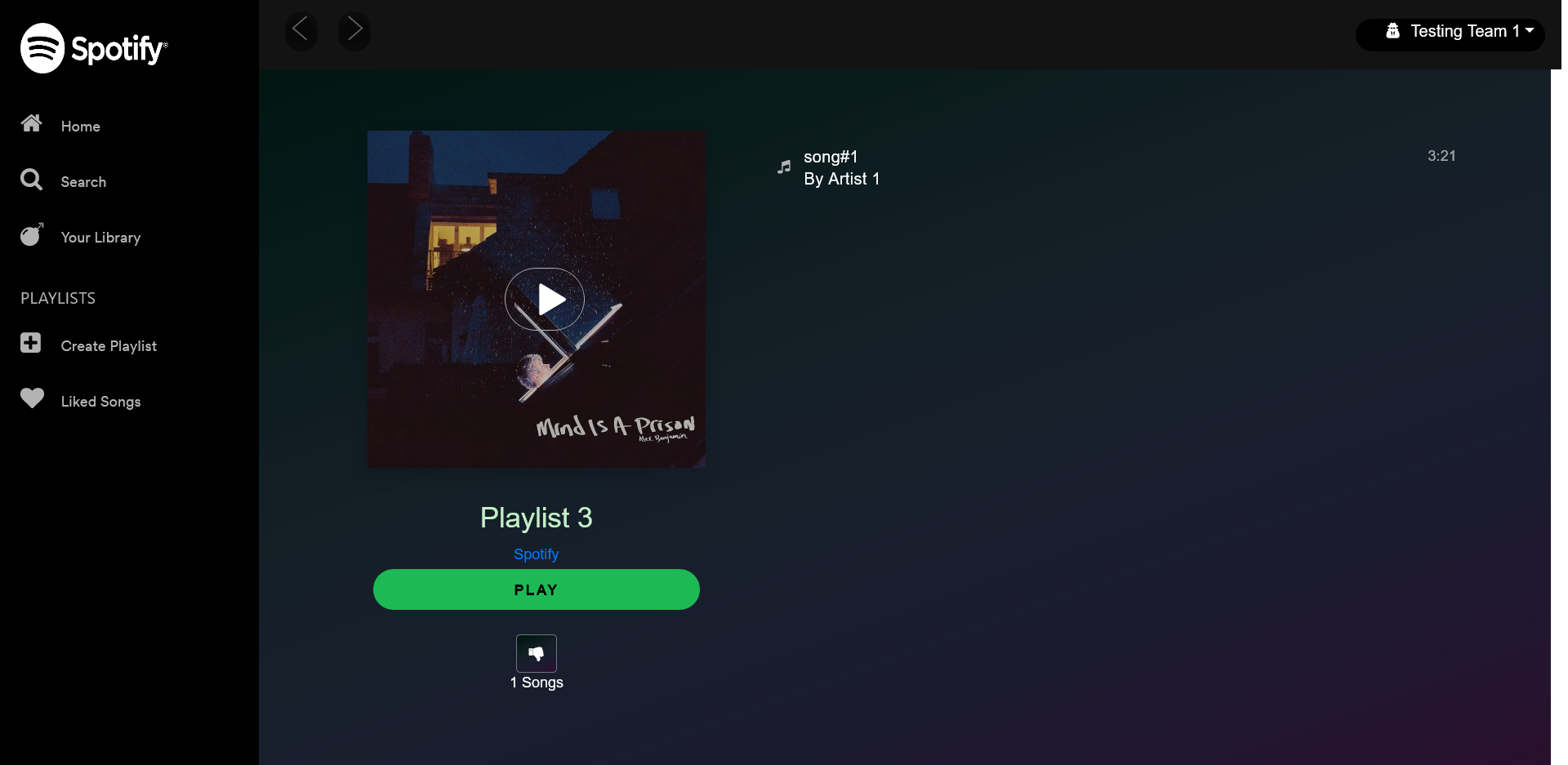
Testing Playlists cards

## Test Setup

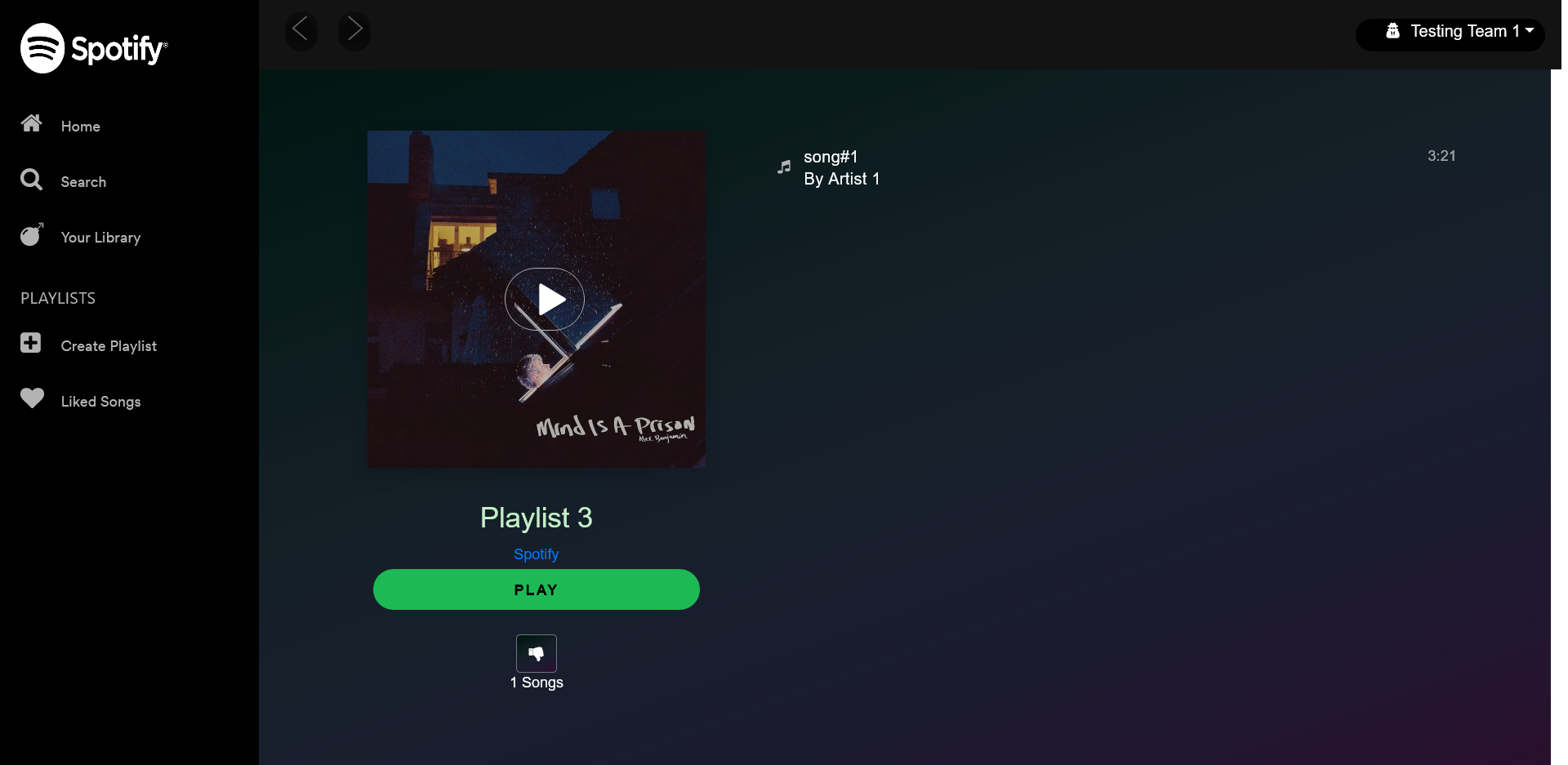
[Fixture] setup

## Test Body

[Attachment] ERROR: Playlist name is not the same as the card name. Card name = Playlist 3  
Playlist, Playlist name = Playlist 3



[Attachment] SUCCESS: Playlist cards are functional



## Test Teardown

[Fixture] setup::0

# Refresh Page - passed

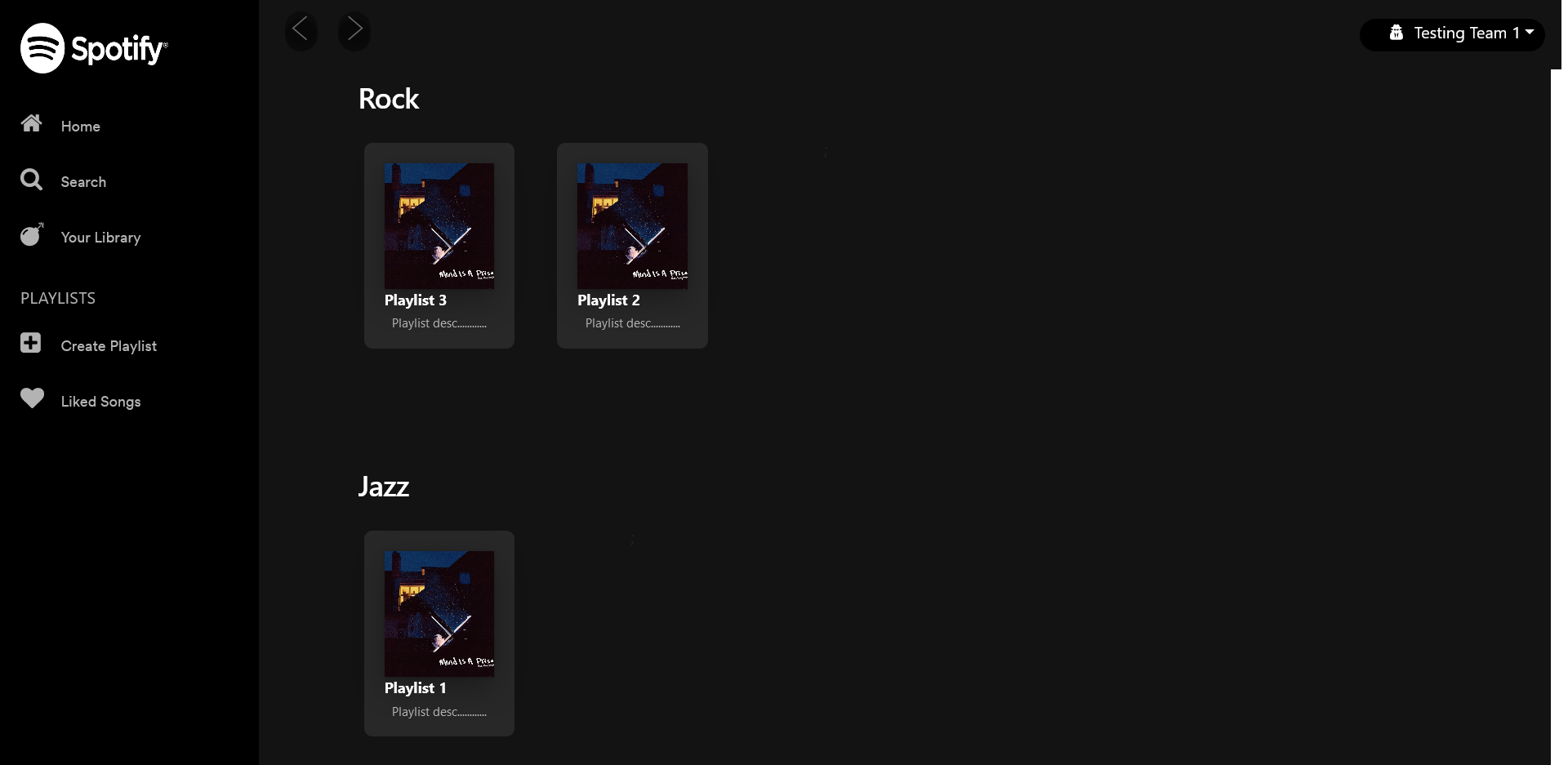
Testing Refresh and make sure that elements are visible

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Elements are still visible after refresh



## Test Teardown

[Fixture] setup::0

# Testing Login button - passed

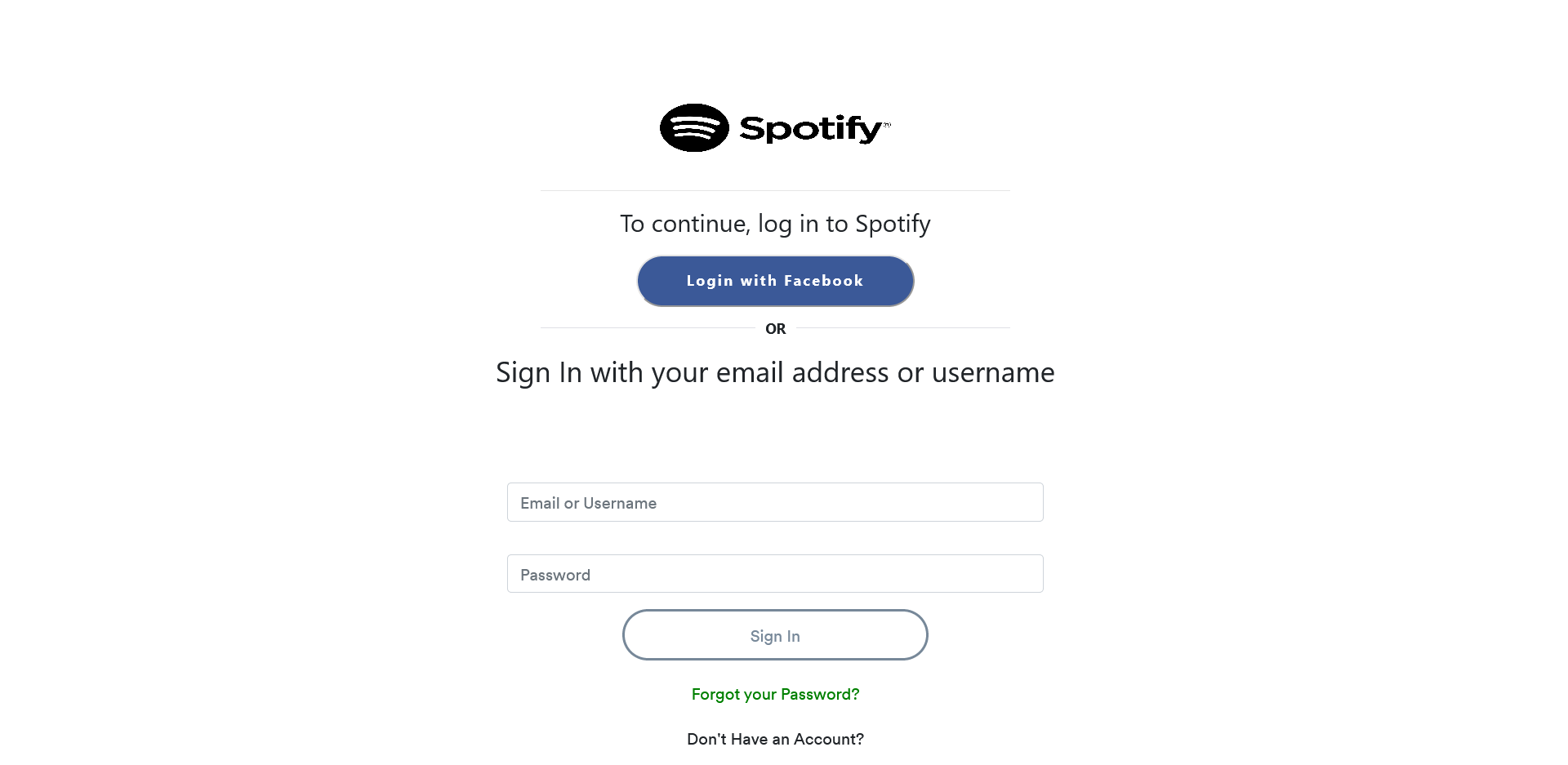
Testing Login button

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] SUCCESS: Login button is functional



## Test Teardown

[Fixture] setup\_final::0

# Testing Sign up button - passed

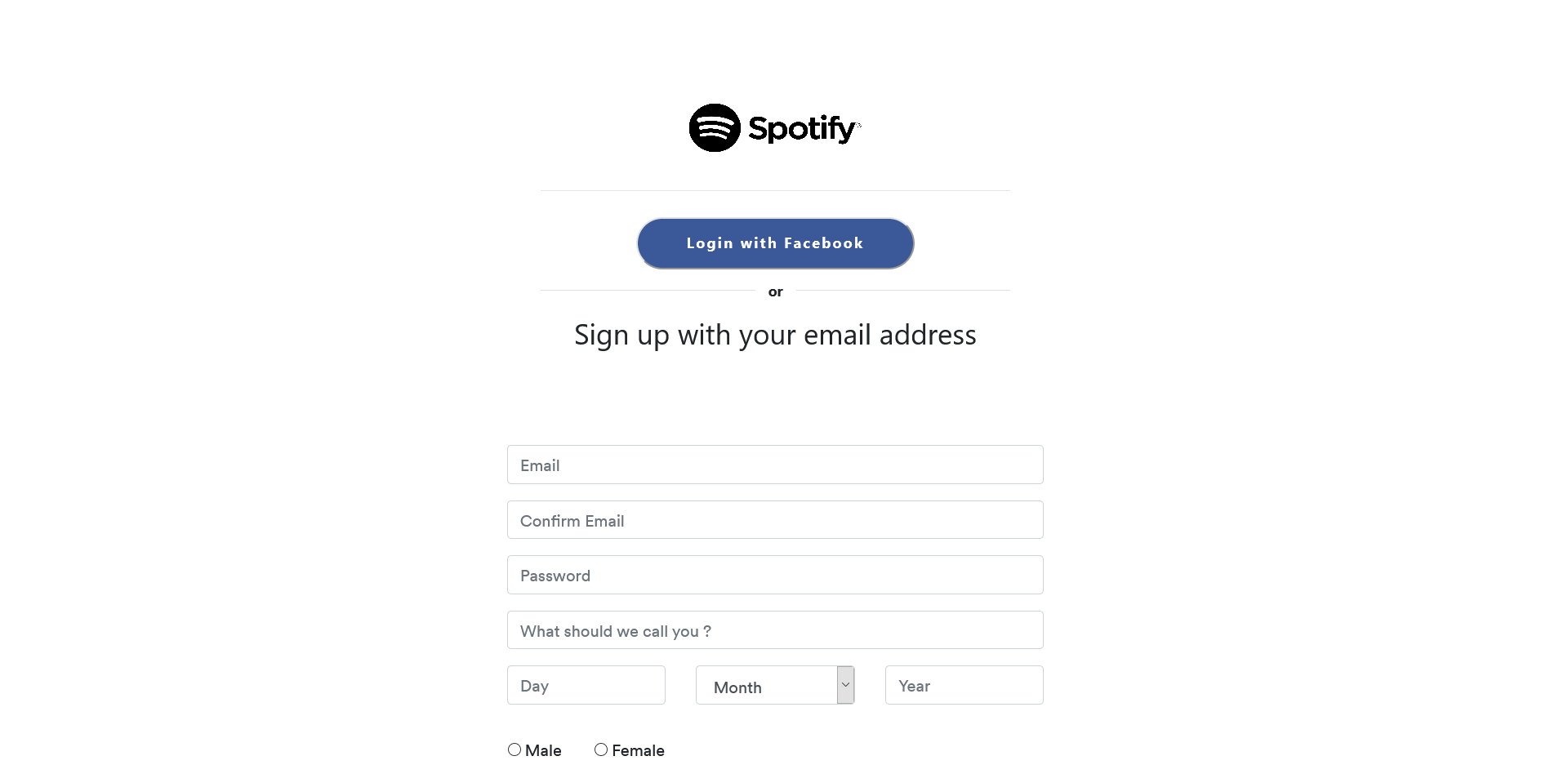
Testing Sign up button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up button is functional



## Test Teardown

[Fixture] setup::0

# Your Library Button - passed

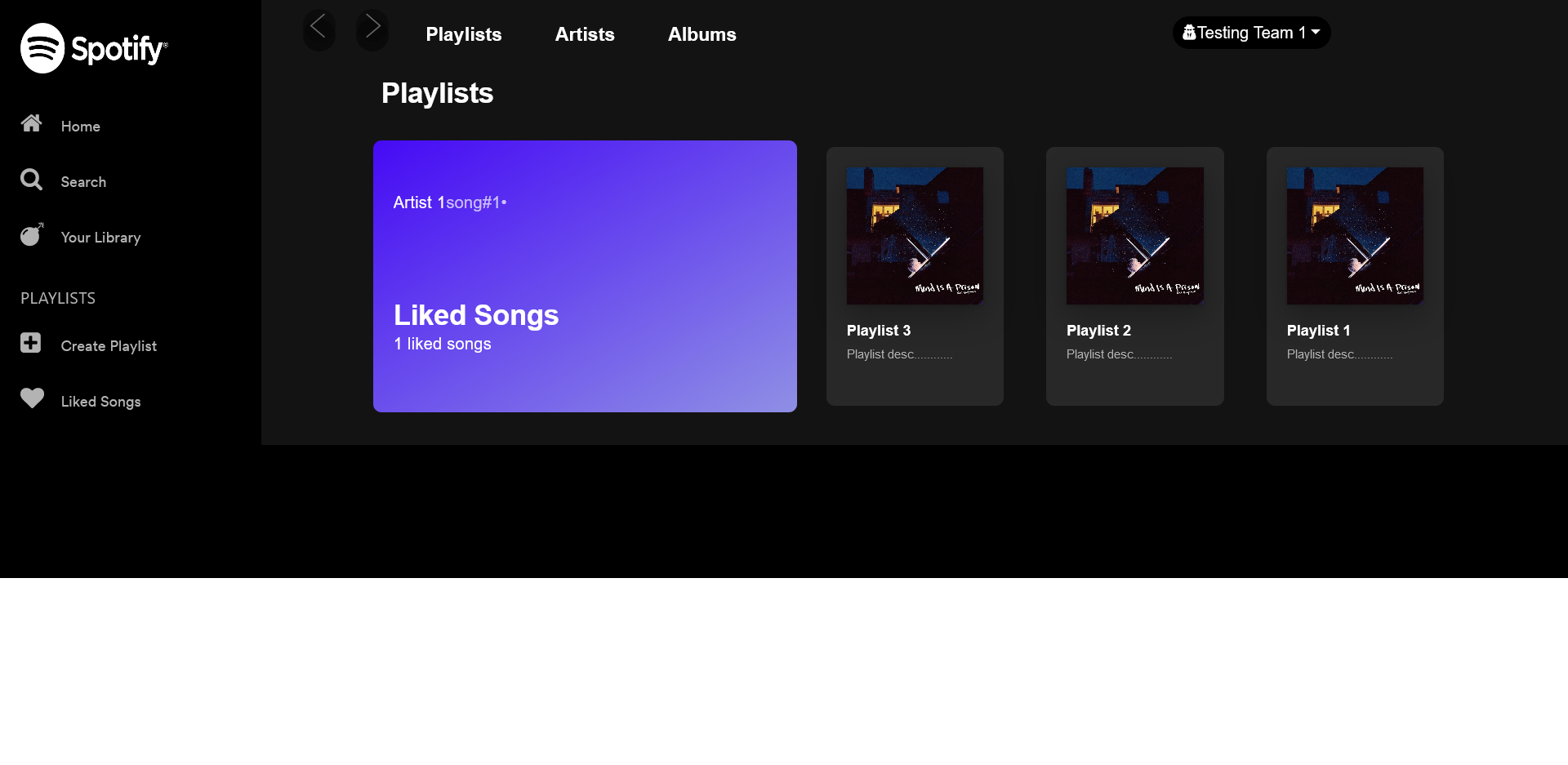
Testing Your Library Button

## Test Setup

[Fixture] setup\_initial

## Test Body

[Attachment] SUCCESS: Your Library button is functional



[Attachment] stdout

## Test Teardown

[Fixture] setup\_initial::0

**Your Library**

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Sat Apr 11 01:53:04 2020 End: Sat Apr 11 01:54:17 2020 Duration: 0:01:12.885000  broken: 0 (0.00%) failed: 2 (40.00%) skipped: 0 (0.00%) passed: 3 (60.00%) | Figure 1 |

Test Results

Right-click to update field.

# Clicking Liked Songs Card text - failed

Clicking on the Liked Songs Text in the card in Your Library Page

## Details

AssertionError: assert False

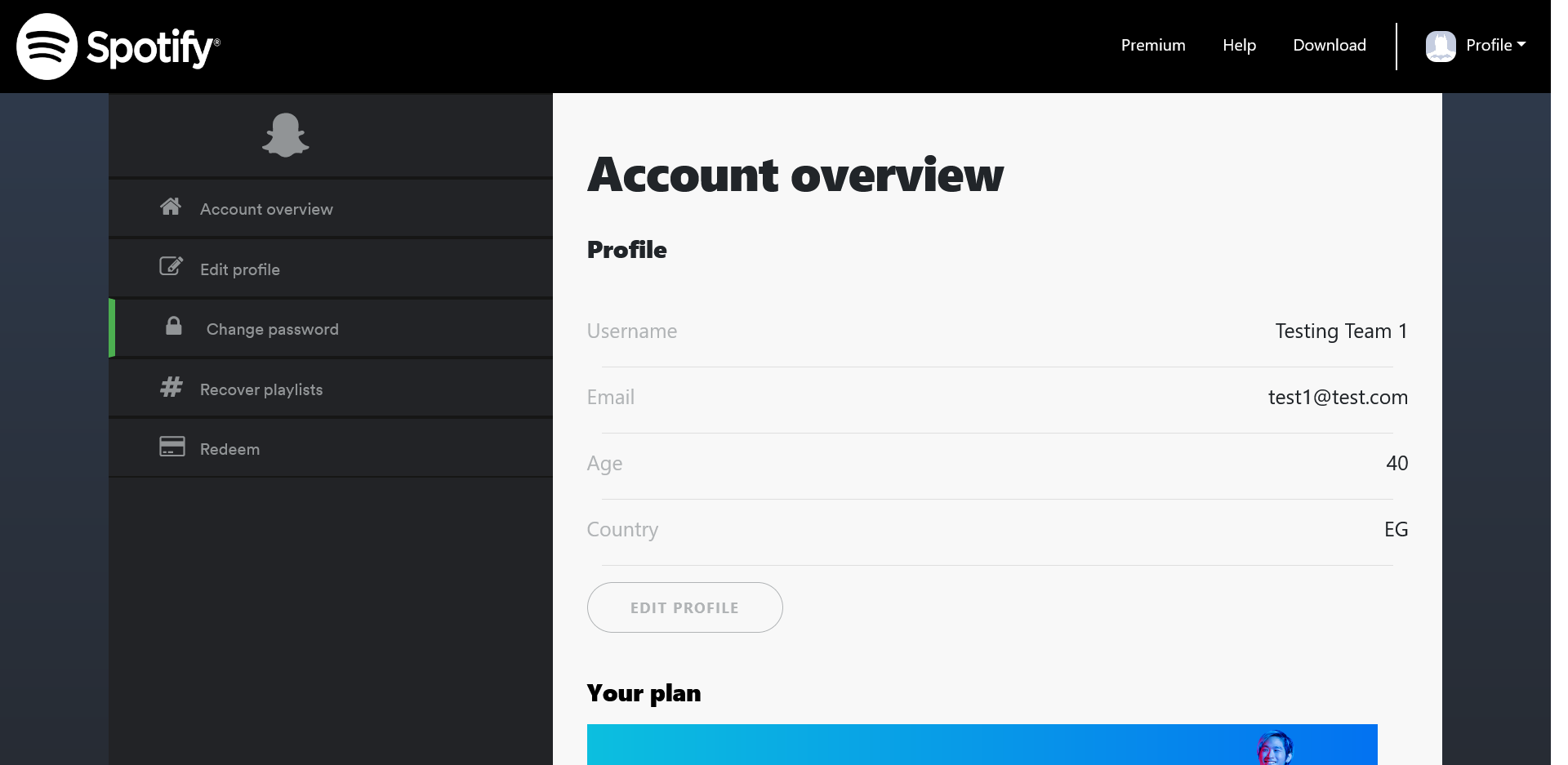
|  |
| --- |
| self = <Web\_Testing.Tests.test\_yourLibrary.TestWebPlayerLibrary object at 0x000001C1C3398DC8>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Clicking on the Liked Songs Text in the card in Your Library Page")  @allure.title("Clicking Liked Songs Card text")  @allure.description("Clicking on the Liked Songs Text in the card in Your Library Page")  @pytest.mark.Do  @pytest.mark.YourLibrary  def test\_case\_3(self, setup):  time.sleep(2)  web\_player\_home = WebPlayerHome(self.driver)  web\_player\_home.click\_your\_library()  time.sleep(2)  web\_player\_library = WebPlayerLibrary(self.driver)  web\_player\_library.click\_liked\_songs\_txt()  if self.helper.url\_has("webplayer/likedplay"):  self.helper.report\_allure("SUCCESS: The Liked Songs Card button in your Library page is functional")  assert True  else:  self.helper.report\_allure("FAILURE: The Liked Songs Card button in your Library page is not functional") > assert False E assert False  Web\_Testing\Tests\test\_yourLibrary.py:117: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] FAILURE: The Liked Songs Card button in your Library page is not functional. Went to account overview instead of Liked songs



## Test Teardown

[Fixture] setup::0

# Liked Songs Play Button - passed

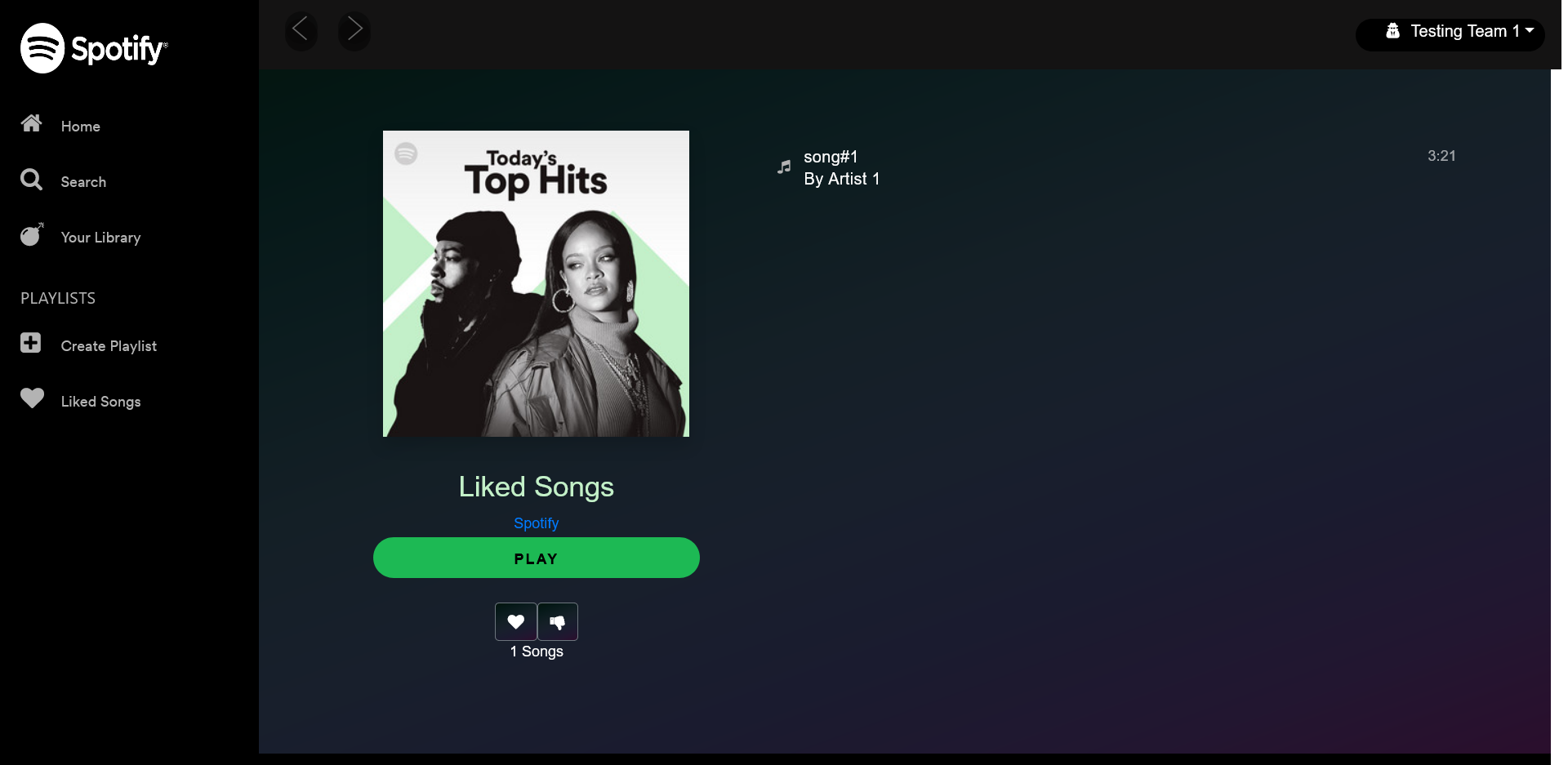
Testing Your Library Liked Songs Play Button

## Test Setup

[Fixture] setup\_initial

## Test Body

[Attachment] SUCCESS: Your Library Liked songs cards are functional



[Attachment] stdout

## Test Teardown

[Fixture] setup\_initial::0

# Liked Songs Play Button with empty playlist - passed

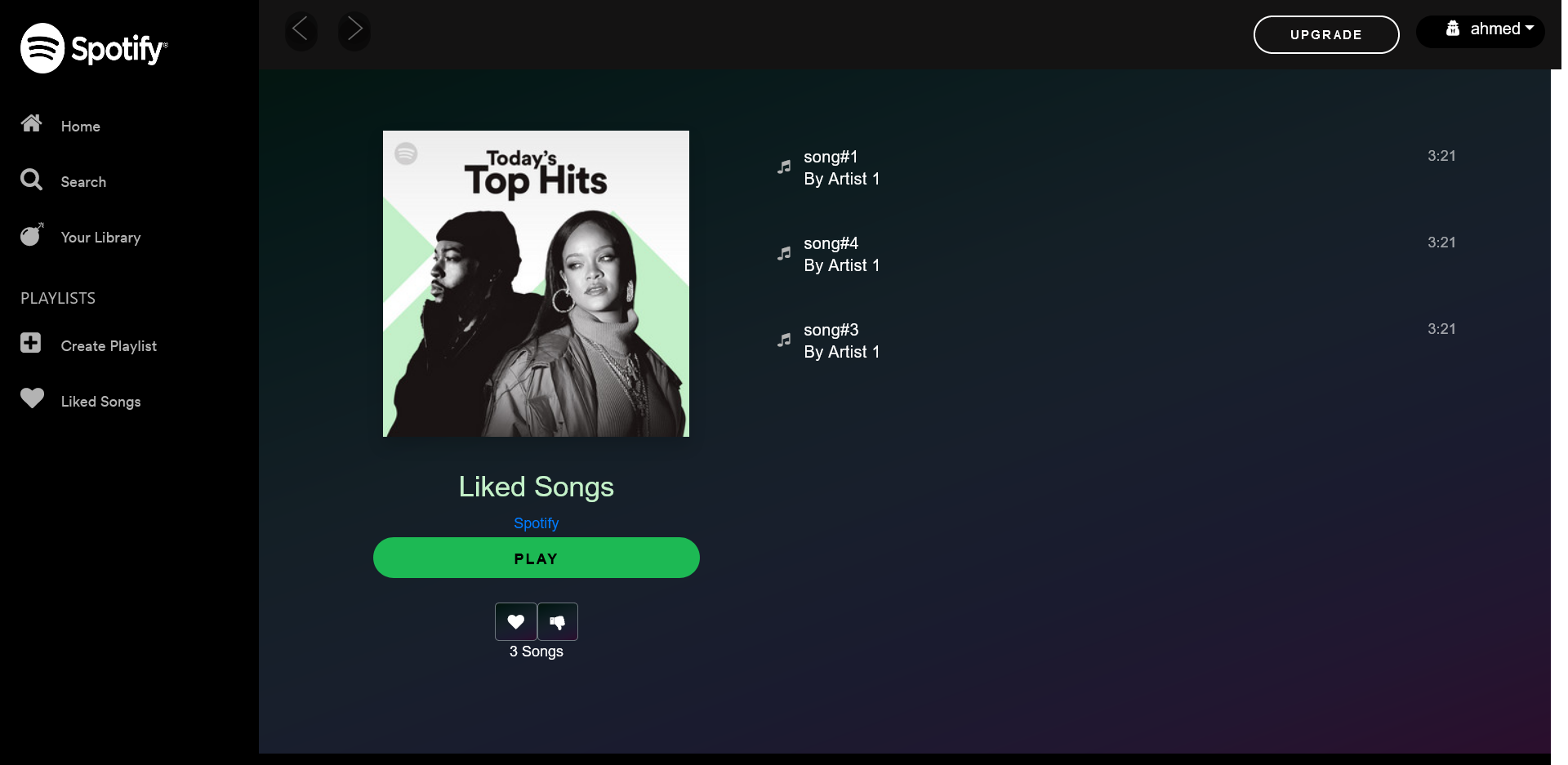
Testing Your Library Liked Songs Play Button with empty playlist

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] No of liked songs in Your Library is not the same as the number of liked songs in Liked Songs page



[Attachment] stdout

## Test Teardown

[Fixture] setup\_final::0

# Playlists cards - passed

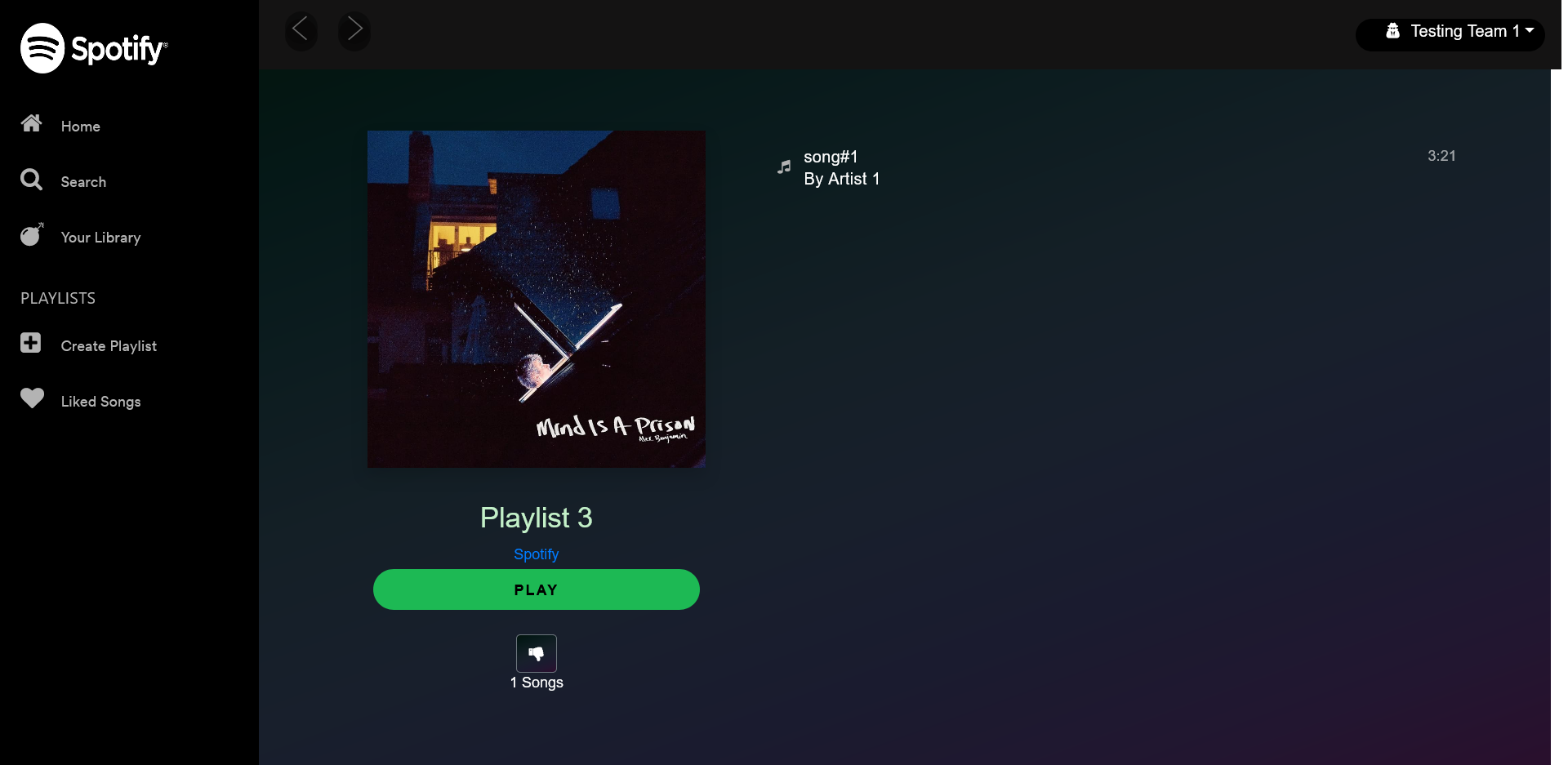
Testing Your Library Playlists cards

## Test Setup

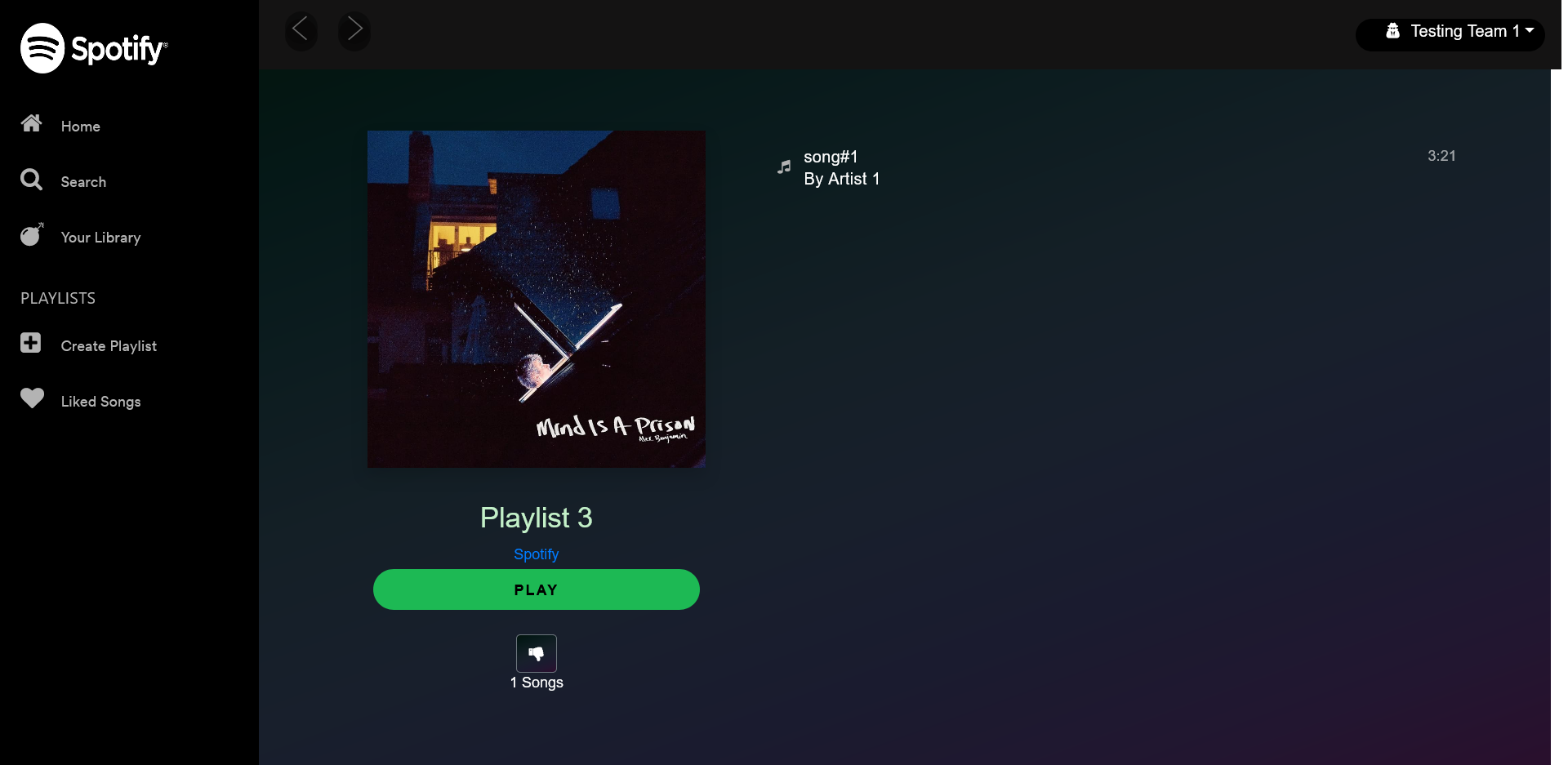
[Fixture] setup

## Test Body

[Attachment] ERROR: Playlist name is not the same as the card name. Card name = Playlist 3  
Playlist, Playlist name = Playlist 3



[Attachment] SUCCESS: Your Library page playlist cards are functional



## Test Teardown

[Fixture] setup::0

# Liked Songs Play Button with empty playlist - failed

Testing Your Library Liked Songs Play Button with empty playlist

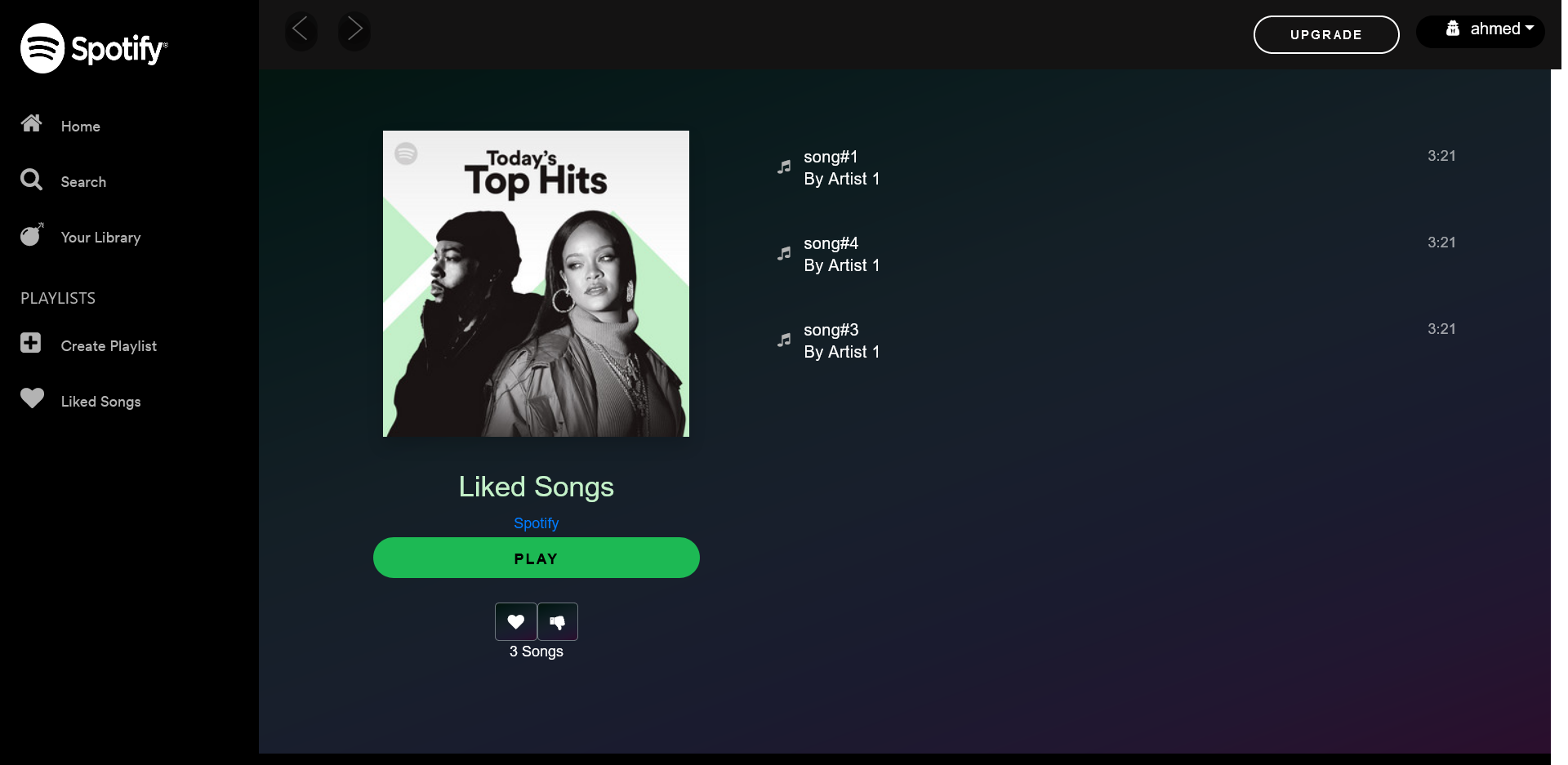
## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] No of liked songs in Your Library is not the same as the number of liked songs in Liked Songs page.

Steps t reproduce the issue -> 1. Unfollow all playlists. 2. No liked songs will appear in Your Library. 3. Some liked songs appear in Liked Songs page



## Test Teardown

[Fixture] setup\_final::0