A Project Report On

Software Testing and Quality Assurance (Mini Project II)

SUBMITTED BY

Abhishek Sawalkar **Roll No: 41402**

Sanket Bhatlawande **Roll No: 41415**

Samarth Bhadane **Roll No: 41414**

CLASS: BE-4

GUIDED BY

Prof. K.C. Waghmare

DEPARTMENT OF COMPUTER ENGINEERING

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DHANKAWADI, PUNE-43

SAVITRIBAI PHULE PUNE UNIVERSITY

2021-22

Title:

Create a small web-based application by selecting relevant system environment/platform and programming languages. Narrate concise Test Plan consisting features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing

Problem Definition:

Perform Web testing and identify the bugs using Selenium WebDriver and IDEand generate test reports encompassing exploratory testing on a self developed web app.

Objective

Perform testing on a Netflixmodified site and write test cases.

Test Environment:

An Ubuntu 20.04 environment

Django 2.0

Selenium web-driver

Selenium IDE

Google Chrome

Theory:

Selenium:

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. Selenium is a suite of software tools to automate Web Browsers. It is an Open source suite of tools mainly used for Functional and Regression Test Automation. Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.It is quite similar to HP Quick Test Pro (QTP now UFT) only that

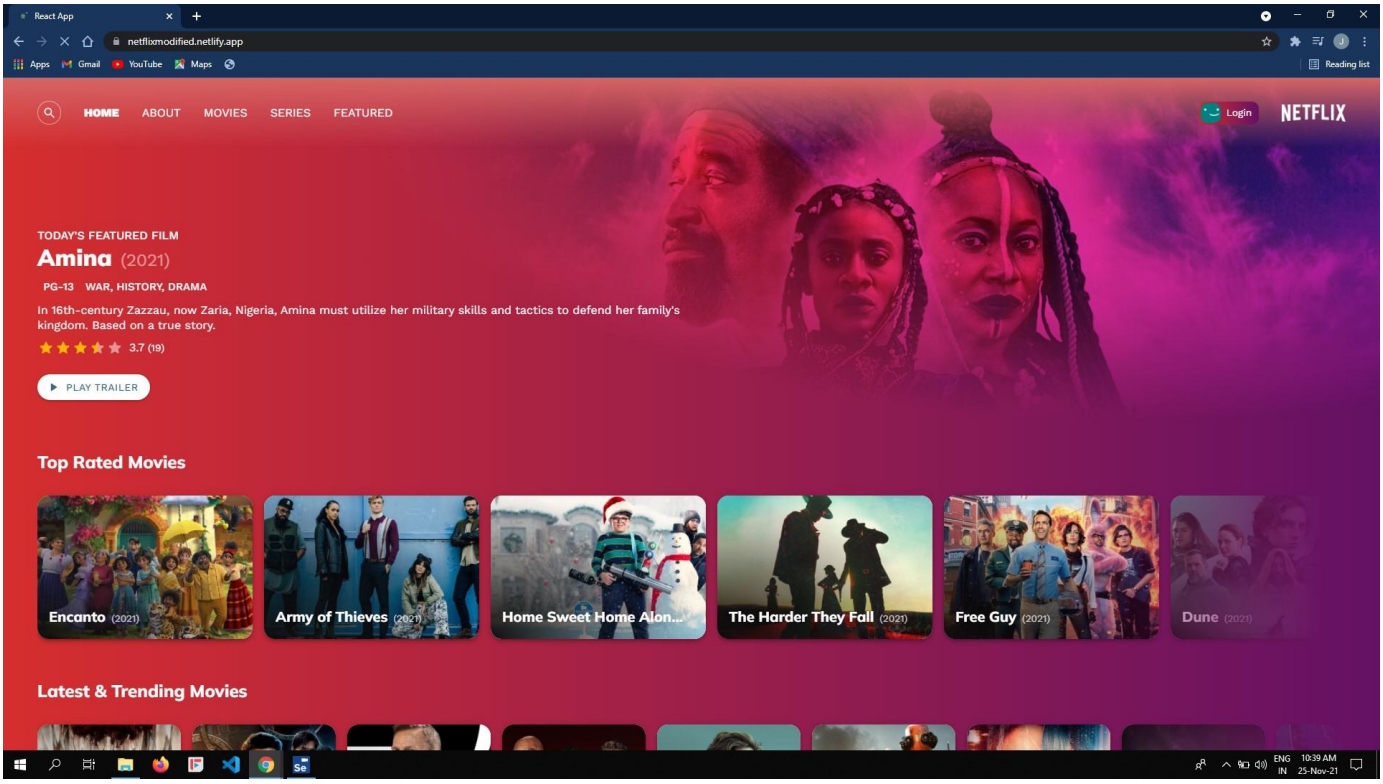
Selenium focuses on automating web-based applications. Testing done using a Selenium tool is usually referred as Selenium Testing.

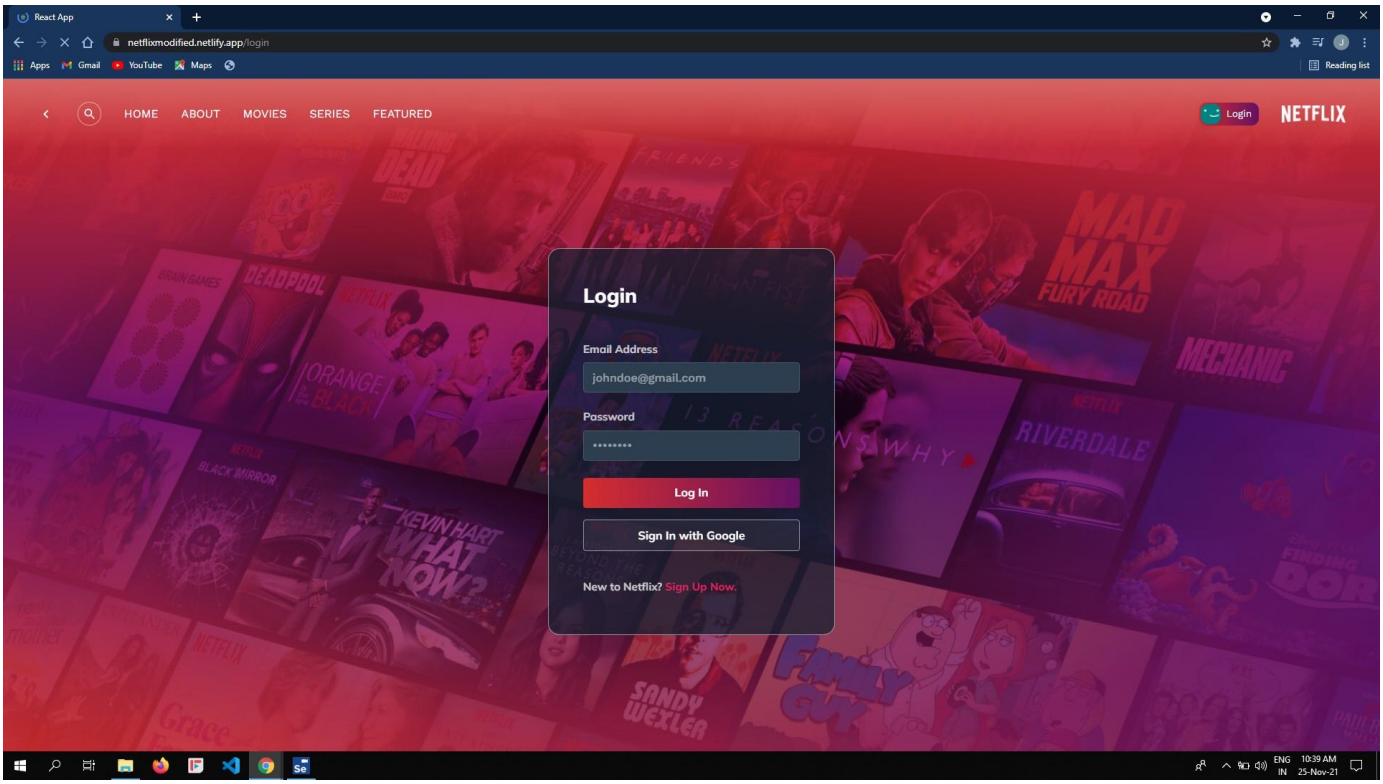
Selenium IDE:

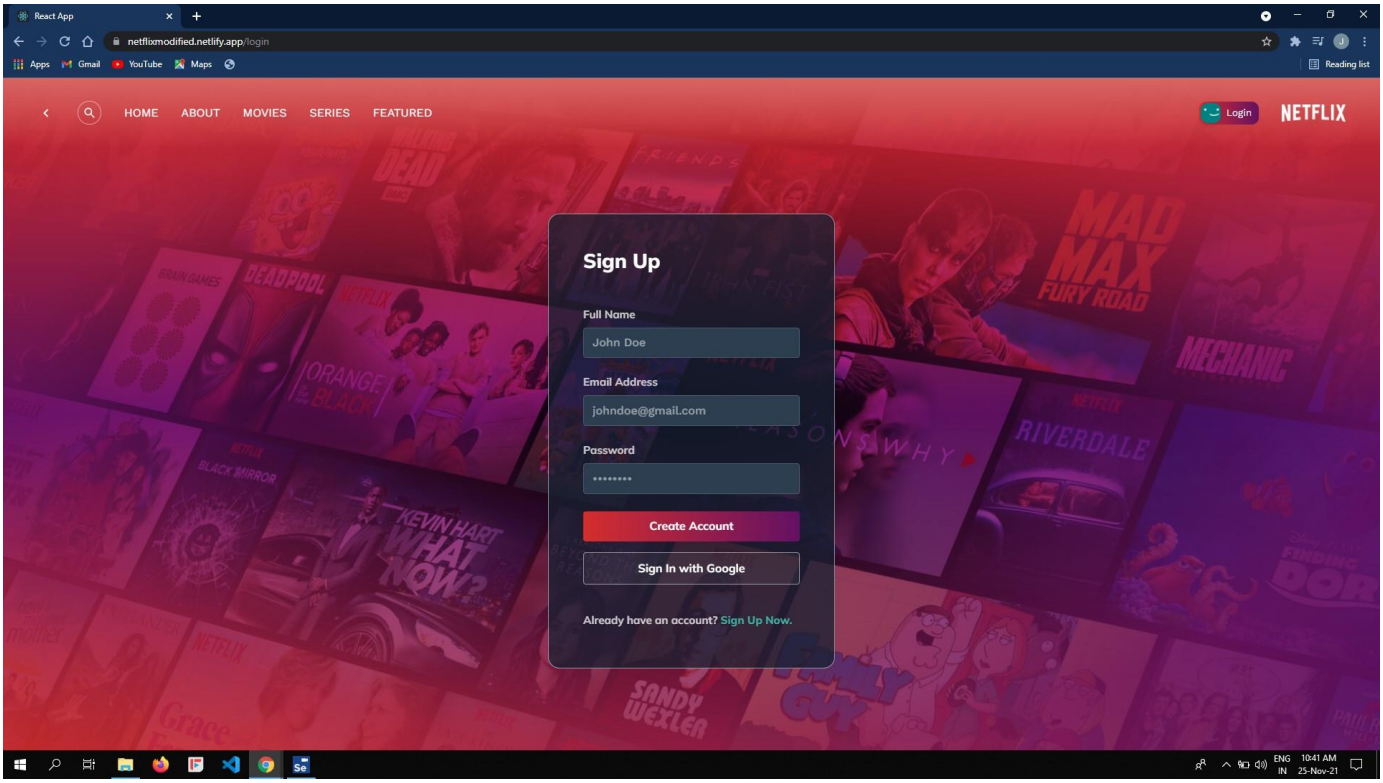
Selenium IDE (Integrated Development Environment) is primarily a record/run tool that a test case developer uses to develop Selenium Test cases. Selenium IDE is an easy to use tool from the Selenium Test Suite and can even be used by someone new to developing automated test cases for their web applications. One does not require any special setup to get started with Selenium IDE. You just need to add the extension of your specific browser. Selenium IDE provides you with a GUI (Graphical User Interface) for easily recording your interactions with the website.

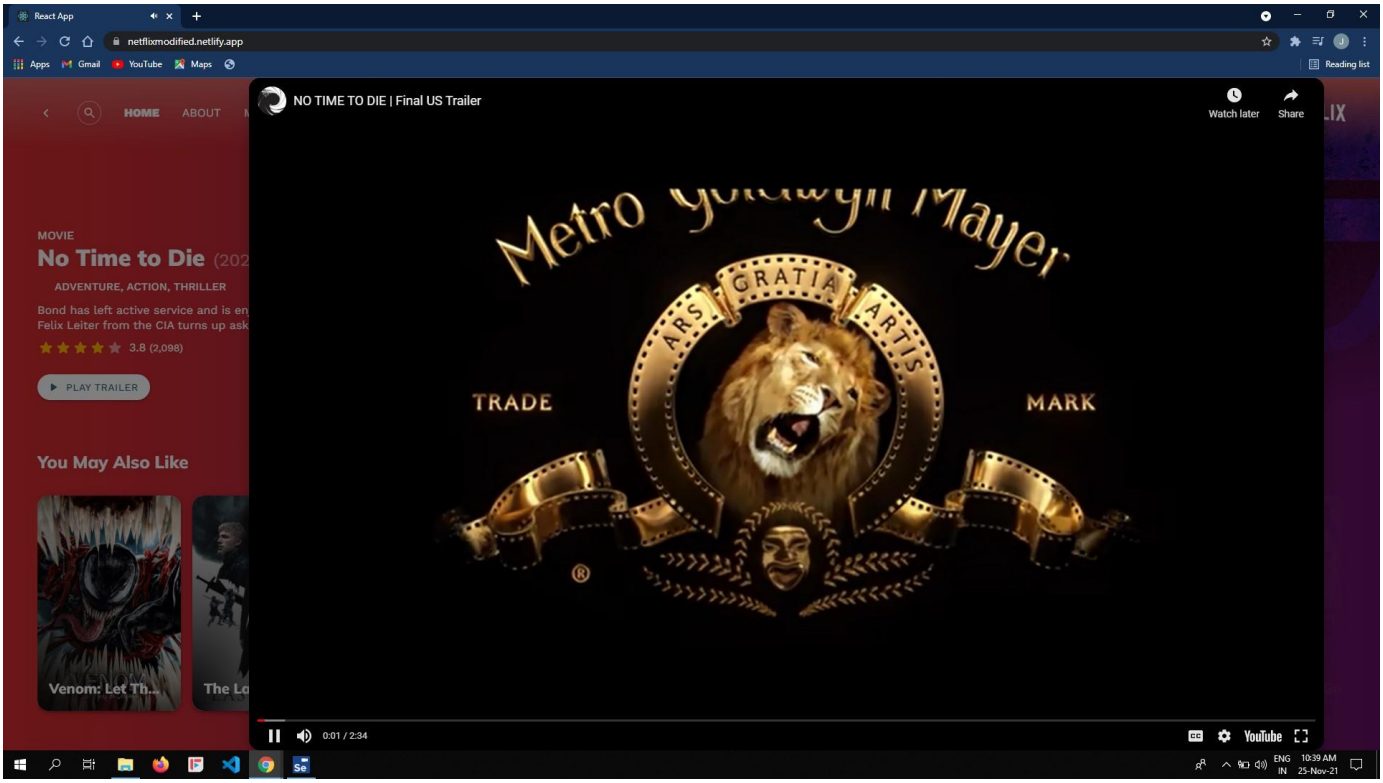
Selenium IDE allows a user or a test case developer to create the test cases and test suites and edit it later as per their requirements. The development environment also provides the capability of converting test cases to different programming languages, which makes it easier for the user and does not mandate the need for knowing a specific programming language.

Sample Screenshots of application



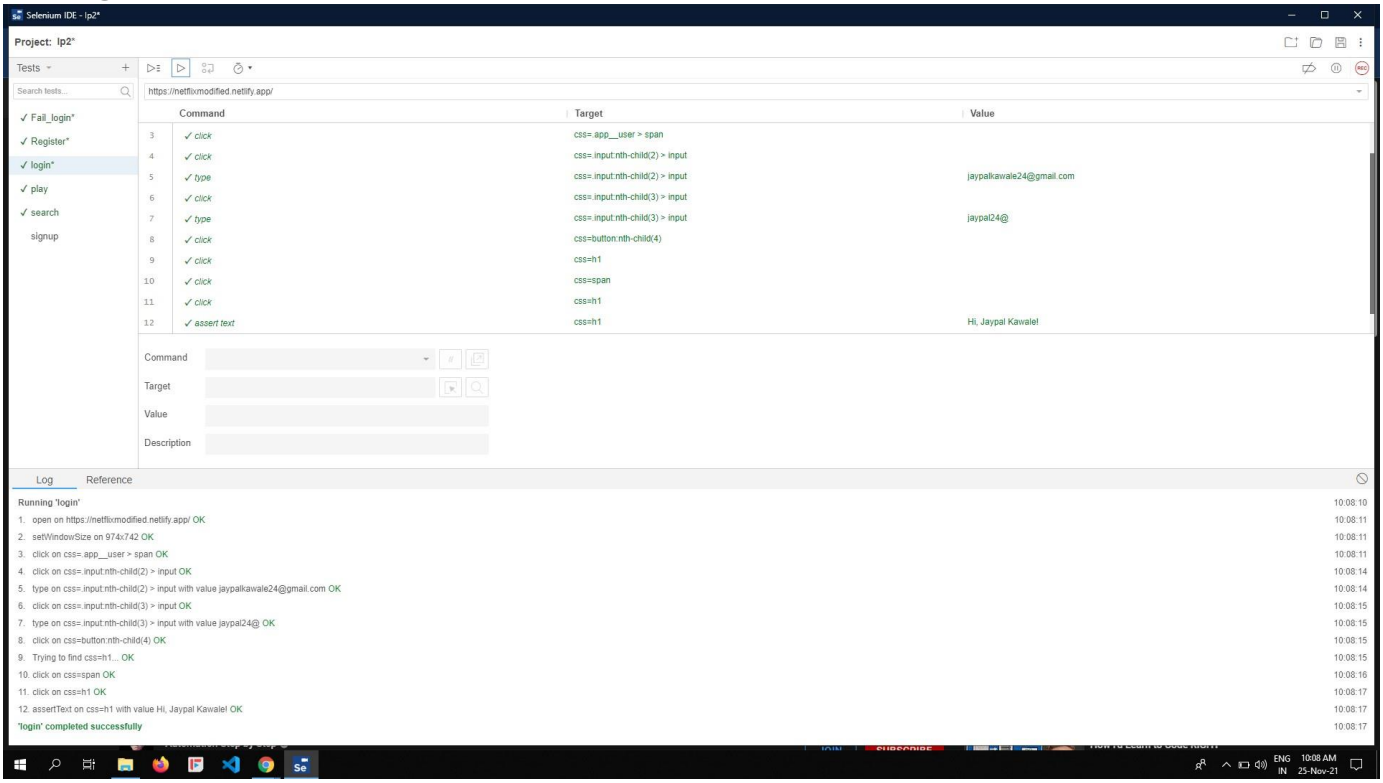

React App
netflixmodified.netlify.app/login
Apps
Gmail
YouTube
Maps
Reading list
HOME
ABOUT
MOVIES
SERIES
FEATURED
Login
NETFLIX
Login
Email Address
johndoe@gmail.com
Password
Log In
Sign In with Google
New to Netflix? Sign Up Now.
ENG
IN
10:39 AM
25-Nov-21


React App
netflixmodified.netlify.app/login
Apps
Gmail
YouTube
Maps
Reading list
HOME
ABOUT
MOVIES
SERIES
FEATURED
Login
NETFLIX
Sign Up
Full Name
John Doe
Email Address
johndoe@gmail.com
Password
Create Account
Sign In with Google
Already have an account? Sign Up Now.

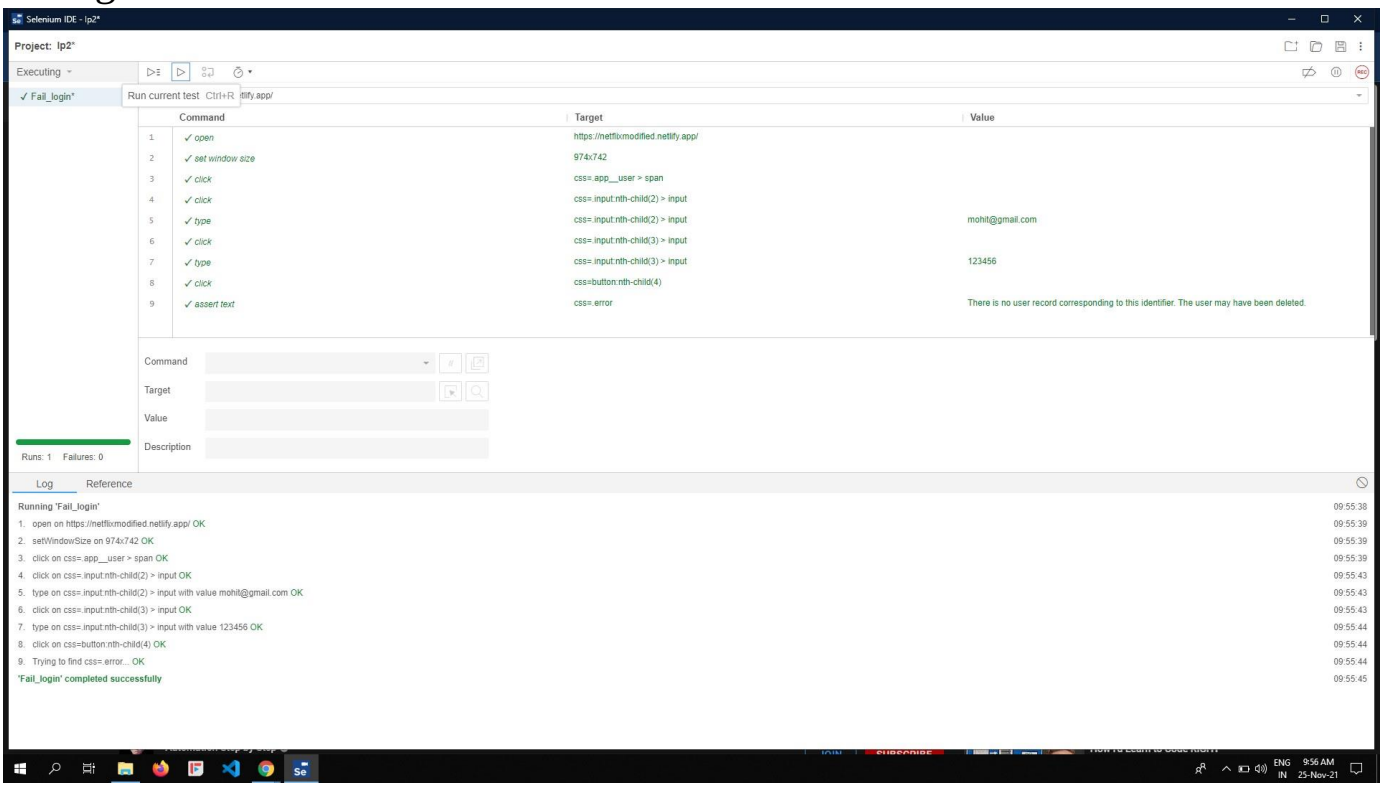

React App
netflixmodified.netlify.app
Apps
Gmail
YouTube
Maps
Reading list
HOME
ABOUT
NO TIME TO DIE | Final US Trailer
Watch later
Share
MOVIE
No Time to Die
ADVENTURE, ACTION, THRILLER
3.8 (2,098)
PLAY TRAILER
You May Also Like
Venom: Let Th...
TRADE
MARK
0:01 / 2:34
YouTube

Output logs of sample tests

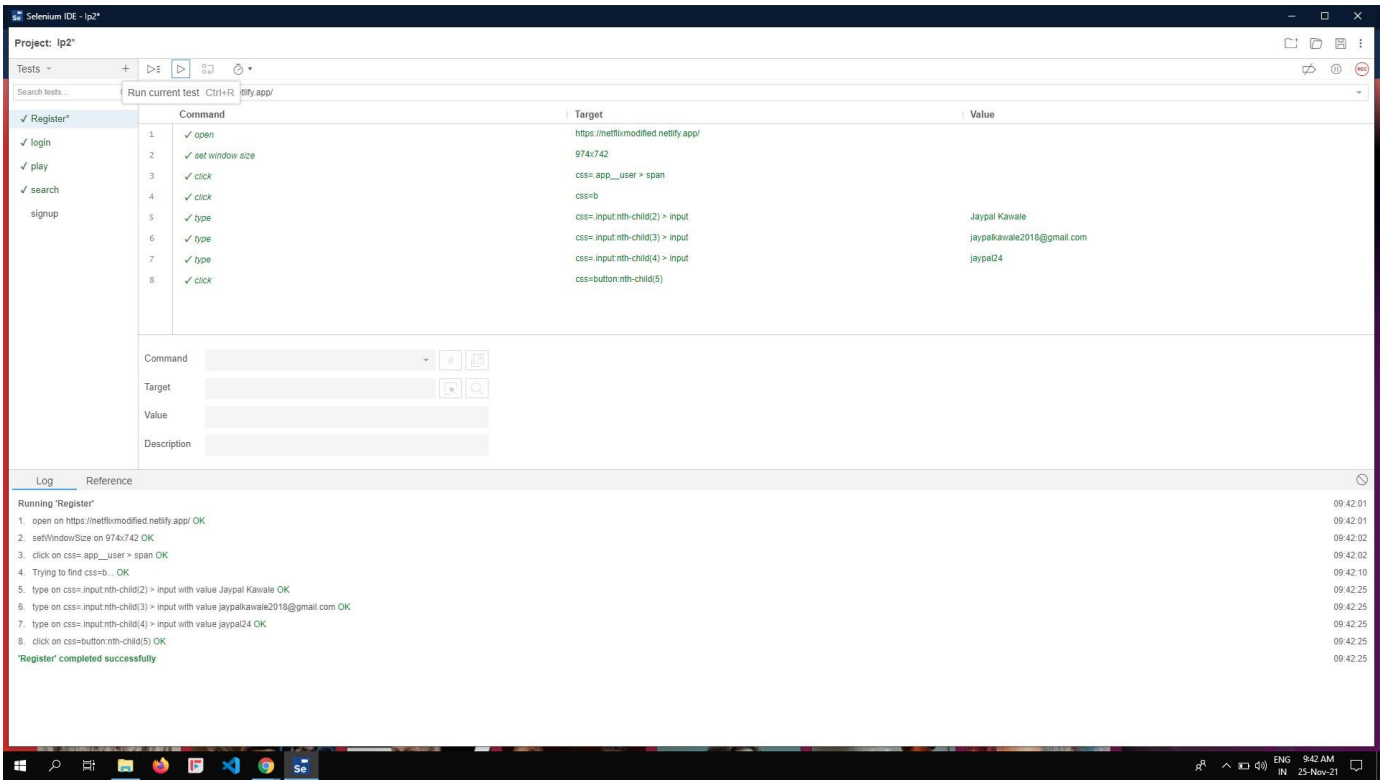
1. Login with valid credential



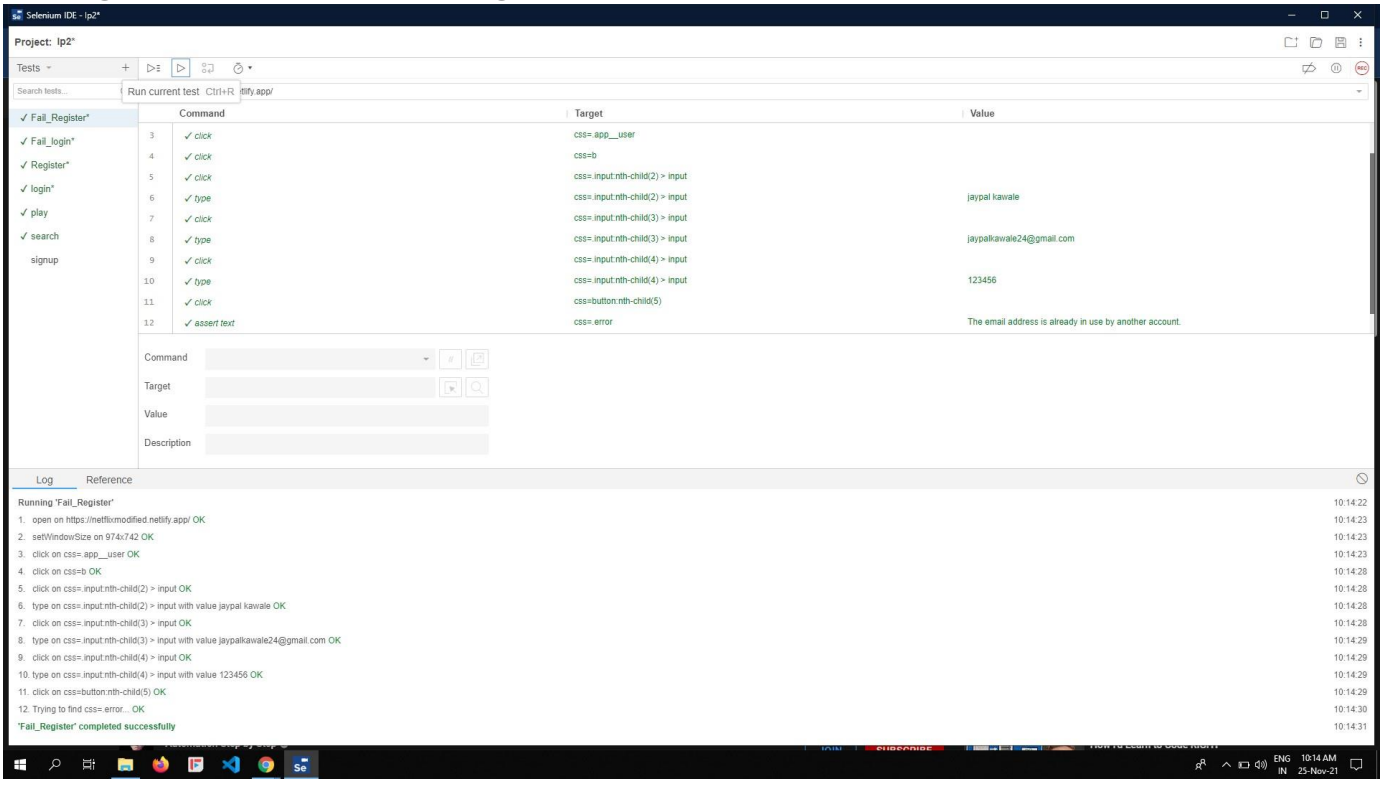
2. Login with invalid credential



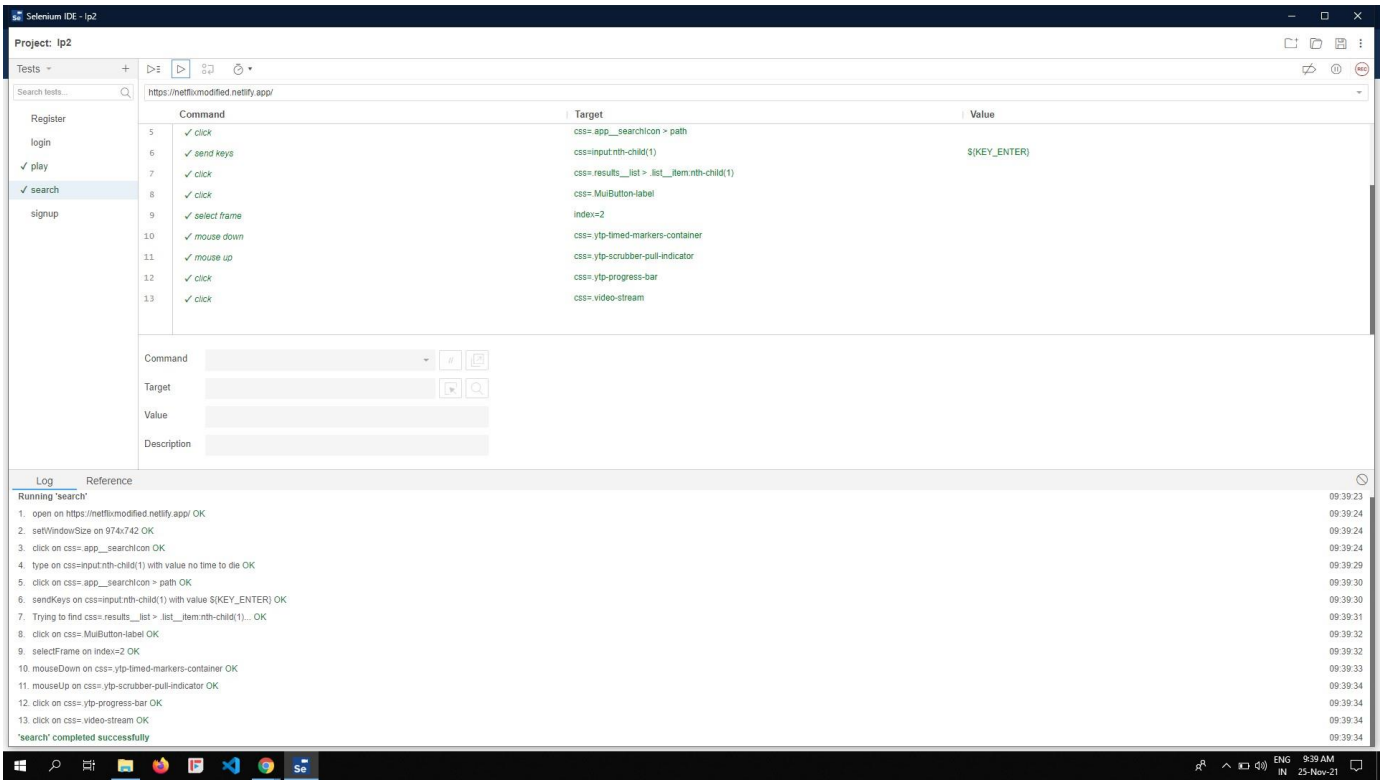
3. Register User with new Credential



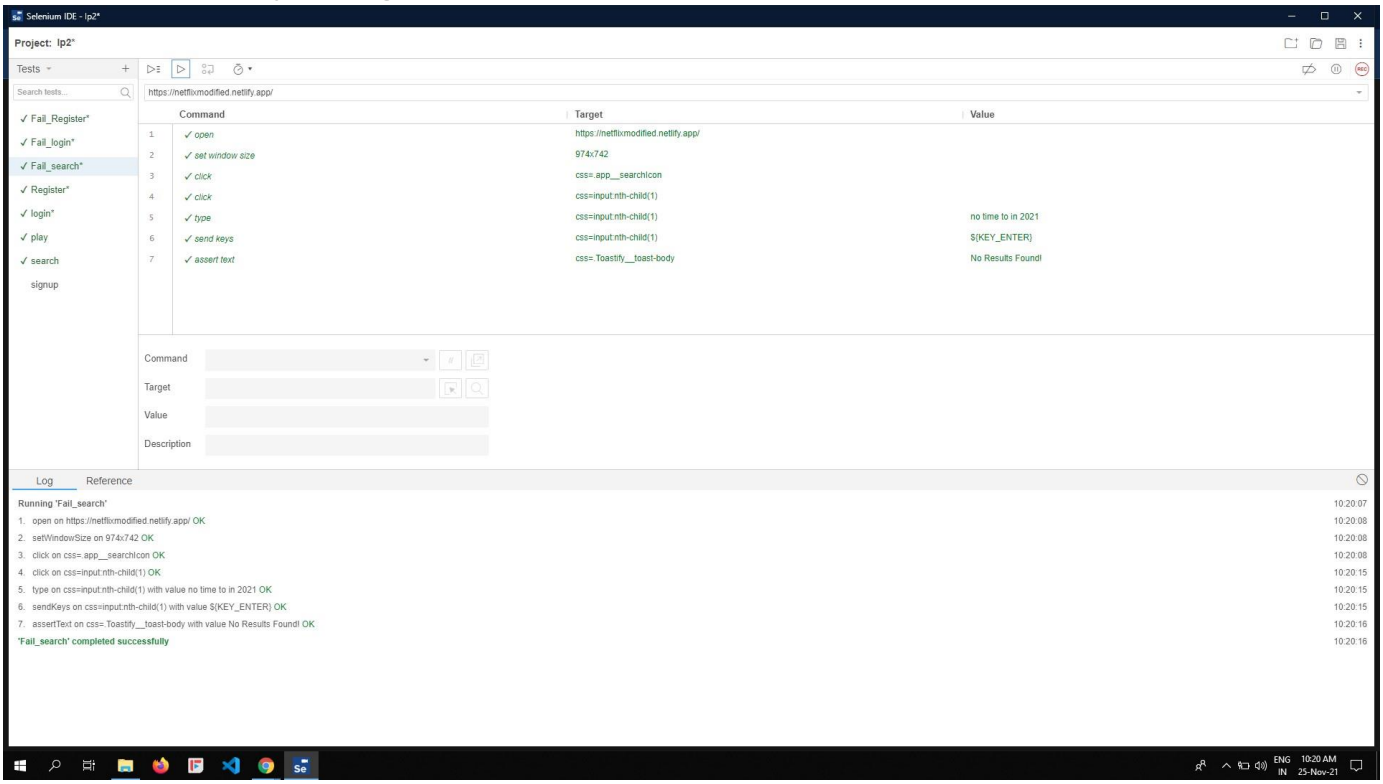
4. Register User with existing credentials



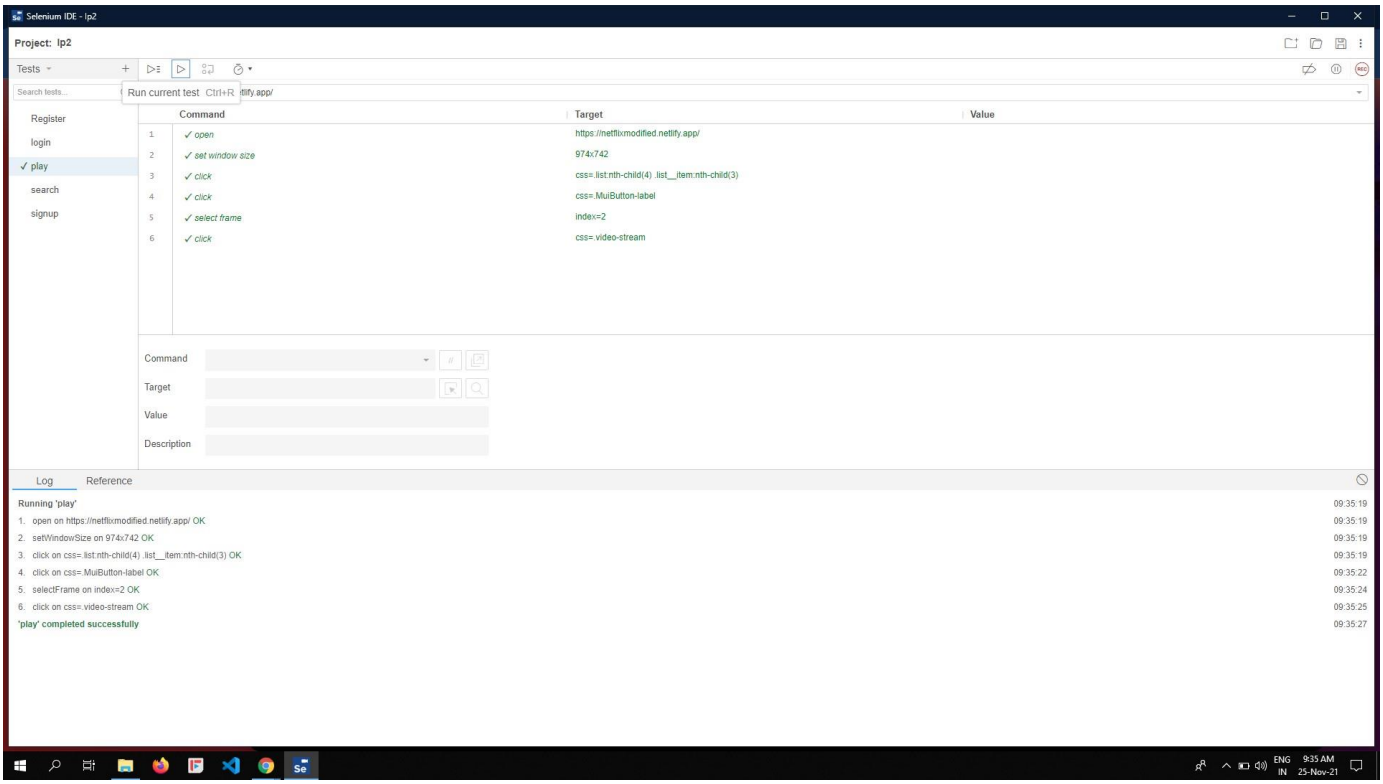
5. Search a Movie by name



6. search movie by wrong name



7. Play movie Trailer


Selenium IDE - lp2
Project: lp2
Tests
Run current test Ctrl+R
tlify.app/
Register
login
play
search
signup
Command
Target
Value
1 open https://netflixmodified.netlify.app/
2 set window size 974x742
3 click css=.list:nth-child(4) .list__item:nth-child(3)
4 click css=.MuiButton-label
5 select frame index=2
6 click css=.video-stream
Command
Target
Value
Description
Log
Reference
Running 'play'
1. open on https://netflixmodified.netlify.app/ OK
2. setWindowSize on 974x742 OK
3. click on css=.list:nth-child(4) .list__item:nth-child(3) OK
4. click on css=.MuiButton-label OK
5. selectFrame on index=2 OK
6. click on css=.video-stream OK
'play' completed successfully
09:35:19
09:35:19
09:35:19
09:35:19
09:35:22
09:35:24
09:35:25
09:35:27
ENG
IN
9:35 AM
25-Nov-21

Source code/ Functions of the application

Login with valid credential

```
# Generated by Selenium IDE import pytest import time import json from
selenium import webdriver from selenium.webdriver.common.by import By from
selenium.webdriver.common.action_chains import ActionChains from
selenium.webdriver.support import expected_conditions from
selenium.webdriver.support.wait import WebDriverWait from
selenium.webdriver.common.keys import Keys from
selenium.webdriver.common.desired_capabilities import DesiredCapabilities


class TestLogin():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_login(self):
    self.driver.get("https://netflixmodified.netlify.app/")
self.driver.set_window_size(974, 742) self.driver.find_element(By.CSS_SELECTOR,
".app__user > span").click() self.driver.find_element(By.CSS_SELECTOR,
".input:nth-child(2) > input").click() self.driver.find_element(By.CSS_SELECTOR,
".input:nth-child(2) > input").send_keys("jaypalkawale24@gmail.com")
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) > input").click()
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) >
input").send_keys("jaypal24@") self.driver.find_element(By.CSS_SELECTOR,
"button:nth-child(4)").click() self.driver.find_element(By.CSS_SELECTOR,
"h1").click() self.driver.find_element(By.CSS_SELECTOR, "span").click()
```

self.driver.find_element(By.CSS_SELECTOR, "h1").click() assert self.driver.find_element(By.CSS_SELECTOR, "h1").text == "Hi, Jaypal Kawale!"

Login with invalid credentials

Generated by Selenium IDE import pytest import time import json from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.action_chains import ActionChains from selenium.webdriver.support import expected_conditions from selenium.webdriver.support.wait import WebDriverWait from selenium.webdriver.common.keys import Keys from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestFaillogin():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()
  def test_faillogin(self):
    self.driver.get("https://netflixmodified.netlify.app/") self.driver.set_window_size(974, 742) self.driver.find_element(By.CSS_SELECTOR, ".app__user > span").click() self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(2) > input").click() self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(2) > input").send_keys("mohit@gmail.com") self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) > input").click() self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) > input").send_keys("123456") self.driver.find_element(By.CSS_SELECTOR, "button:nth-child(4)").click() assert self.driver.find_element(By.CSS_SELECTOR, ".error").text == "There is no user record corresponding to this identifier. The user may have been deleted."

Register New User

```
# Generated by Selenium IDE import pytest import time import json from
selenium import webdriver from selenium.webdriver.common.by import By from
selenium.webdriver.common.action_chains import ActionChains from
selenium.webdriver.support import expected_conditions from
selenium.webdriver.support.wait import WebDriverWait from
selenium.webdriver.common.keys import Keys from
selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestRegister():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_register(self):
    self.driver.get("https://netflixmodified.netlify.app/") self.driver.set_window_size(974, 742)
    self.driver.find_element(By.CSS_SELECTOR, ".app__user > span").click()
    self.driver.find_element(By.CSS_SELECTOR, "b").click()
    self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(2) >
    input").send_keys("Jaypal
Kawale")
    self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) >
input").send_keys("jaypalkawale2018@gmail.com")
    self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(4) >
input").send_keys("jaypal24")
    self.driver.find_element(By.CSS_SELECTOR, "button:nth-child(5)").click()
```

Register Existing User

```
# Generated by Selenium IDE import pytest import time import json from
selenium import webdriver from selenium.webdriver.common.by import By from
selenium.webdriver.common.action_chains import ActionChains from
selenium.webdriver.support import expected_conditions from
selenium.webdriver.support.wait import WebDriverWait from
selenium.webdriver.common.keys import Keys from
selenium.webdriver.common.desired_capabilities import DesiredCapabilities
class TestFailRegister():

  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}


  def teardown_method(self, method):
    self.driver.quit()


  def test_failRegister(self):
    self.driver.get("https://netflixmodified.netlify.app/") self.driver.set_window_size(974, 742)
self.driver.find_element(By.CSS_SELECTOR, ".app__user").click()
self.driver.find_element(By.CSS_SELECTOR, "b").click()
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(2) > input").click()
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(2) > input").send_keys("jaypal
kawale") self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) > input").click()
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(3) >
input").send_keys("jaypalkawale24@gmail.com")
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(4) > input").click()
self.driver.find_element(By.CSS_SELECTOR, ".input:nth-child(4) >
input").send_keys("123456") self.driver.find_element(By.CSS_SELECTOR, "button:nth-
```

child(5)").click() assert self.driver.find_element(By.CSS_SELECTOR, ".error").text == "The email address is already in use by another account."

Search a Movie by Name

Generated by Selenium IDE import pytest import time import json from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.action_chains import ActionChains from selenium.webdriver.support import expected_conditions from selenium.webdriver.support.wait import WebDriverWait from selenium.webdriver.common.keys import Keys from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

```
class TestSearch():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_search(self):
    self.driver.get("https://netflixmodified.netlify.app/") self.driver.set_window_size(974, 742)
self.driver.find_element(By.CSS_SELECTOR, ".app__searchIcon").click()
self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)").send_keys("no time to die")
self.driver.find_element(By.CSS_SELECTOR, ".app__searchIcon > path").click()
self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)").send_keys(Keys.ENTER)
self.driver.find_element(By.CSS_SELECTOR, ".results__list > .list__item:nth-child(1)").click()
self.driver.find_element(By.CSS_SELECTOR, ".MuiButton-label").click()
self.driver.switch_to.frame(2) element = self.driver.find_element(By.CSS_SELECTOR, ".ytp-timed-markers-container") actions = ActionChains(self.driver)
```

actions.move_to_element(element).click_and_hold().perform() element = self.driver.find_element(By.CSS_SELECTOR, ".ytp-scrubber-pull-indicator") actions = ActionChains(self.driver) actions.move_to_element(element).release().perform() self.driver.find_element(By.CSS_SELECTOR, ".ytp-progress-bar").click() self.driver.find_element(By.CSS_SELECTOR, ".video-stream").click()

search a random text

Generated by Selenium IDE import pytest import time import json from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.action_chains import ActionChains from selenium.webdriver.support import expected_conditions from selenium.webdriver.support.wait import WebDriverWait from selenium.webdriver.common.keys import Keys from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestFailsearch():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}

  def teardown_method(self, method):
    self.driver.quit()

  def test_failsearch(self):
    self.driver.get("https://netflixmodified.netlify.app/") self.driver.set_window_size(974, 742)
    self.driver.find_element(By.CSS_SELECTOR, ".app__searchIcon").click()
    self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)").click()
    self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)").send_keys("no time to
    in

2021") self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)").send_keys(Keys.ENTER) assert self.driver.find_element(By.CSS_SELECTOR, ".Toastify__toast-body").text == "No Results Found!"

Play Trailer

Generated by Selenium IDE import pytest import time import json from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.action_chains import ActionChains from selenium.webdriver.support import expected_conditions from selenium.webdriver.support.wait import WebDriverWait from selenium.webdriver.common.keys import Keys from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

```
class TestPlay():
  def setup_method(self, method):
    self.driver = webdriver.Chrome()
    self.vars = {}
  def teardown_method(self, method):
    self.driver.quit()

  def test_play(self):
    self.driver.get("https://netflixmodified.netlify.app/")
    self.driver.set_window_size(974, 742)
    self.driver.find_element(By.CSS_SELECTOR, ".list:nth-child(4)
.list__item:nth-child(3)").click()
    self.driver.find_element(By.CSS_SELECTOR, ".MuiButton-label").click()
    self.driver.switch_to.frame(2)
    self.driver.find_element(By.CSS_SELECTOR, ".video-stream").click()
```

Conclusion :

Performed automation testing on a self developed Netflixmodified site and verified that no bugs or defects were found.