Software Engineering Project

E-commerce Shopping Price Comparison Website

GROUP MEMBERS:-

Suryansh Sahay (**202251137**)
Susmit Kumar Bharti (**202251138**)
Smita Patel (2**02251129**)
Sarah Neha Mortha (**202251118**)
Shailly Yadav (**202251123**)
Tanishka Sharma (**202251140**)

Project Details:-

The project is **a web-based tool** for comparing prices amongst different on-line suppliers of electronic goods in order to assist users **in finding the best discounts**. The website features an easy-to-use design that **helps users compare costs** and see what offers are available for items like laptops, tablets, smartphones, cameras, and smartwatches.

Key features of the platform include:
Price Comparison: Users can select an electronic product and view its prices across different online platforms, such as Amazon, Flipkart, eBay, and Best Buy. This allows users to quickly find the best price available for the product they are interested in.

Links and Navigation: There is a menu on the website that leads to several parts, including contact, price comparison, about, and home. Links to Google Maps locations and social media pages can also be followed by users.

Contact Form: Users can send messages or inquiries using the contact form on the website. To manage form submissions and transmit the data to a Google Sheet, the form makes use of Google Apps Script.

Responsive Design: The website is designed to be responsive and adjusts its layout according to different screen sizes, ensuring a seamless experience for users on various devices such as smartphones, tablets, and desktop computers.

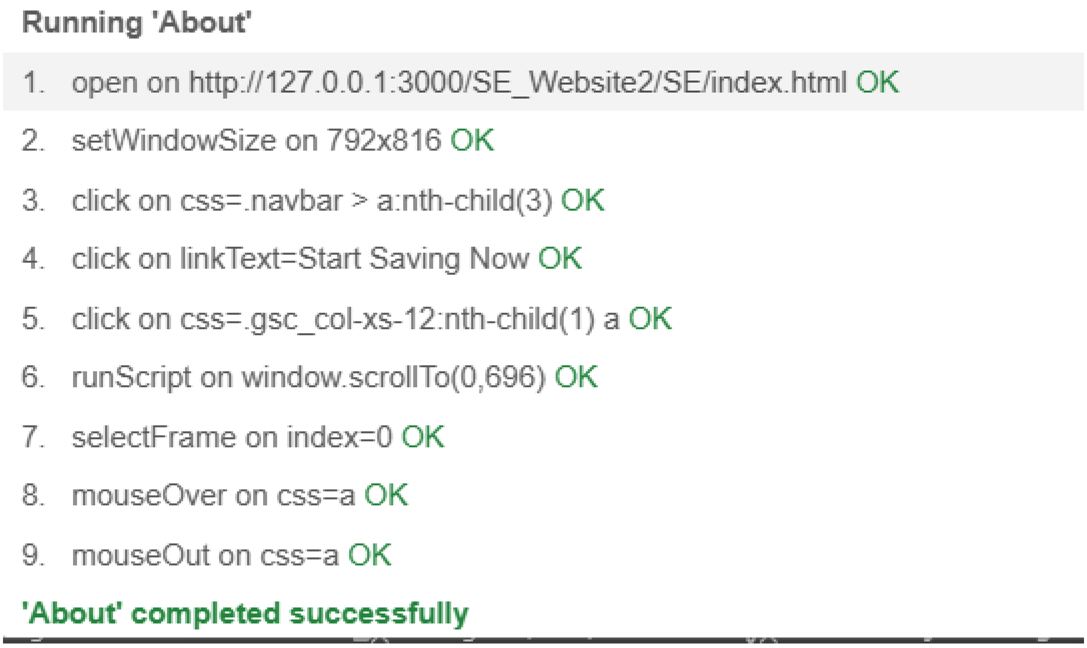
Social Media Integration: The website's footer provides links to Facebook, Instagram, LinkedIn, Twitter, and other social media sites. This makes it easier for users to access updates and promotions and stay in touch with the platform.

Map Integration: Users can find directions and contact details for physical stores and offices by clicking on links in the footer that lead to Google Maps locations for different branches.

The project's overall goal is to give consumers an easy and quick way to compare electronic product prices from several online retailers, enabling them to make wise decisions and save money.

```
# 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 TestAbout():
  def setup_method(self, method):
    self.driver = webdriver.Remote(command_executor='http://localhost:4444/wd/hub',
desired_capabilities=DesiredCapabilities.CHROME)
    self.vars = {}

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

  def test_about(self):
    self.driver.get("http://127.0.0.1:3000/SE_Website2/SE/index.html")
    self.driver.set_window_size(792, 816)
    self.driver.find_element(By.CSS_SELECTOR, ".navbar > a:nth-child(3)").click()
    self.driver.find_element(By.LINK_TEXT, "Start Saving Now").click()
    self.driver.find_element(By.CSS_SELECTOR, ".gsc_col-xs-12:nth-child(1) a").click()
    self.driver.execute_script("window.scrollTo(0,696)")
    self.driver.switch_to.frame(0)
    element = self.driver.find_element(By.CSS_SELECTOR, "a")
    actions = ActionChains(self.driver)
    actions.move_to_element(element).perform()
    element = self.driver.find_element(By.CSS_SELECTOR, "body")
    actions = ActionChains(self.driver)
    actions.move_to_element(element, 0, 0).perform()
```

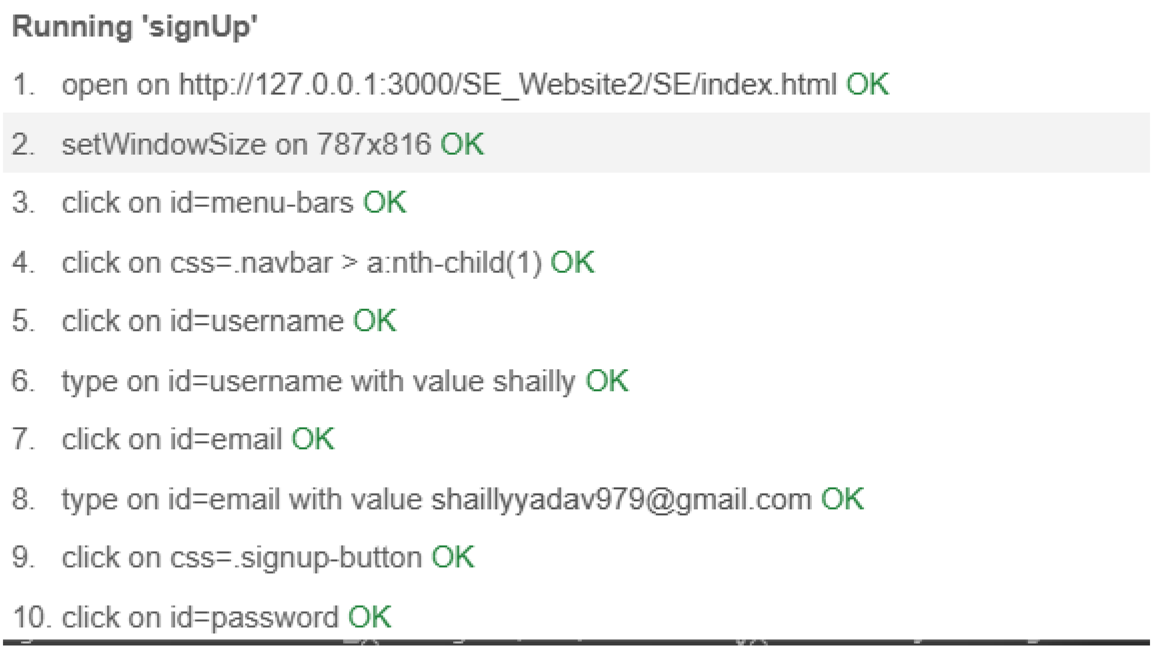
OUTPUT:



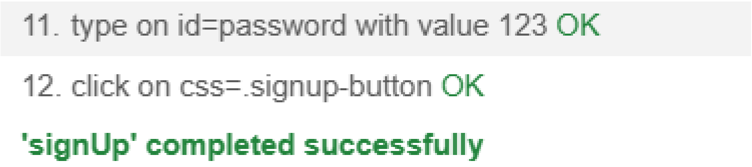
```
Running 'About'
1.  open on http://127.0.0.1:3000/SE_Website2/SE/index.html OK
2.  setWindowSize on 792x816 OK
3.  click on css=.navbar > a:nth-child(3) OK
4.  click on linkText=Start Saving Now OK
5.  click on css=.gsc_col-xs-12:nth-child(1) a OK
6.  runScript on window.scrollTo(0,696) OK
7.  selectFrame on index=0 OK
8.  mouseOver on css=a OK
9.  mouseOut on css=a OK
'About' completed successfully
```

```
# 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 TestSignUp():
  def setup_method(self, method):
    self.driver = webdriver.Remote(command_executor='http://localhost:4444/wd/hub',
desired_capabilities=DesiredCapabilities.CHROME)
    self.vars = {}

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

  def test_signUp(self):
    self.driver.get("http://127.0.0.1:3000/SE_Website2/SE/index.html")
    self.driver.set_window_size(787, 816)
    self.driver.find_element(By.ID, "menu-bars").click()
    self.driver.find_element(By.CSS_SELECTOR, ".navbar > a:nth-child(1)").click()
    self.driver.find_element(By.ID, "username").click()
    self.driver.find_element(By.ID, "username").send_keys("shailly")
    self.driver.find_element(By.ID, "email").click()
    self.driver.find_element(By.ID, "email").send_keys("shaillyyadav979@gmail.com")
    self.driver.find_element(By.CSS_SELECTOR, ".signup-button").click()
    self.driver.find_element(By.ID, "password").click()
    self.driver.find_element(By.ID, "password").send_keys("123")
    self.driver.find_element(By.CSS_SELECTOR, ".signup-button").click()
```

OUTPUT:



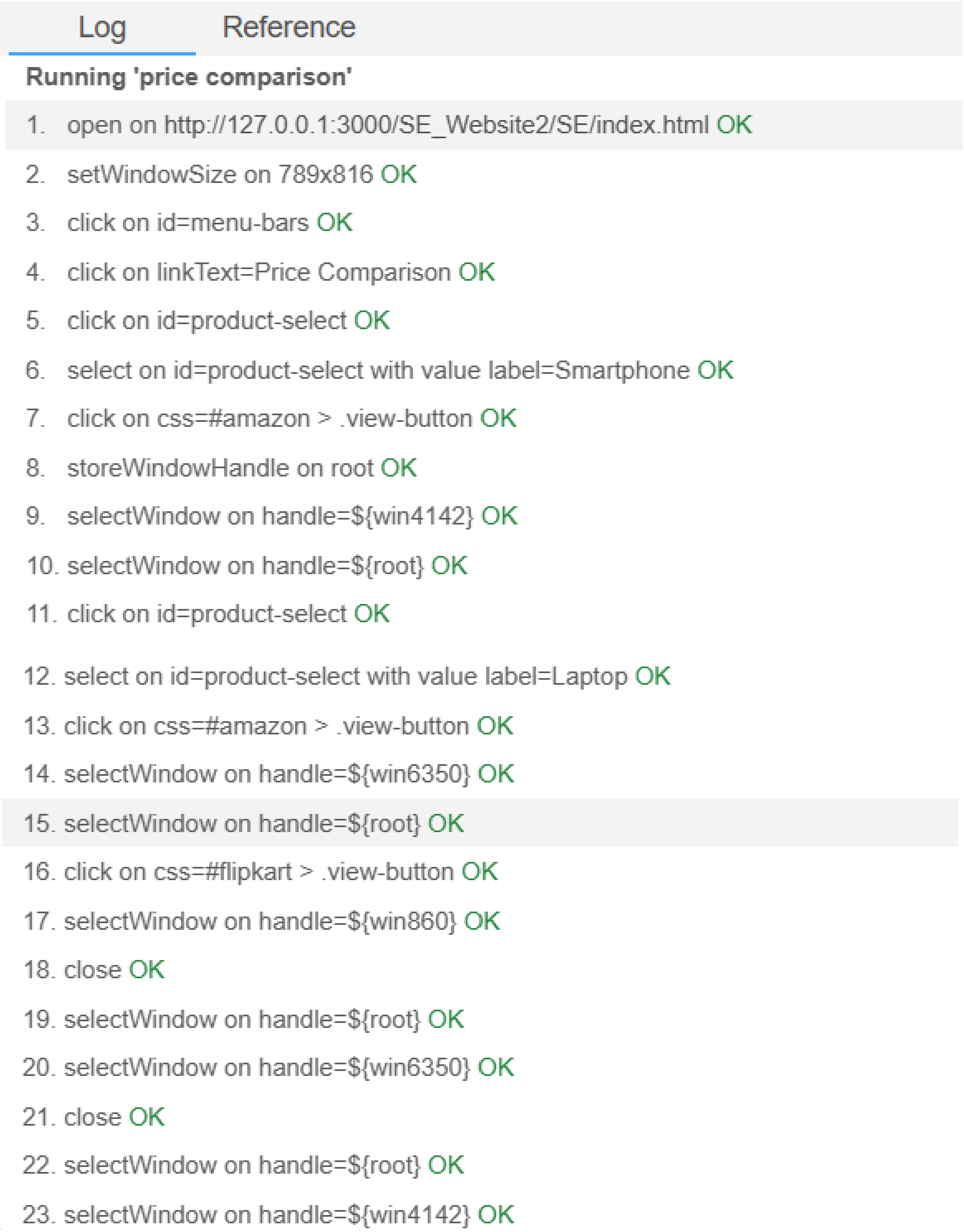
```
Running 'signUp'
1.  open on http://127.0.0.1:3000/SE_Website2/SE/index.html OK
2.  setWindowSize on 787x816 OK
3.  click on id=menu-bars OK
4.  click on css=.navbar > a:nth-child(1) OK
5.  click on id=username OK
6.  type on id=username with value shailly OK
7.  click on id=email OK
8.  type on id=email with value shaillyyadav979@gmail.com OK
9.  click on css=.signup-button OK
10. click on id=password OK
```



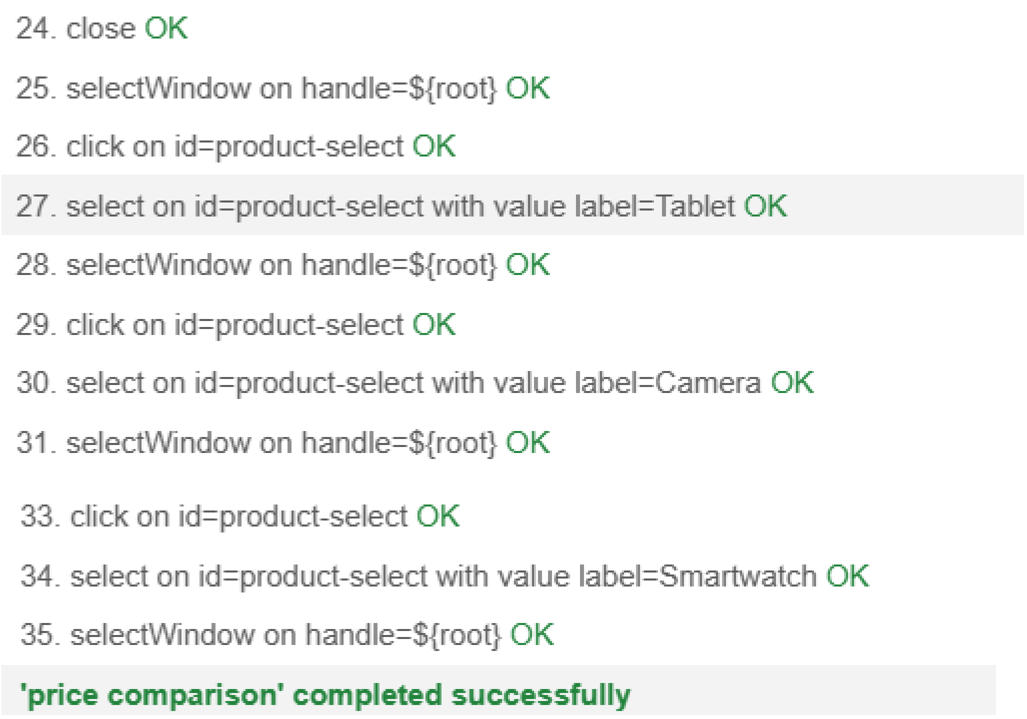
```
11. type on id=password with value 123 OK
12. click on css=.signup-button OK
'signUp' completed successfully
```

```
# 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 TestPricecomparison():
  def setup_method(self, method):
    self.driver = webdriver.Remote(command_executor='http://localhost:4444/wd/hub', desired_capabilities=DesiredCapabilities.CHROME)
    self.vars = {}

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

  def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
      return set(wh_now).difference(set(wh_then)).pop()

  def test_pricecomparison(self):
    self.driver.get("http://127.0.0.1:3000/SE_Website2/SE/index.html")
    self.driver.set_window_size(789, 816)
    self.driver.find_element(By.ID, "menu-bars").click()
    self.driver.find_element(By.LINK_TEXT, "Price Comparison").click()
    self.driver.find_element(By.ID, "product-select").click()
    dropdown = self.driver.find_element(By.ID, "product-select")
    dropdown.find_element(By.XPATH, "//option[. = 'Smartphone']").click()
    self.vars["window_handles"] = self.driver.window_handles
    self.driver.find_element(By.CSS_SELECTOR, "#amazon > .view-button").click()
    self.vars["win4142"] = self.wait_for_window(2000)
    self.vars["root"] = self.driver.current_window_handle
    self.driver.switch_to.window(self.vars["win4142"])
    self.driver.switch_to.window(self.vars["root"])
    self.driver.find_element(By.ID, "product-select").click()
    dropdown = self.driver.find_element(By.ID, "product-select")
    dropdown.find_element(By.XPATH, "//option[. = 'Laptop']").click()
    self.vars["window_handles"] = self.driver.window_handles
    self.driver.find_element(By.CSS_SELECTOR, "#amazon > .view-button").click()
    self.vars["win6350"] = self.wait_for_window(2000)
self.driver.switch_to.window(self.vars["win6350"])
self.driver.switch_to.window(self.vars["root"])
self.vars["window_handles"] = self.driver.window_handles
self.driver.find_element(By.CSS_SELECTOR, "#flipkart > .view-button").click()
self.vars["win860"] = self.wait_for_window(2000)
self.driver.switch_to.window(self.vars["win860"])
self.driver.close()
self.driver.switch_to.window(self.vars["root"])
self.driver.switch_to.window(self.vars["win6350"])
self.driver.close()
self.driver.switch_to.window(self.vars["root"])
self.driver.switch_to.window(self.vars["win4142"])
self.driver.close()
self.driver.switch_to.window(self.vars["root"])
self.driver.find_element(By.ID, "product-select").click()
dropdown = self.driver.find_element(By.ID, "product-select")
dropdown.find_element(By.XPATH, "//option[. = 'Tablet']").click()
self.driver.switch_to.window(self.vars["root"])
self.driver.find_element(By.ID, "product-select").click()
dropdown = self.driver.find_element(By.ID, "product-select")
dropdown.find_element(By.XPATH, "//option[. = 'Camera']").click()
self.driver.switch_to.window(self.vars["root"])
self.driver.find_element(By.ID, "product-section").click()
self.driver.find_element(By.ID, "product-select").click()
dropdown = self.driver.find_element(By.ID, "product-select")
dropdown.find_element(By.XPATH, "//option[. = 'Smartwatch']").click()
self.driver.switch_to.window(self.vars["root"])
```

OUTPUT:



```
Running 'price comparison'
1.  open on http://127.0.0.1:3000/SE_Website2/SE/index.html OK
2.  setWindowSize on 789x816 OK
3.  click on id=menu-bars OK
4.  click on linkText=Price Comparison OK
5.  click on id=product-select OK
6.  select on id=product-select with value label=Smartphone OK
7.  click on css=#amazon > .view-button OK
8.  storeWindowHandle on root OK
9.  selectWindow on handle=${win4142} OK
10. selectWindow on handle=${root} OK
11. click on id=product-select OK
12. select on id=product-select with value label=Laptop OK
13. click on css=#amazon > .view-button OK
14. selectWindow on handle=${win6350} OK
15. selectWindow on handle=${root} OK
16. click on css=#flipkart > .view-button OK
17. selectWindow on handle=${win860} OK
18. close OK
19. selectWindow on handle=${root} OK
20. selectWindow on handle=${win6350} OK
21. close OK
22. selectWindow on handle=${root} OK
23. selectWindow on handle=${win4142} OK
```



24. close OK

25. selectWindow on handle=${root} OK

26. click on id=product-select OK

27. select on id=product-select with value label=Tablet OK

28. selectWindow on handle=${root} OK

29. click on id=product-select OK

30. select on id=product-select with value label=Camera OK

31. selectWindow on handle=${root} OK

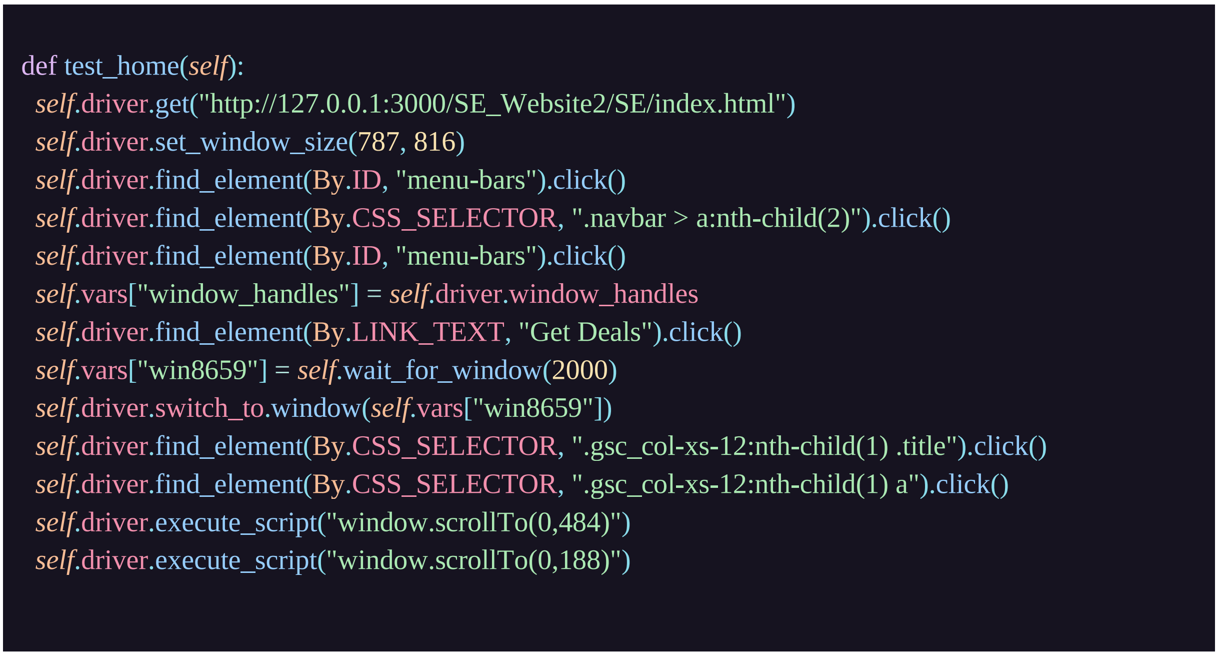
33. click on id=product-select OK

34. select on id=product-select with value label=Smartwatch OK

35. selectWindow on handle=${root} OK

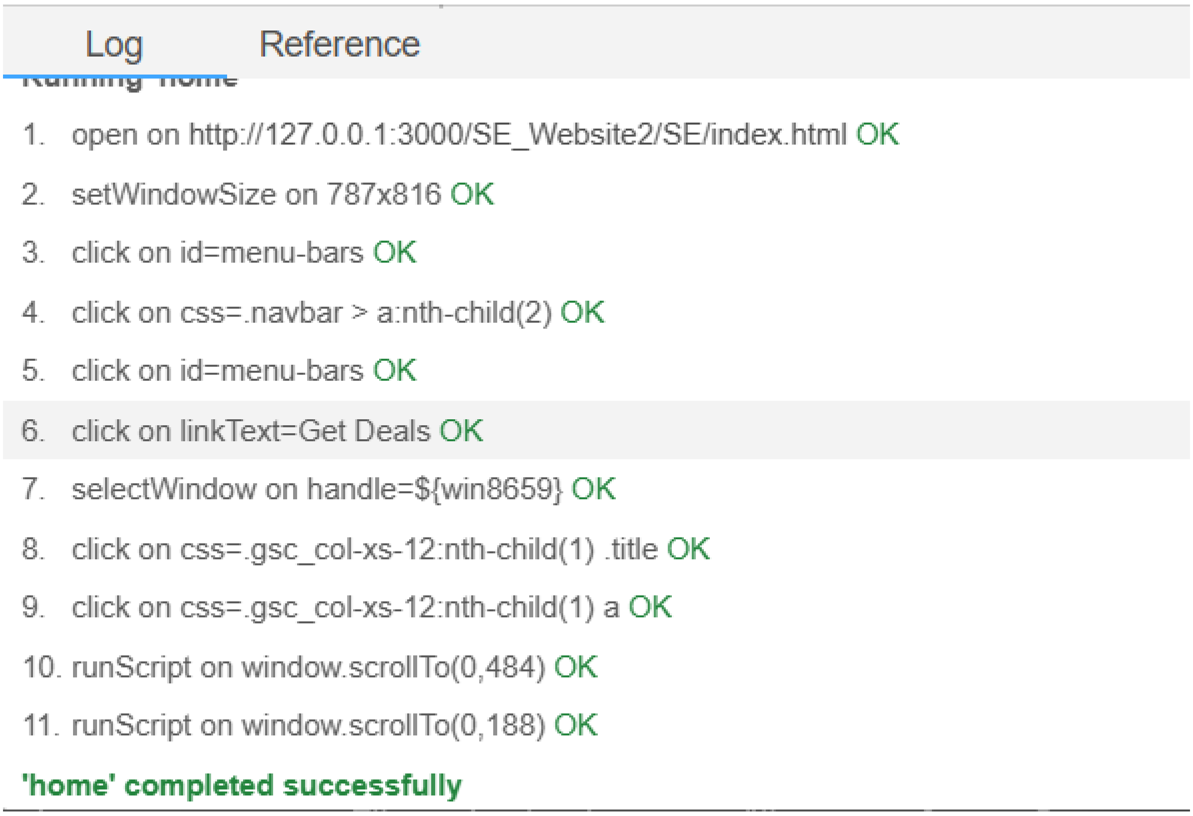
'price comparison' completed successfully

```
# 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 TestHome():
  def setup_method(self, method):
    self.driver = webdriver.Remote(command_executor='http://localhost:4444/wd/hub', desired_capabilities=DesiredCapabilities.CHROME)
    self.vars = {}

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

  def wait_for_window(self, timeout = 2):
    time.sleep(round(timeout / 1000))
    wh_now = self.driver.window_handles
    wh_then = self.vars["window_handles"]
    if len(wh_now) > len(wh_then):
      return set(wh_now).difference(set(wh_then)).pop()
```

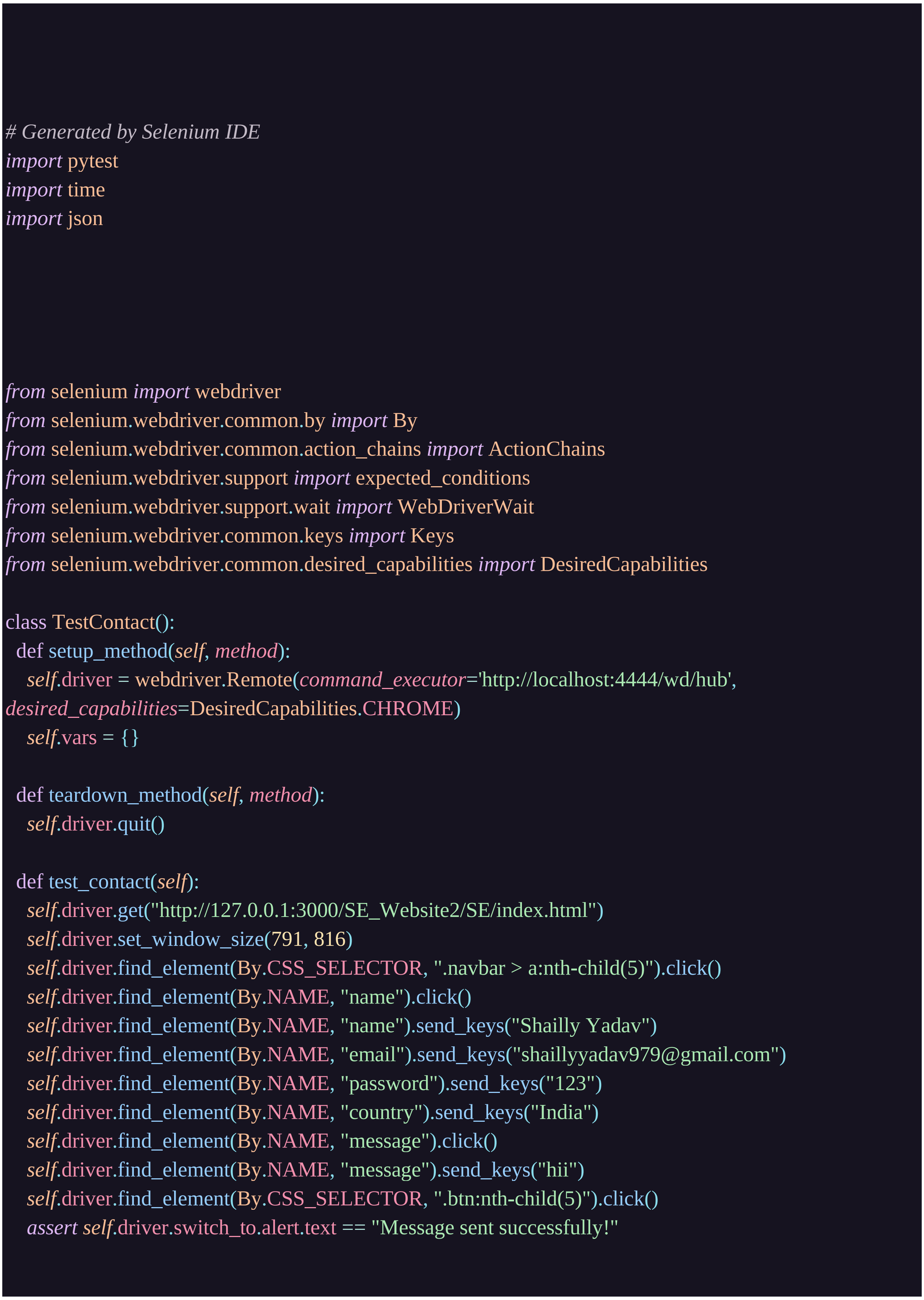


```
def test_home(self):
  self.driver.get("http://127.0.0.1:3000/SE_Website2/SE/index.html")
  self.driver.set_window_size(787, 816)
  self.driver.find_element(By.ID, "menu-bars").click()
  self.driver.find_element(By.CSS_SELECTOR, ".navbar > a:nth-child(2)").click()
  self.driver.find_element(By.ID, "menu-bars").click()
  self.vars["window_handles"] = self.driver.window_handles
  self.driver.find_element(By.LINK_TEXT, "Get Deals").click()
  self.vars["win8659"] = self.wait_for_window(2000)
  self.driver.switch_to.window(self.vars["win8659"])
  self.driver.find_element(By.CSS_SELECTOR, ".gsc_col-xs-12:nth-child(1) .title").click()
  self.driver.find_element(By.CSS_SELECTOR, ".gsc_col-xs-12:nth-child(1) a").click()
  self.driver.execute_script("window.scrollTo(0,484)")
  self.driver.execute_script("window.scrollTo(0,188)")
```

OUTPUT:

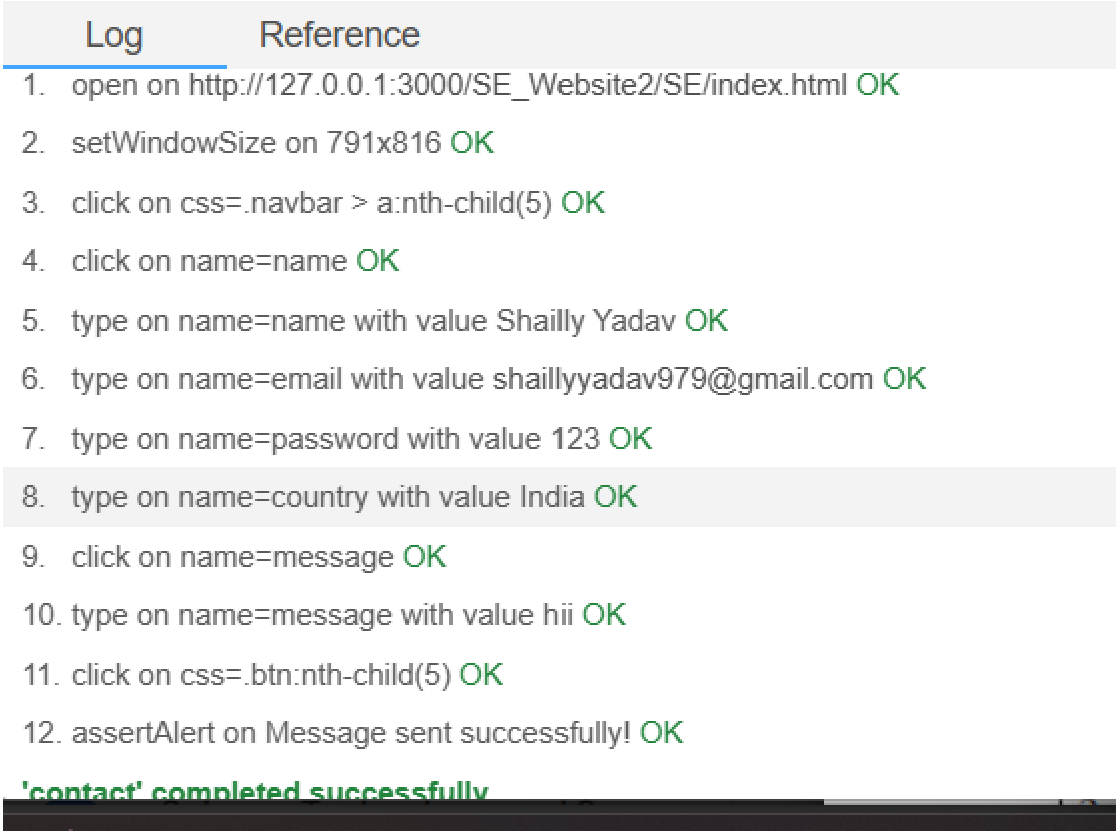


```
Running home
1. open on http://127.0.0.1:3000/SE_Website2/SE/index.html OK
2. setWindowSize on 787x816 OK
3. click on id=menu-bars OK
4. click on css=.navbar > a:nth-child(2) OK
5. click on id=menu-bars OK
6. click on linkText=Get Deals OK
7. selectWindow on handle=${win8659} OK
8. click on css=.gsc_col-xs-12:nth-child(1) .title OK
9. click on css=.gsc_col-xs-12:nth-child(1) a OK
10. runScript on window.scrollTo(0,484) OK
11. runScript on window.scrollTo(0,188) OK
'home' completed successfully
```



```
# 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 TestContact():
  def setup_method(self, method):
    self.driver = webdriver.Remote(command_executor='http://localhost:4444/wd/hub',
desired_capabilities=DesiredCapabilities.CHROME)
    self.vars = {}

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

  def test_contact(self):
    self.driver.get("http://127.0.0.1:3000/SE_Website2/SE/index.html")
    self.driver.set_window_size(791, 816)
    self.driver.find_element(By.CSS_SELECTOR, ".navbar > a:nth-child(5)").click()
    self.driver.find_element(By.NAME, "name").click()
    self.driver.find_element(By.NAME, "name").send_keys("Shailly Yadav")
    self.driver.find_element(By.NAME, "email").send_keys("shaillyyadav979@gmail.com")
    self.driver.find_element(By.NAME, "password").send_keys("123")
    self.driver.find_element(By.NAME, "country").send_keys("India")
    self.driver.find_element(By.NAME, "message").click()
    self.driver.find_element(By.NAME, "message").send_keys("hii")
    self.driver.find_element(By.CSS_SELECTOR, ".btn:nth-child(5)").click()
    assert self.driver.switch_to.alert.text == "Message sent successfully!"
```

OUTPUT:



```
1.  open on http://127.0.0.1:3000/SE_Website2/SE/index.html OK
2.  setWindowSize on 791x816 OK
3.  click on css=.navbar > a:nth-child(5) OK
4.  click on name=name OK
5.  type on name=name with value Shailly Yadav OK
6.  type on name=email with value shaillyyadav979@gmail.com OK
7.  type on name=password with value 123 OK
8.  type on name=country with value India OK
9.  click on name=message OK
10. type on name=message with value hii OK
11. click on css=.btn:nth-child(5) OK
12. assertAlert on Message sent successfully! OK
'contact' completed successfully
```