# Selenium Automation Testing: Hotstar Sports Section

## Introduction

- **Objective:** To automate testing tasks on the Hotstar Sports section using Selenium.

- **Tools Used**: Selenium WebDriver, GeckoDriver, Firefox Browser.

## Prerequisites

### Environmental SetUp:

- Selenium WebDriver installed.
- GeckoDriver executable downloaded and placed in 'C:\Users\shakt\Downloads\gecko'.
- Firefox browser installed.

## Selenium Automation Workflow

### 1. Setting Up the Environment

- Imported necessary libraries:os for environment variables.
- Selenium for WebDriver automation.
- Set the path to the GeckoDriver executable.
- Suppressed Firefox logging warning.
- Created a FirefoxService instance with the specified path.

### 2. WebDriver Initialization

- Created a Firefox WebDriver instance and specified the service.

**3. Navigating to Hotstar Sports Section**

- Used the WebDriver to navigate directly to the Sports section on Hotstar (https://www.hotstar.com/in/sports).

**4. Searching for Cricket Content**

- Checked for the presence of a search input field.
- Entered "Cricket" as a search query and submitted the search.

**5. Handling Search Results**

- Utilized WebDriverWait to wait for search results (up to 10 seconds).
- Located the first search result with the class name "episode-item."

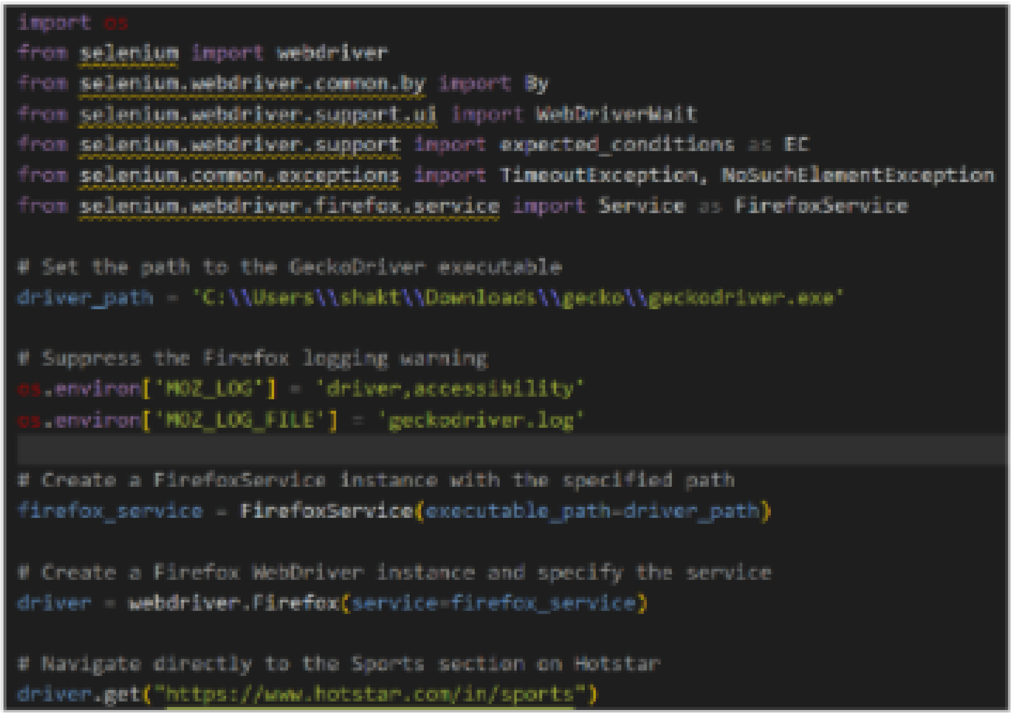
### 6. Handling Exceptions

- Implemented exception handling to deal with potential errors:NoSuchElementException: When the search input is not found.
- TimeoutException: When the search results take too long to load.

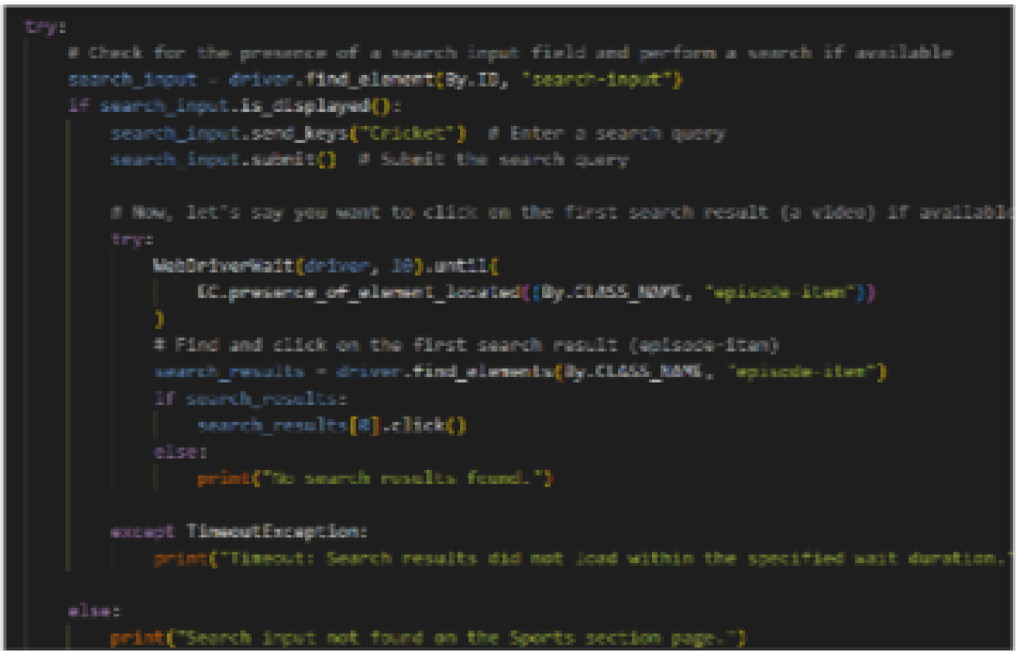
### 7. Clean-Up

- Ensured the WebDriver was closed after completing the automation.

## Code:

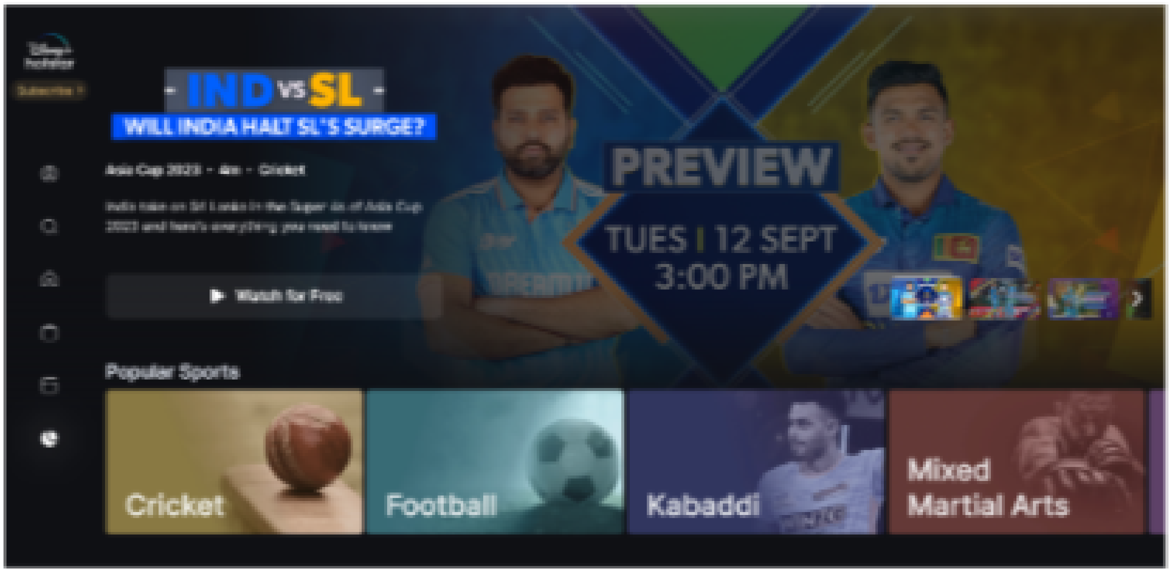


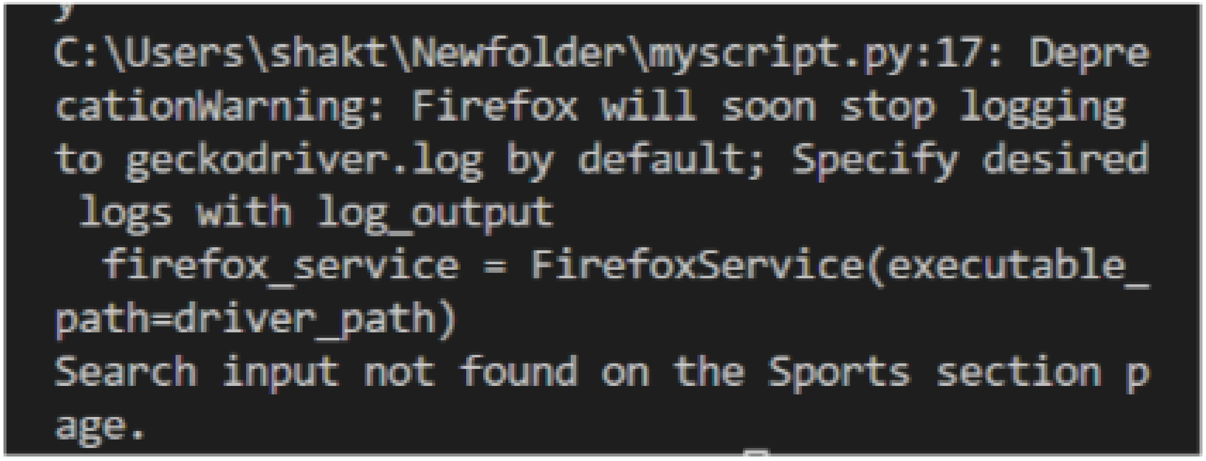
```python
import os
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException, NoSuchElementException
from selenium.webdriver.firefox.service import Service as FirefoxService

# Set the path to the GeckoDriver executable
driver_path = 'C:\\Users\\shakt\\Downloads\\gecko\\geckodriver.exe'

# Suppress the Firefox logging warning
os.environ['MOZ_LOG'] = 'driver,accessibility'
os.environ['MOZ_LOG_FILE'] = 'geckodriver.log'

# Create a FirefoxService instance with the specified path
firefox_service = FirefoxService(executable_path=driver_path)

# Create a Firefox WebDriver instance and specify the service
driver = webdriver.Firefox(service=firefox_service)

# Navigate directly to the Sports section on Hotstar
driver.get("https://www.hotstar.com/in/sports")
```



```python
try:
    # Check for the presence of a search input field and perform a search if available
    search_input = driver.find_element(By.ID, "search-input")
    if search_input.is_displayed():
        search_input.send_keys("Cricket")  # Enter a search query
        search_input.submit()  # Submit the search query

        # Now, let's say you want to click on the first search result (a video) if available
        try:
            WebDriverWait(driver, 10).until(
                EC.presence_of_element_located((By.CLASS_NAME, "episode-item"))
            )
            # Find and click on the first search result (episode-item)
            search_results = driver.find_elements(By.CLASS_NAME, "episode-item")
            if search_results:
                search_results[0].click()
            else:
                print("No search results found.")

        except TimeoutException:
            print("Timeout: Search results did not load within the specified wait duration."

    else:
        print("Search input not found on the Sports section page.")
```

Output:





```
C:\Users\shakt\Newfolder\myscript.py:17: Depre
cationWarning: Firefox will soon stop logging
to geckodriver.log by default; Specify desired
 logs with log_output
  firefox_service = FirefoxService(executable_
path=driver_path)
Search input not found on the Sports section p
age.
```

## Conclusion

- Successfully automated the testing of the Hotstar Sports section using Selenium WebDriver.
- Demonstrated techniques for handling exceptions and waiting for elements to load.
- This automation process can be extended to cover more test scenarios and integrated into a broader test suite.