Allure

Test Report

Test Session Summary

|  |  |
| --- | --- |
| Start: Mon Mar 23 11:12:16 2020 End: Wed Mar 25 06:28:51 2020 Duration: 1 day, 19:16:35.977000  broken: 1 (4.17%) failed: 10 (41.67%) skipped: 0 (0.00%) passed: 13 (54.17%) |  |

Test Results

[Test Results 1](#_Toc36010185)

[Sign up with different confirmation email - broken 3](#_Toc36010186)

[Clicking Download link - failed 5](#_Toc36010187)

[Clicking account link - failed 6](#_Toc36010188)

[Clicking get premium link - failed 7](#_Toc36010189)

[Clicking help link - failed 8](#_Toc36010190)

[Clicking home link - failed 9](#_Toc36010191)

[Clicking logout link - failed 10](#_Toc36010192)

[Clicking profile link - failed 11](#_Toc36010193)

[Login with correct credentials - failed 12](#_Toc36010194)

[Sign up correctly - failed 13](#_Toc36010195)

[Spotify logo - failed 15](#_Toc36010196)

[Login with empty email - passed 16](#_Toc36010197)

[Login with empty email and password - passed 17](#_Toc36010198)

[Login with empty password - passed 18](#_Toc36010199)

[Login with wrong email - passed 19](#_Toc36010200)

[Login with wrong password - passed 20](#_Toc36010201)

[Sign up clicking - passed 21](#_Toc36010202)

[Signup with already registered email - passed 22](#_Toc36010203)

[Signup with empty email - passed 23](#_Toc36010204)

[Signup with empty password - passed 24](#_Toc36010205)

[Signup with unselected day - passed 25](#_Toc36010206)

[Signup with unselected gender - passed 26](#_Toc36010207)

[Signup with unselected month - passed 27](#_Toc36010208)

[Signup with unselected year - passed 28](#_Toc36010209)

# Sign up with different confirmation email - broken

Signing up with different confirmation email from the original email field

## Details

SystemExit: Testing failed in signing up with credentials : test\_projectX@hotmail.com & test\_pass

|  |
| --- |
| self = <Web\_Testing.Pages.signup.SignupPage object at 0x108f53070>, profile = <Web\_Testing.helperClasses.Profile object at 0x108f3eac0>   def signup\_to\_spotify(self, profile):  timeout = 5  try: > self.set\_credentials(profile.email, profile.c\_email, profile.password)  Web\_Testing/Pages/signup.py:86:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.signup.SignupPage object at 0x108f53070>, email = 'test\_projectX@hotmail.com', conf\_email = 'another\_email@hotmail.com', password = 'test\_pass'   def set\_credentials(self, email, conf\_email, password): > self.set\_email(email)  Web\_Testing/Pages/signup.py:59:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.signup.SignupPage object at 0x108f53070>, email = 'test\_projectX@hotmail.com'   def set\_email(self, email): > self.email\_txt.send\_keys(email) E AttributeError: 'NoneType' object has no attribute 'send\_keys'  Web\_Testing/Pages/signup.py:37: AttributeError  During handling of the above exception, another exception occurred:  self = <Web\_Testing.Tests.test\_signup.TestLogin object at 0x108ebfd60>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Failing Signup")  @allure.sub\_suite("Sign up with different confirmation email")  @allure.title("Sign up with different confirmation email")  @allure.description("Signing up with different confirmation email from the original email field")  @pytest.mark.Do  @pytest.mark.Signup  def test\_case\_1(self, setup):  logged\_out\_page = LoggedOutHome(self.driver)  logged\_out\_page.tb\_signup\_btn.click()  sp = SignupPage(self.driver)  time.sleep(3)  if not sp.check\_signup\_page():  helper().report\_allure(self.driver, "TIMEOUT: Sign up page not loading")  assert False    constants = ConstantsClass()  emails = constants.get\_test\_emails()  profile = Profile(emails[0], "test\_pass", "Mohammad Osama", DOB(31, 1, 2002), Gender.MALE, "another\_email@hotmail.com")   > if sp.signup\_to\_spotify(profile):  Web\_Testing/Tests/test\_signup.py:60:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ Web\_Testing/Pages/signup.py:124: in signup\_to\_spotify  exit('Testing failed in signing up with credentials : ' + profile.email + " & " + profile.password) \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = Use exit() or Ctrl-D (i.e. EOF) to exit, code = 'Testing failed in signing up with credentials : test\_projectX@hotmail.com & test\_pass'   def \_\_call\_\_(self, code=None):  # Shells like IDLE catch the SystemExit, but listen when their  # stdin wrapper is closed.  try:  sys.stdin.close()  except:  pass > raise SystemExit(code) E SystemExit: Testing failed in signing up with credentials : test\_projectX@hotmail.com & test\_pass  /Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/\_sitebuiltins.py:26: SystemExit |

## Test Setup

[Fixture] setup

## Test Body

No test body information available.

## Test Teardown

[Fixture] setup::0

# Clicking Download link - failed

Clicking Download link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343D245048>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking Download link")  @allure.title("Clicking Download link")  @allure.description("Clicking Download link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_3(self, setup):  pp = PremiumPage(self.driver) > pp.click\_download\_link()  Web\_Testing\Tests\test\_premium.py:82:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343D245B88>   def click\_download\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.download\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:54: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Clicking account link - failed

Clicking account link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343CD86508>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking account link")  @allure.title("Clicking account link")  @allure.description("Clicking account link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_2(self, setup):  pp = PremiumPage(self.driver) > pp.click\_profile\_link()  Web\_Testing\Tests\test\_premium.py:64:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343CDA6A08>   def click\_profile\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.profile\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:47: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Clicking get premium link - failed

Clicking get premium link

## Details

AssertionError: assert False

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343D24F548>, setup\_final = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking get premium link")  @allure.title("Clicking get premium link")  @allure.description("Clicking get premium link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_8(self, setup\_final):  pp = PremiumPage(self.driver)  pp.get\_premium\_button  if self.driver.current\_url == "http://localhost:3000/getpremium":  assert True  else:  helper().report\_allure(self.driver, "Premium button not functional") > assert False E assert False  Web\_Testing\Tests\test\_premium.py:174: AssertionError |

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] Premium button not functional



## Test Teardown

[Fixture] setup\_final::0

# Clicking help link - failed

Clicking help link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343CD14EC8>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking help link")  @allure.title("Clicking help link")  @allure.description("Clicking help link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_4(self, setup):  pp = PremiumPage(self.driver) > pp.click\_help\_link()  Web\_Testing\Tests\test\_premium.py:99:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343C3E7C08>   def click\_help\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.help\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:61: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Clicking home link - failed

Clicking home link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343CDA61C8>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking home link")  @allure.title("Clicking home link")  @allure.description("Clicking home link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_5(self, setup):  pp = PremiumPage(self.driver) > pp.click\_home\_link()  Web\_Testing\Tests\test\_premium.py:116:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343CDA6408>   def click\_home\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.home\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:68: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Clicking logout link - failed

Clicking logout link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343D25BBC8>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking logout link")  @allure.title("Clicking logout link")  @allure.description("Clicking logout link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_6(self, setup):  pp = PremiumPage(self.driver) > pp.click\_profile\_link()  Web\_Testing\Tests\test\_premium.py:133:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343D25B248>   def click\_profile\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.profile\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:47: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Clicking profile link - failed

Clicking profile link

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343D280C48>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Clicking profile link")  @allure.title("Clicking profile link")  @allure.description("Clicking profile link")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_7(self, setup):  pp = PremiumPage(self.driver) > pp.click\_profile\_link()  Web\_Testing\Tests\test\_premium.py:151:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343D280AC8>   def click\_profile\_link(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.profile\_link).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:47: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Login with correct credentials - failed

Signing in with the following credentials email : kamelmohsenkamel@gmail.com & password: Kimo2010

## Details

AssertionError: assert False

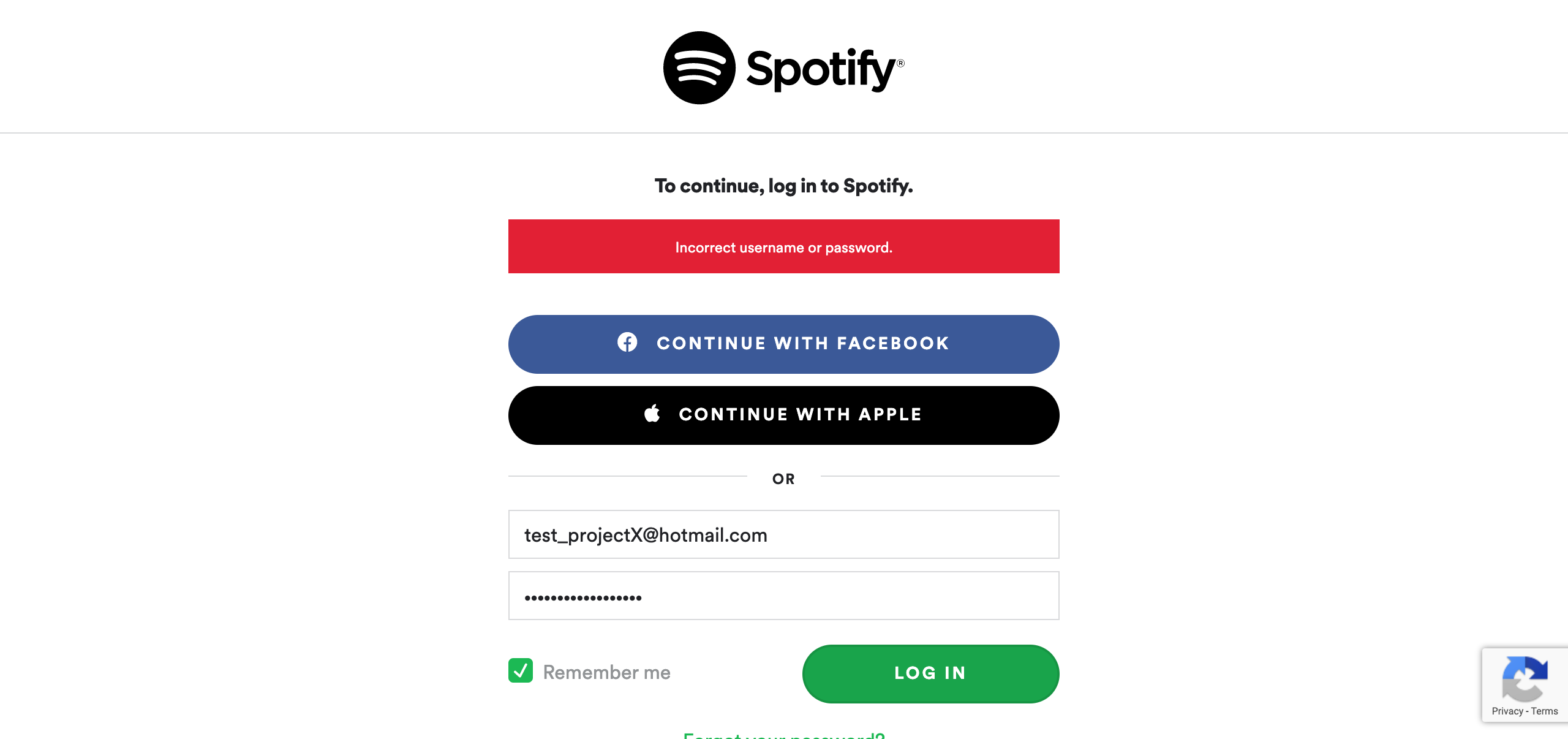
|  |
| --- |
| self = <Web\_Testing.Tests.test\_signin.TestLogin object at 0x108f41c10>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Passing login")  @allure.sub\_suite("Login with correct credentials")  @allure.title("Login with correct credentials")  @allure.description("Signing in with the following credentials email : kamelmohsenkamel@gmail.com & password: Kimo2010")  @pytest.mark.Do  @pytest.mark.Login  def test\_case\_1(self, setup):  logged\_out\_page = LoggedOutHome(self.driver)  logged\_out\_page.tb\_login\_btn.click()  lp = LoginPage(self.driver)  lp.check\_login\_page()  time.sleep(3)  if lp.login\_to\_spotify("test\_projectX@hotmail.com", ConstantsClass().get\_pass("test\_projectX@hotmail.com")):  helper().report\_allure(self.driver, "SUCCESS: Login succeeded with correct credentials")  assert True  else:  helper().report\_allure(self.driver, "ERROR: Login failed with correct credentials") > assert False E assert False  Web\_Testing/Tests/test\_signin.py:57: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] ERROR: Login failed with correct credentials



## Test Teardown

[Fixture] setup::0

# Sign up correctly - failed

Signing up with correct format with Kamel, kamelmohsenkamel@gmail.com, Kimo2010, 21, M

## Details

AssertionError: assert False

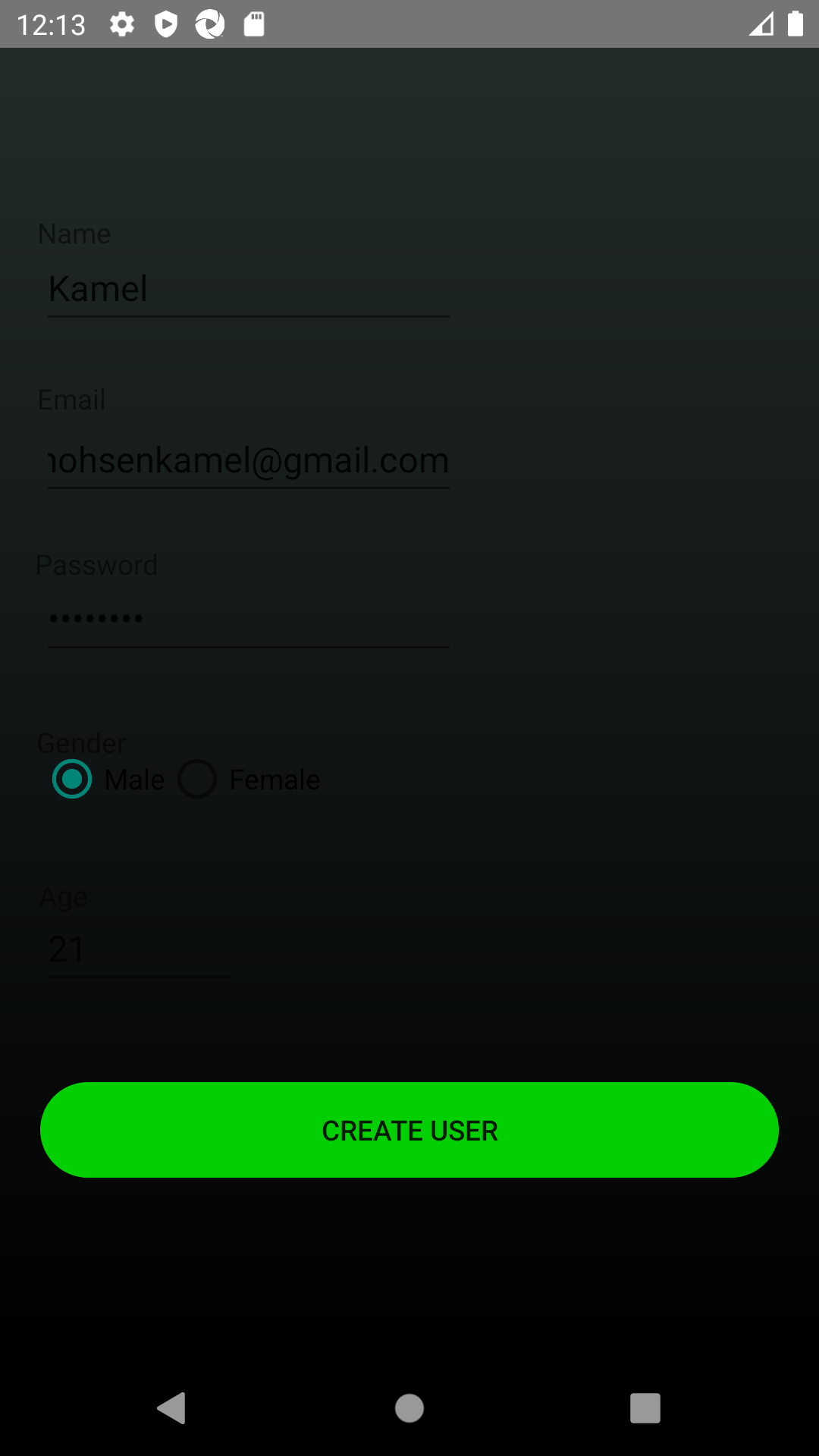
|  |
| --- |
| self = <Mobile\_Testing.Tests.test\_signup.TestAuthentication object at 0x106ee8fd0>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Signup Tests")  @allure.sub\_suite("Sign up correctly")  @allure.title("Sign up correctly")  @allure.description("Signing up with correct format with Kamel, kamelmohsenkamel@gmail.com, Kimo2010, 21, M")  @pytest.mark.Do  @pytest.mark.Signup  def test\_case\_1(self, setup):  ap = AuthenticationPage(self.driver)  ap.click\_signup\_button()  sp = SignupPage(self.driver)  sp.do\_the\_signup("Kamel", "kamelmohsenkamel@gmail.com", "Kimo2010", "21", "M")  if Helper.element\_exists\_by\_id(self.driver, HomePage.logout\_button\_id):  assert True  else:  print(self.driver.current\_activity)  Helper.report\_allure(self.driver, "Sign up Failed") > assert False E assert False  Mobile\_Testing/Tests/test\_signup.py:42: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Sign up Failed



[Attachment] stdout

## Test Teardown

[Fixture] setup::0

# Spotify logo - failed

Spotify logo

## Details

AssertionError

|  |
| --- |
| self = <Web\_Testing.Tests.test\_premium.TestPremium object at 0x000002343CD86788>, setup = None   @allure.severity(allure.severity\_level.BLOCKER)  @allure.story("Premium tests")  @allure.sub\_suite("Spotify logo")  @allure.title("Spotify logo")  @allure.description("Spotify logo")  @pytest.mark.Do  @pytest.mark.Premium  def test\_case\_1(self, setup):  pp = PremiumPage(self.driver)   > pp.click\_spotify\_logo()  Web\_Testing\Tests\test\_premium.py:47:  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_  self = <Web\_Testing.Pages.premium.PremiumPage object at 0x000002343CD86988>   def click\_spotify\_logo(self):  if helper().element\_exists\_by\_xpath(self.driver, self.spotify\_logo):  helper().find\_element\_by\_xpath(self.driver, self.spotify\_logo).click()  else:  helper().report\_allure(self.driver, "Not Found") > assert False E AssertionError  Web\_Testing\Pages\premium.py:25: AssertionError |

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Not Found



## Test Teardown

[Fixture] setup::0

# Login with empty email - passed

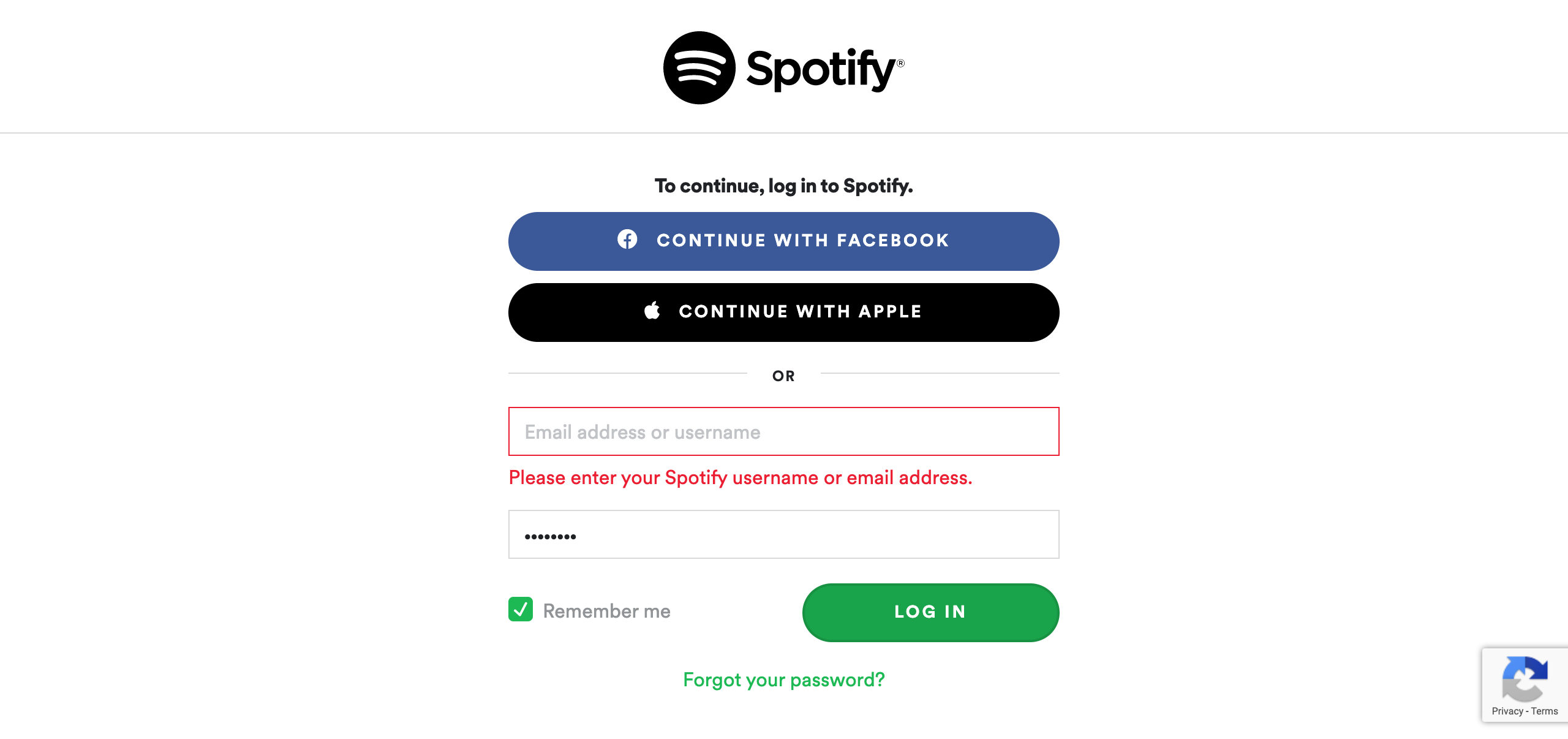
Signing in with empty email

## 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

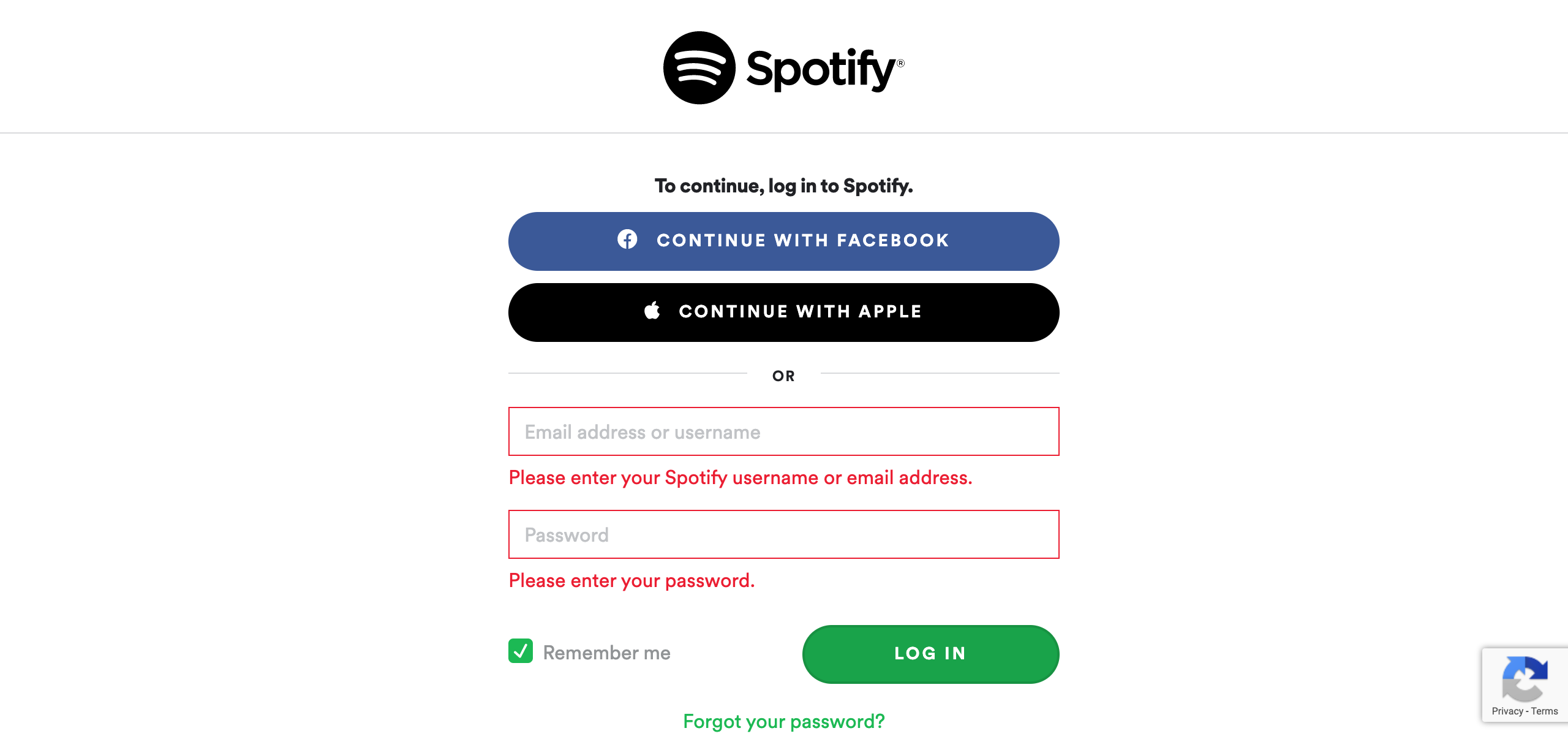
Signing in with empty email and password

## Test Setup

[Fixture] setup\_final

## Test Body

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



[Attachment] stdout

## Test Teardown

[Fixture] setup\_final::0

# Login with empty password - passed

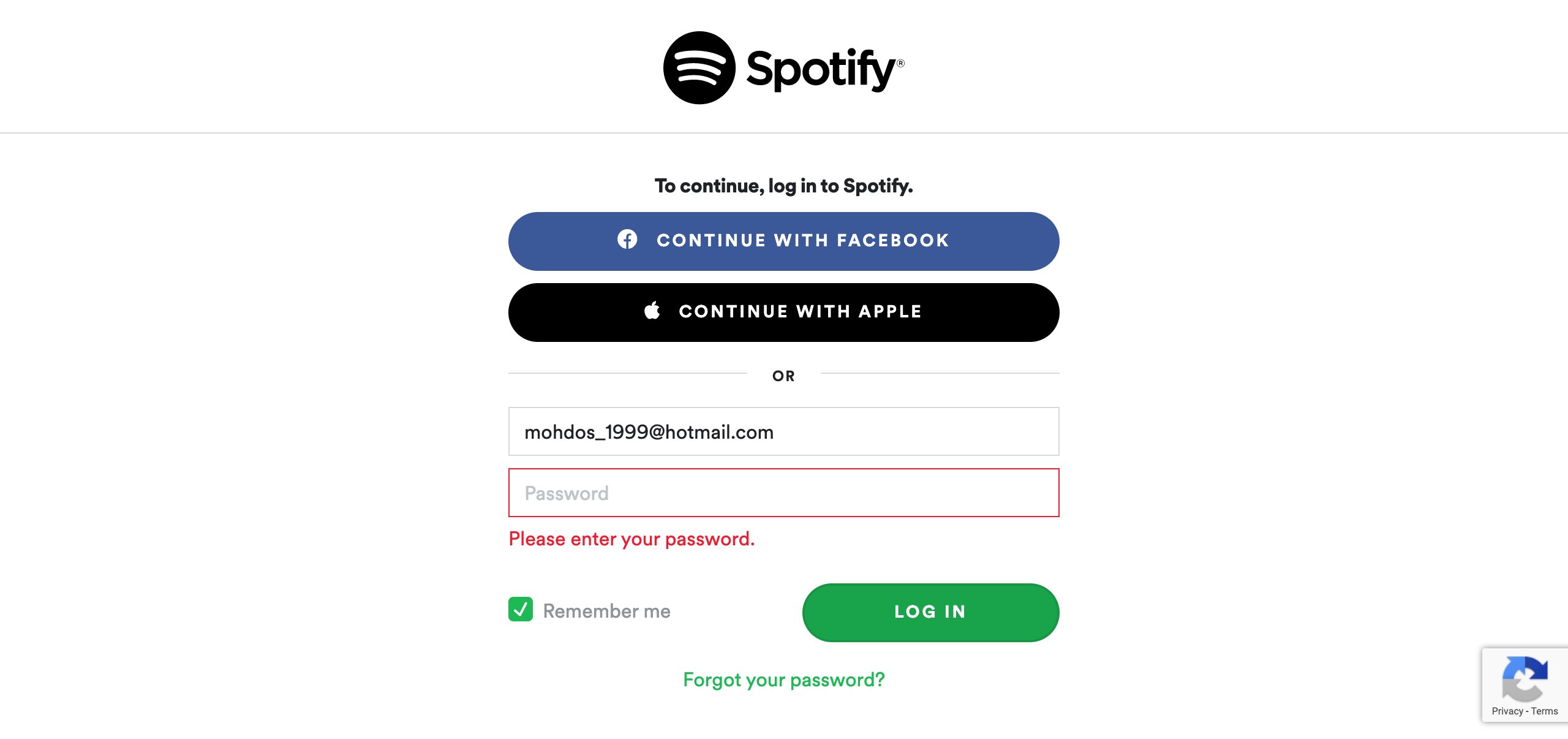
Signing in with empty password

## 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

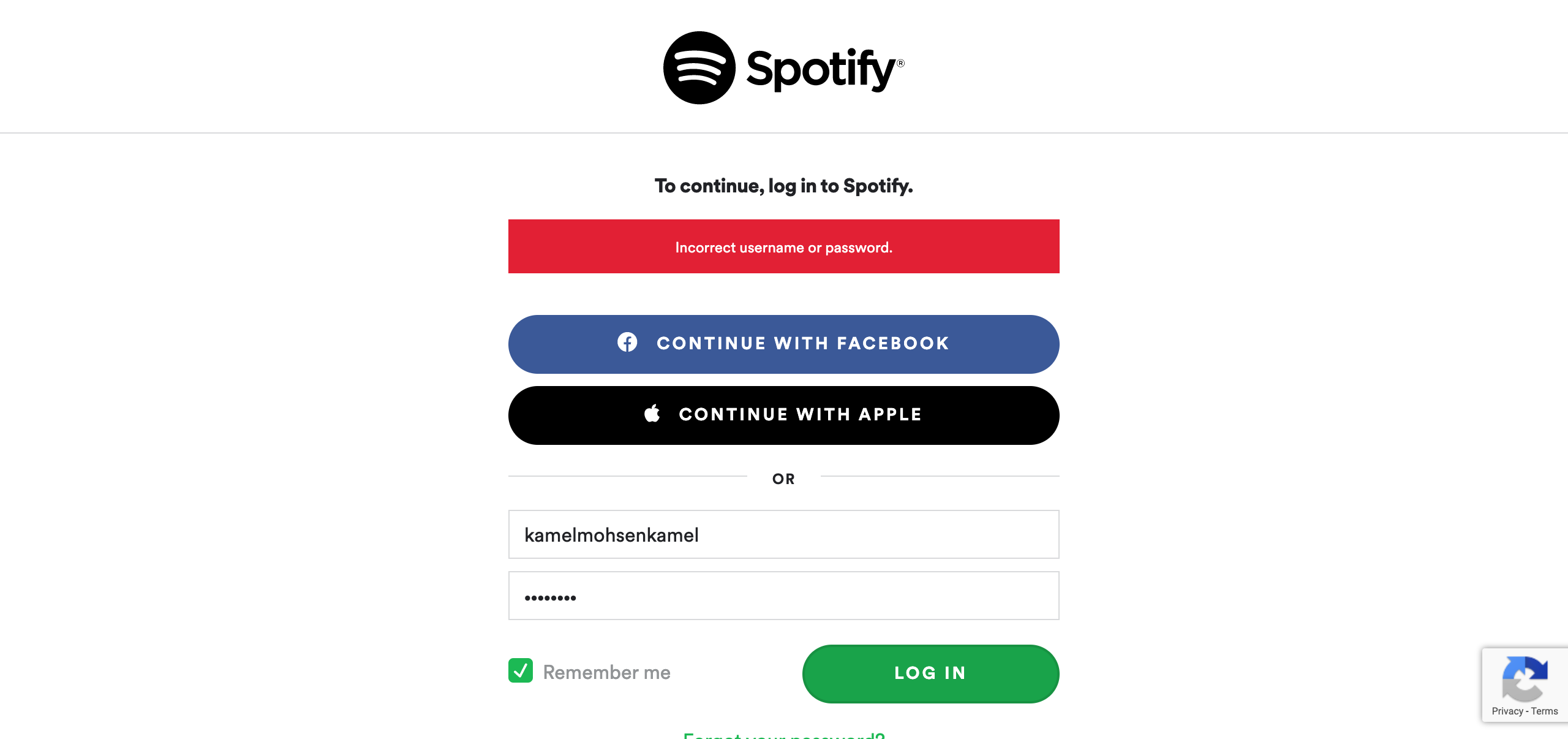
Signing in with the following credentials email : kamelmohsenkamel & password: Kimo2010

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with incorrect credentials



## Test Teardown

[Fixture] setup::0

# Login with wrong password - passed

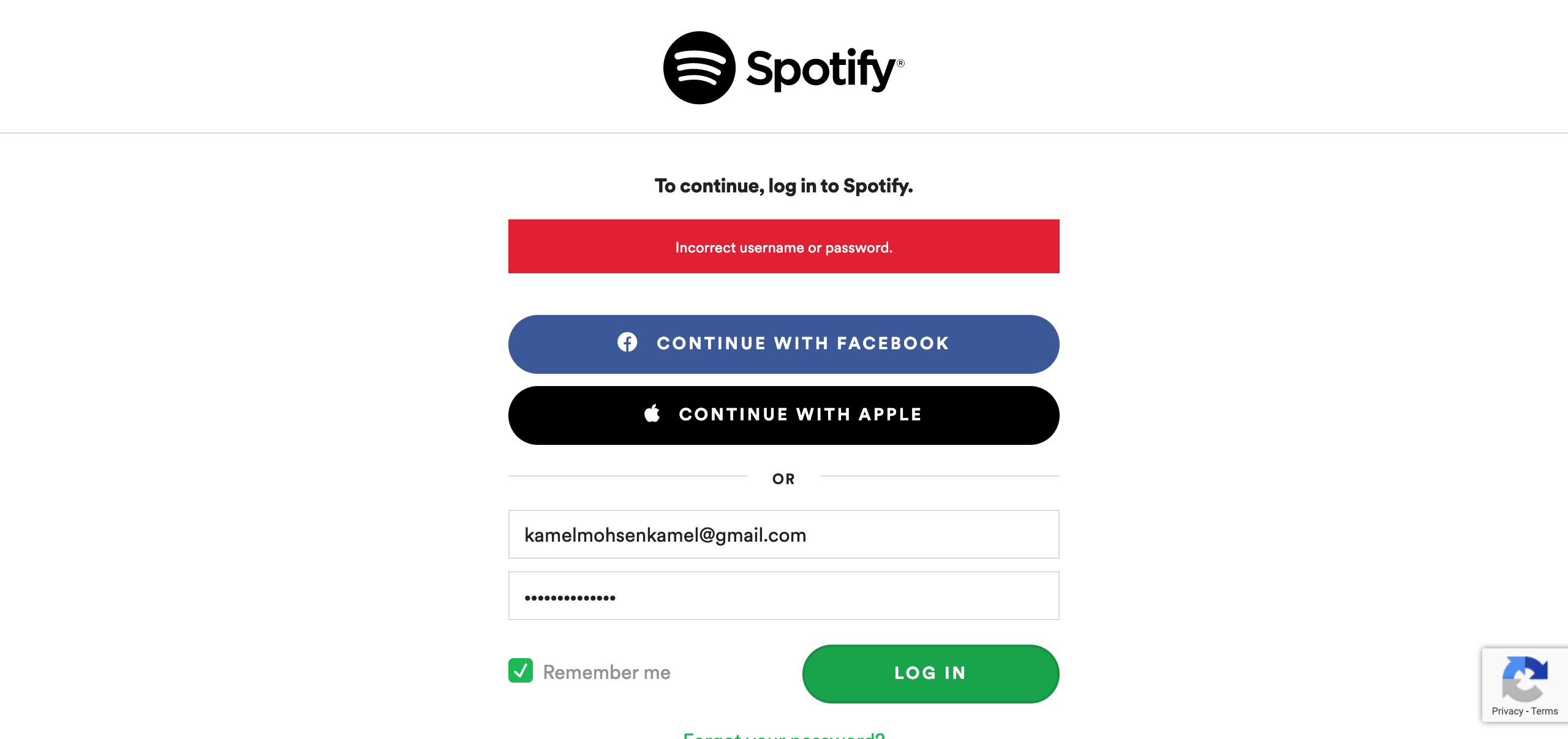
Signing in with the following credentials email : kamelmohsenkamel@gmail.com & password: wrong password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Login failed with incorrect credentials



## Test Teardown

[Fixture] setup::0

# Sign up clicking - passed

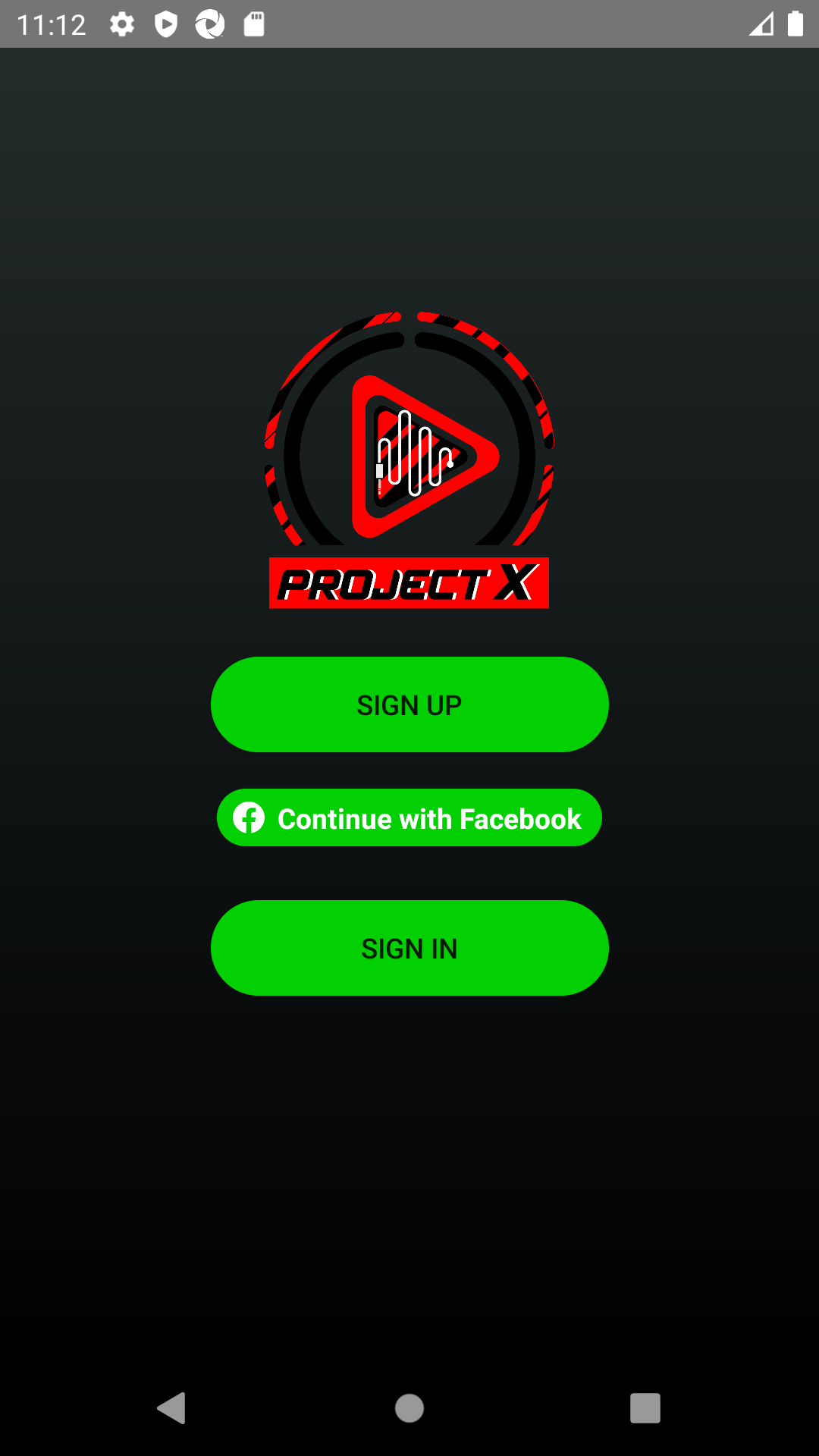
Clicking the sign up button

## Test Setup

[Fixture] setup

## Test Body

[Attachment] Sign up Page



## Test Teardown

[Fixture] setup::0

# Signup with already registered email - passed

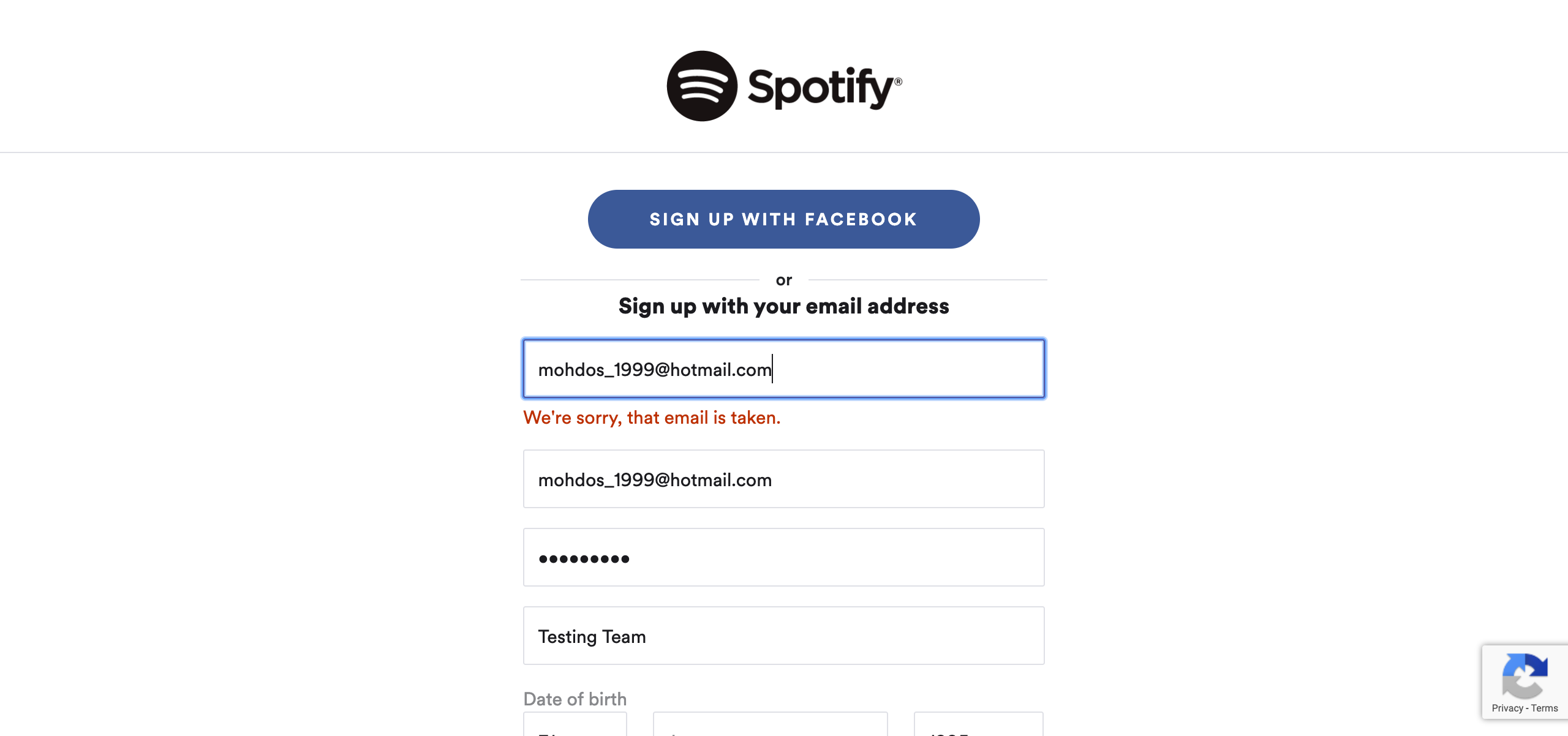
Signing up with the following credentials email: test\_projectX@hotmail.com

## Test Setup

[Fixture] setup

## Test Body

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



## Test Teardown

[Fixture] setup::0

# Signup with empty email - passed

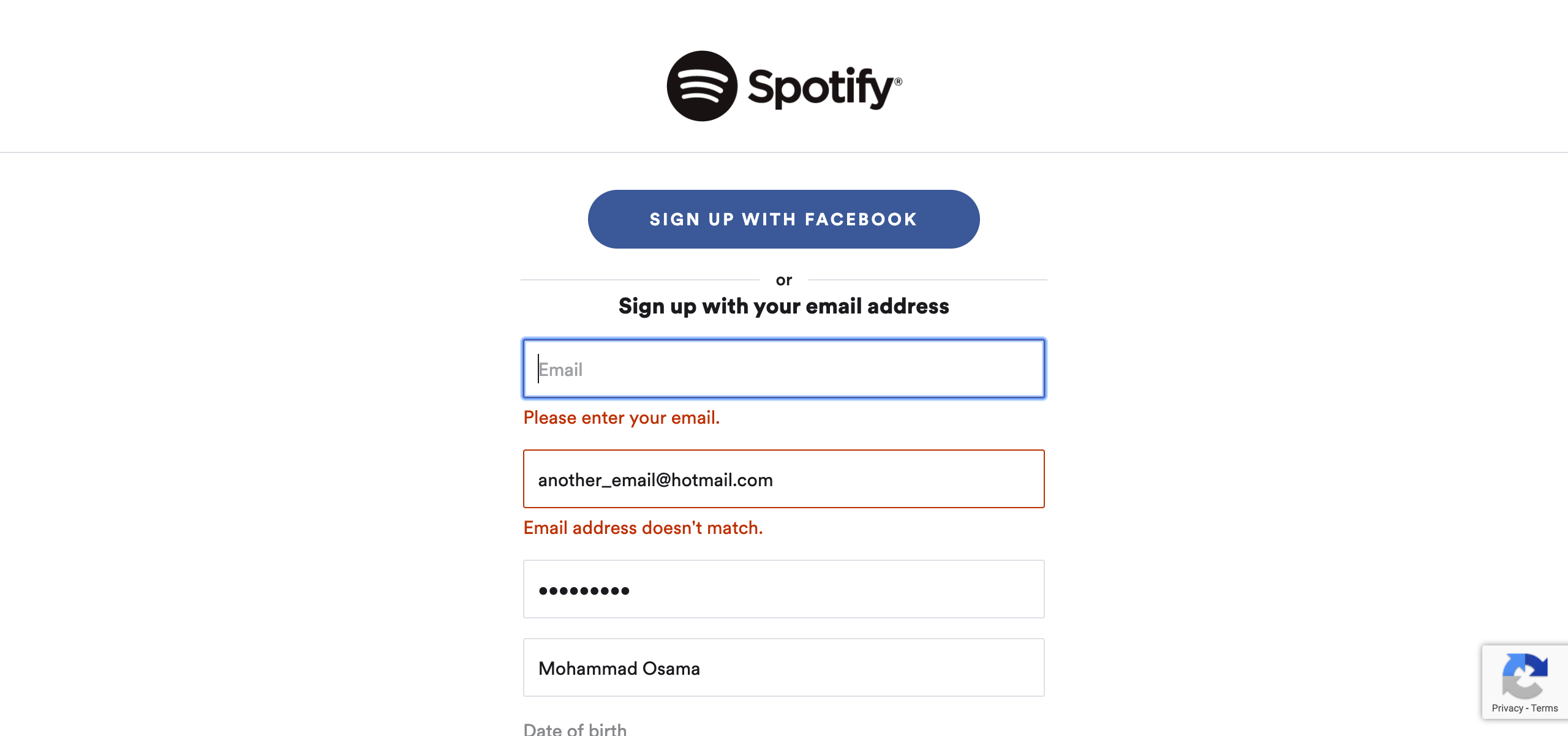
Signing up with empty email

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with empty email



## Test Teardown

[Fixture] setup::0

# Signup with empty password - passed

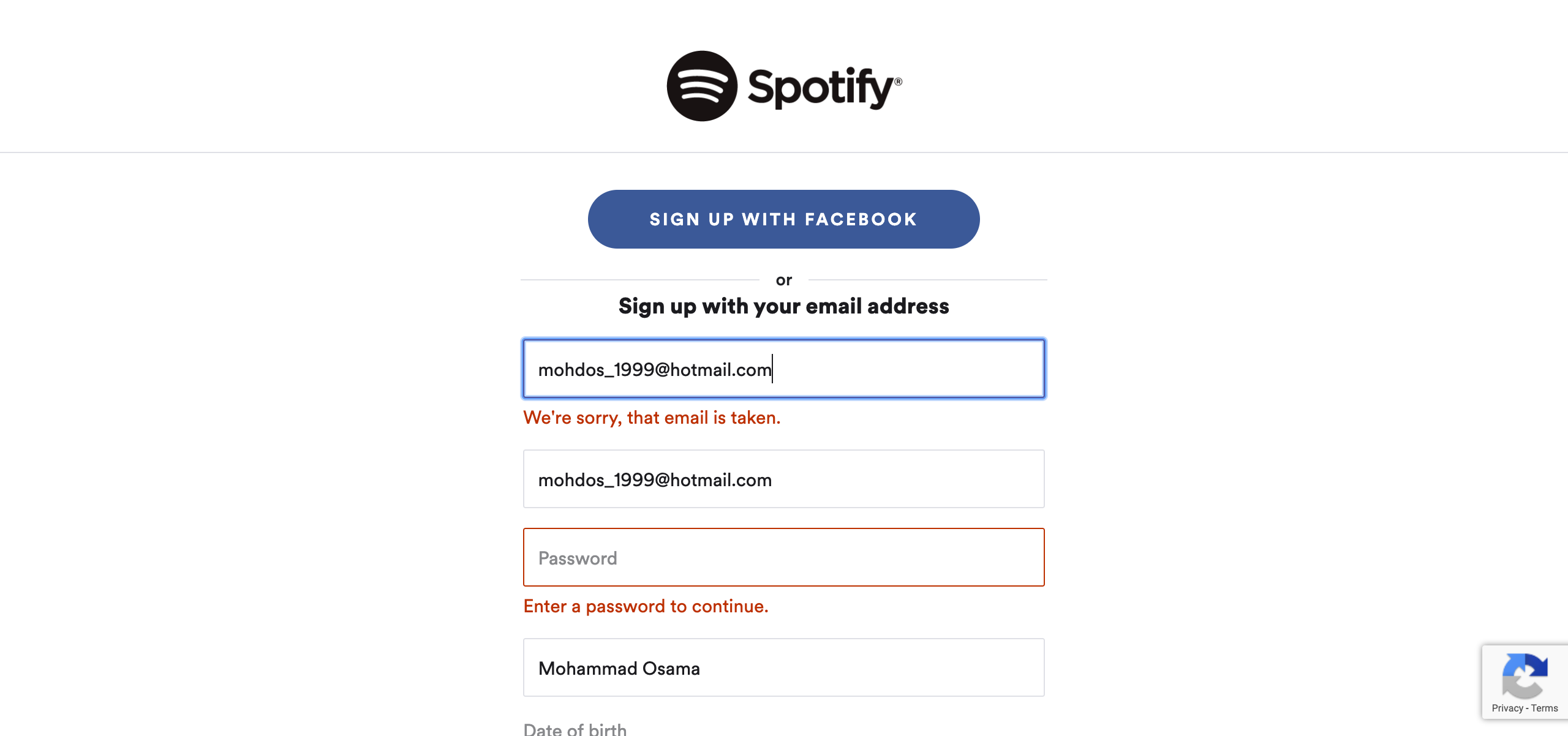
Signing up with empty password

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with empty password



## Test Teardown

[Fixture] setup::0

# Signup with unselected day - passed

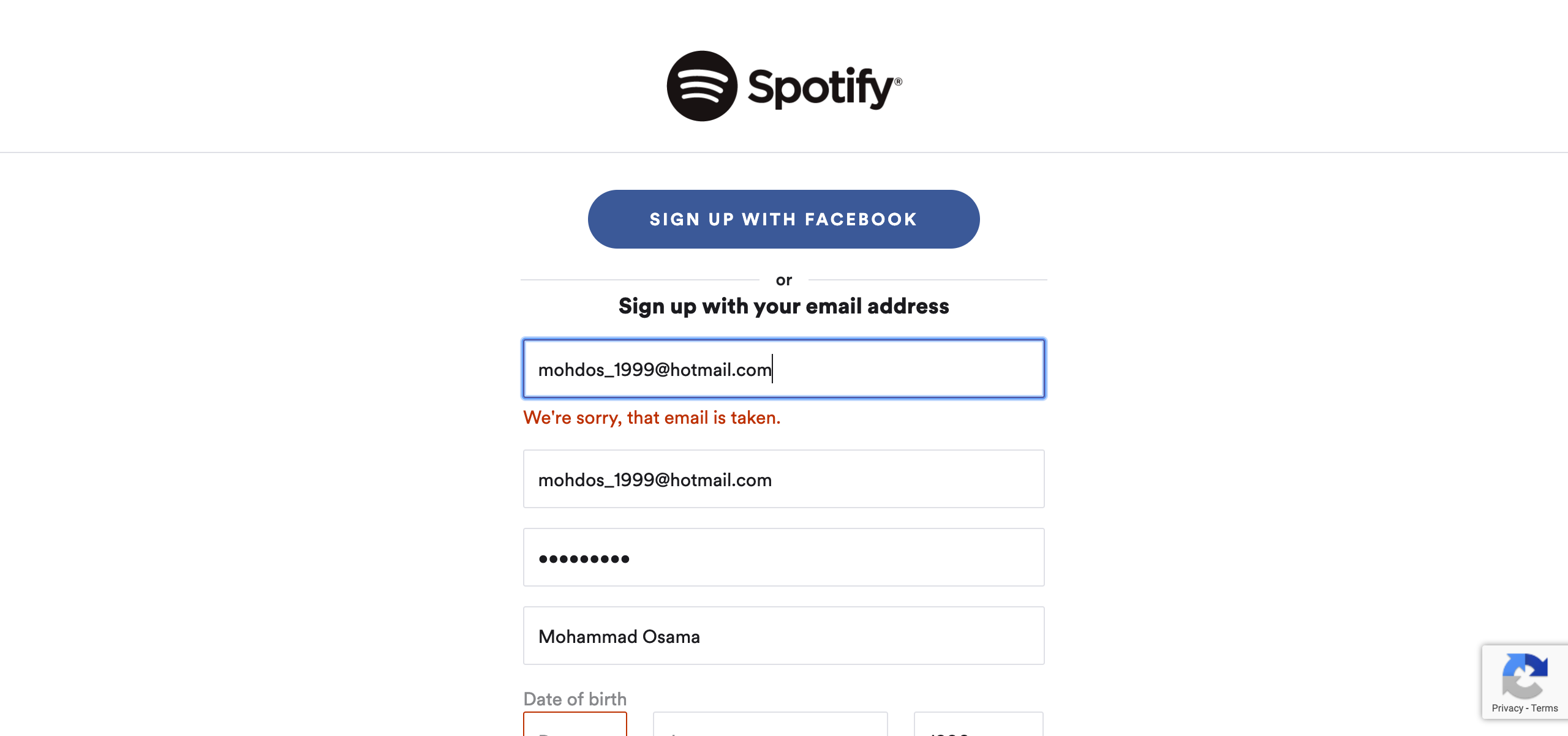
Signing up with unselected day

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected day



## Test Teardown

[Fixture] setup::0

# Signup with unselected gender - passed

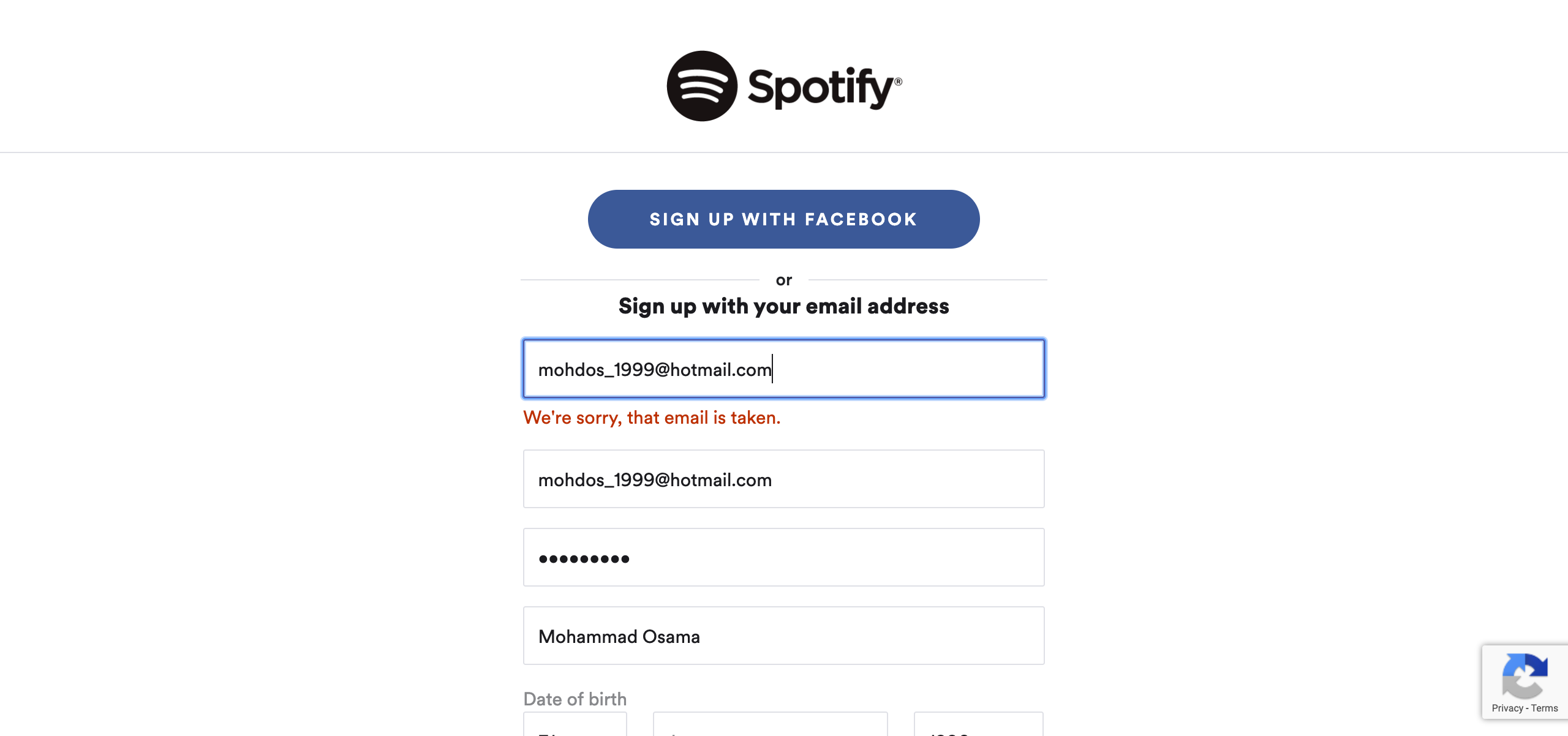
Signing up with unselected gender

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected gender



## Test Teardown

[Fixture] setup::0

# Signup with unselected month - passed

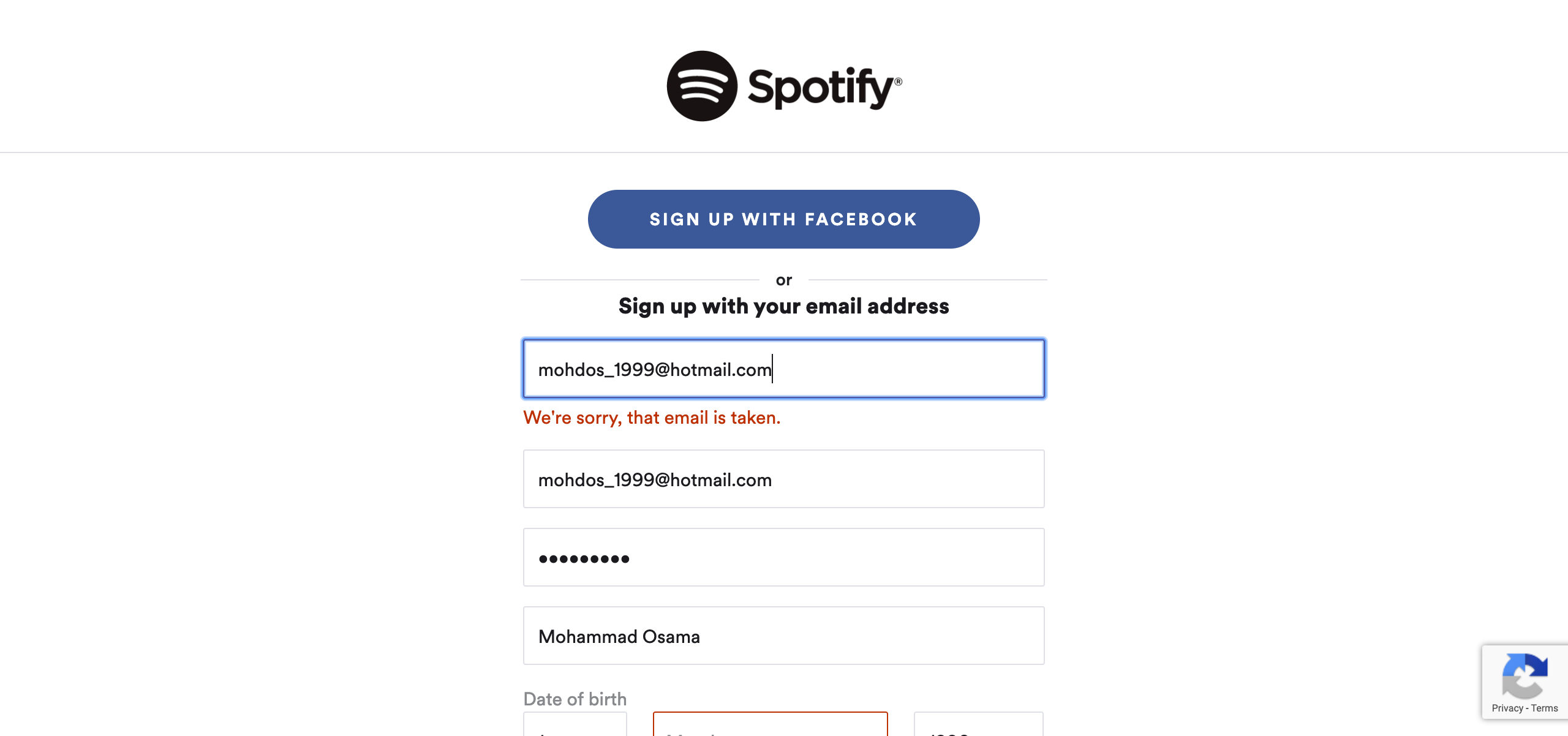
Signing up with unselected month

## Test Setup

[Fixture] setup

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected month



## Test Teardown

[Fixture] setup::0

# Signup with unselected year - passed

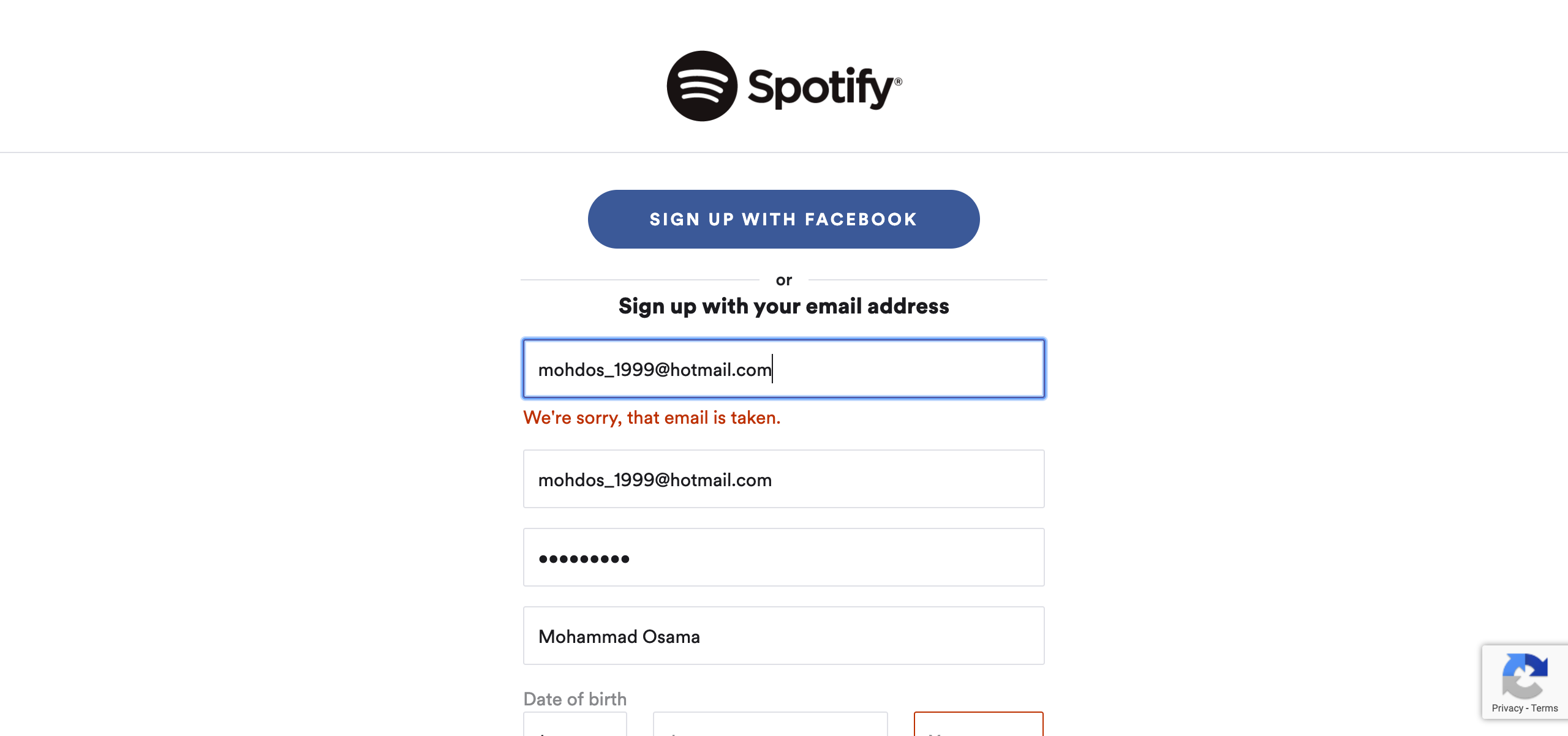
Signing up with unselected year

## Test Setup

[Fixture] setup\_final

## Test Body

[Attachment] SUCCESS: Sign up stopped with unselected year



## Test Teardown

[Fixture] setup\_final::0