```
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
import time
import pytest


@pytest.fixture()
def setup():
    driver = webdriver.Chrome()
    driver.get('https://svburger1.co.il/#/HomePage')
    driver.maximize_window()
    driver.implicitly_wait(10)
    yield driver
    driver.close()


def test_sanity(setup):
    # step 1 - Open SVBurger on Chrome
    driver = setup
    # step 2 - Click the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Enter the email in the email field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys(
        'svburger_web@gmail.com')
    # Step 4 - Enter the password in the password field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys('123456A')
    # Step 5 - Click on the "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # Step 6 - Click on the "Combo Meal"
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/
div/div/div[2]/div').click()
    # Step 7 - Click on the "Reserve" button after it changes to clickable
    # Waiting until the button changes to clickable
    reserve_btn = WebDriverWait(driver, 10).until(
        EC.element_to_be_clickable((By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/
div/button[2]')))
    driver.execute_script("arguments[0].click();", reserve_btn)
    # Step 8 - Click on the "Send" button
    send_btn = WebDriverWait(driver, 10).until(EC.visibility_of_element_located(
        (By.XPATH, '/html/body/div/div[2]/div[1]/div/div/div[2]/div/div/div/div[2]/
div[4]/div[1]/button')))
    driver.execute_script("arguments[0].click();", send_btn)
    # Test if the SVBurger Summary modal opens
    time.sleep(20)
    assert driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/
div[2]/div').is_displayed()


"""
Suite 1 - Signing Up
We are performing functionality tests, in this script we are testing the cases:
    - 1.1 > Sign Up with Hotmail mail
    - 1.2 > Sign Up with Outlook mail
    - 1.3 > Sign Up with Walla mail
    - 1.4 > Sign up with 8 characters in the password
    - 1.5 > Sign up with 9 characters in the password
"""


users_list1 = [['svburger_web100@hotmail.com', '123456A'],
['svburger_web100@outlook.com', '123456A'], ['svburger_web100@walla.com',
'123456A'], ['svburger_web100@gmail.com', '1234567A'],
['svburger_web200@gmail.com', '12345678A']]


@pytest.mark.parametrize("list_of_users", users_list1)
def test_signing_up(setup, list_of_users):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # Step 2 - Click on the "Sign Up" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[2]/
button').click()
    # Step 3 - Fill the first name textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/
input[1]').send_keys('aaaaaa')
    # Step 4 - Fill the last name textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/
input[2]').send_keys('aaaaaa')
    # Step 5 - Fill the email textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/
input[3]').send_keys(list_of_users[0])
    # Step 6 -Fill the new password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/
input[4]').send_keys(list_of_users[1])
    # Step 7 - Fill the confirmation password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/
input[5]').send_keys(list_of_users[1])
    # Step 8 - Click on the "Sign Up" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/button').click()
    # If we successfully sign up we can know if the "Log Out" button below is displayed
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/button[1]').is_displayed()


"""
Suite 1 - Signing Up
We are performing error handling tests, in this script we are testing the cases:
    - 1.6 > Sign Up without filling the first name
    - 1.7 > Sign Up without filling the last name
"""
users_list2 = [['', 'aaaaaa', 'svburger_web300@gmail.com', 'First name must be in English letters only'], ['aaaaaa', '', 'svburger_web400@gmail.com', 'Last name must be in English letters only']]


@pytest.mark.parametrize("list_of_users", users_list2)
def test_signing_up_with_missing_fields(setup, list_of_users):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # Step 2 - Click on the "Sign Up" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[2]/button').click()
    # Step 3 - Fill the first name textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/input[1]').send_keys(list_of_users[0])
    # Step 4 - Fill the last name textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/input[2]').send_keys(list_of_users[1])
    # Step 5 - Fill the email textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/input[3]').send_keys(list_of_users[2])
    # Step 6 -Fill the new password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/input[4]').send_keys('123456A')
    # Step 7 - Fill the confirmation password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/input[5]').send_keys('123456A')
    # Step 8 - Click on the "Sign Up" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/form/button').click()
    # We should get an error modal, this error modal should have an error
```

```
message which we need to check if it equals the error message that we should
get for each error
    time.sleep(10)
    my_alert = driver.switch_to.alert
    alert_message = my_alert.text
    my_alert.dismiss()
    assert alert_message == list_of_users[3]


"""
Suite 2 - Signing In
We are performing functionality tests, in this script we are testing the cases:
    - 2.1 > Sign In with Hotmail mail
    - 2.2 > Sign In with Outlook mail
    - 2.3 > Sign In with Walla mail
    - 2.4 > Sign In with 8 characters in the password
    - 2.5 > Sign In with 9 characters in the password
"""
users_list3 = [['svburger_web100@hotmail.com', '123456A'],
['svburger_web100@outlook.com', '123456A'], ['svburger_web100@walla.com',
'123456A'], ['svburger_web100@gmail.com', '1234567A'],
['svburger_web200@gmail.com', '12345678A']]


@pytest.mark.parametrize("list_of_users", users_list3)
def test_signing_in(setup, list_of_users):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # Step 2 - Click on the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Fill the email textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys(list_of_users[0])
    # Step 4 - Fill the password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys(list_of_users[1])
    # Step 5 - Click on "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # If we successfully signed in we can know if the "Log Out" button below is
displayed
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/
button[1]').is_displayed()


"""
```

```
Suite 2 - Signing In
We are performing error handling tests, in this script we are testing the cases:
    - 2.6 > Sign Up without filling the password
    - 2.7 > Sign Up with a wrong password
"""
users_list4 = [['svburger_web@gmail.com ', ''], ['svburger_web@gmail.com ',
'123456788A']]


@pytest.mark.parametrize("list_of_users", users_list4)
def test_signing_in_with_errors(setup, list_of_users):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # Step 2 - Click on the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Fill the email textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys(list_of_users[0])
    # Step 4 - Fill the password textbox
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys(list_of_users[1])
    # Step 5 - Click on "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # We should get an error modal, this error modal should have an error
message which we need to check if it equals the error message that we should
get for each error
    time.sleep(20)
    my_alert = driver.switch_to.alert
    error_message = my_alert.text
    my_alert.dismiss()
    assert error_message == 'Failed to log in'



"""
Suite 3 - ordering
We are performing functionality tests, in this script we are testing the cases:
    - 3.1 > The price of ordering the 2 meals “Combo Meal” & “Kids Meal”
    - 3.2 > The price of ordering “Combo Meal” & “Burger”
    - 3.3 > The price of ordering “Combo Meal” & “Vegan”
    - 3.4 > The price of ordering “Combo Meal” & “Sides”
"""

products_list = [['//*[@id="root"]/div[2]/div[1]/div/div/div/div[2]/div/div',
'107.8$'], ['//*[@id="root"]/div[2]/div[1]/div/div/div/div[3]/div/div', '114.4$'], ['//
*[@id="root"]/div[2]/div[1]/div/div/div/div[4]/div/div', '114.4$'], ['//*[@id="root"]/
div[2]/div[1]/div/div/div/div[5]/div/div', '78.1$']]


@pytest.mark.parametrize("list_of_products", products_list)
def test_ordering_two_meals(setup, list_of_products):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # step 2 - Click the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Enter the email in the email field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys('svburger_web@gmail.com')
    # Step 4 - Enter the password in the password field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys('123456A')
    # Step 5 - Click on the "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # Step 6 - Click on the "Combo Meal"
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/
div/div/div[2]/div').click()
    # Step 7 - Click on the second meal
    driver.find_element(By.XPATH, list_of_products[0]).click()
    # Step 8 - Click on the "Reserve" button after it changes to clickable
    # Waiting until the button changes to clickable
    reserve_btn = WebDriverWait(driver,
10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="root"]/div[2]/div[1]/
div/div/button[2]')))
    driver.execute_script("arguments[0].click();", reserve_btn)
    # Step 9 - Click on the "Send" button
    send_btn = WebDriverWait(driver,
10).until(EC.visibility_of_element_located((By.XPATH, '/html/body/div/div[2]/
div[1]/div/div/div[2]/div/div/div/div[2]/div[5]/div[1]/button')))
    driver.execute_script("arguments[0].click();", send_btn)
    # Test if the SVBurger Summary modal opens and shows the total price for
the order
    time.sleep(5)
    total_price_string = WebDriverWait(driver,
10).until(EC.visibility_of_element_located((By.XPATH, '//*[@id="root"]/div[2]/
div[1]/div/div/div[2]/div/div/div/div[2]/h2[1]')))
    total_price = total_price_string.text.split(' ')
    print(total_price[1])
    assert total_price[1] == list_of_products[1]
```

```
"""
Suite 3 - ordering
We are performing functionality test, in this script we are testing the case:
    - 3.5 > Unselecting a selected meal
"""


def test_unselecting_selected_meal(setup):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # step 2 - Click the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/button').click()
    # Step 3 - Enter the email in the email field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/input[1]').send_keys('svburger_web@gmail.com')
    # Step 4 - Enter the password in the password field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/input[2]').send_keys('123456A')
    # Step 5 - Click on the "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/button').click()
    # Step 6 - Click on the "Combo Meal" to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/div/div/div[2]/div').click()
    # Step 7 - Click on the "Combo Meal" to unselect it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/div/div/div[2]/div').click()
    # We need to test if the meal card has the white background in the style attribute to make sure it was unselected
    meal_card_style_attribute = driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/div').get_attribute("style")
    assert meal_card_style_attribute.__contains__('background-color: white')


"""
Suite 3 - ordering
We are performing error handling tests, in this script we are testing the case:
    - 3.6 > Selecting 4 meal cards
"""


def test_selecting_four_meals(setup):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # step 2 - Click the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Enter the email in the email field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys('svburger_web@gmail.com')
    # Step 4 - Enter the password in the password field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys('123456A')
    # Step 5 - Click on the "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # Step 6 - Click on the "Combo Meal" to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/
div/div/div[2]/div').click()
    # Step 7 - Click on the "Kids Meal" to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[2]/
div/div').click()
    # Step 8 - Click on the "Burger" meal to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/
div[3]/div/div').click()
    # Step 9 - Click on the "Vegan" meal to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/
div[4]/div/div').click()
    time.sleep(5)
    # We need to test if the "Vegan" meal card has the white background in the
style attribute to make sure it was unselected
    meal_card_style_attribute = driver.find_element(By.XPATH, '/html/body/div/
div[2]/div[1]/div/div/div/div[4]/div').get_attribute("style")
    print(meal_card_style_attribute)
    assert meal_card_style_attribute.__contains__('background-color: white')


"""
Suite 3 - ordering
We are performing error handling tests, in this script we are testing the case:
    - 3.7 > Selecting more than 2 meals in the meal quantity
"""


def test_selecting_more_than_two_in_the_meal_quantity(setup):
    # Step 1 - Open SVBurger on Chrome
    driver = setup
    # step 2 - Click the "Sign In" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/a[1]/
button').click()
    # Step 3 - Enter the email in the email field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[1]').send_keys('svburger_web@gmail.com')
```

```
    # Step 4 - Enter the password in the password field
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
input[2]').send_keys('123456A')
    # Step 5 - Click on the "Sign in" button
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/form/div/
button').click()
    # Step 6 - Click on the "Combo Meal" to select it
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div/div[1]/
div/div/div[2]/div').click()
    # Step 7 - Click on the "Reserve" button after it changes to clickable
    # Waiting until the button changes to clickable
    reserve_btn = WebDriverWait(driver,
10).until(EC.element_to_be_clickable((By.XPATH, '//*[@id="root"]/div[2]/div[1]/
div/div/button[2]')))
    driver.execute_script("arguments[0].click();", reserve_btn)
    # Step 8 - Fill the value 3 in the meal quantity
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div[2]/div/
div/div/div[2]/div[2]/div[1]/div/input').clear()
    driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/div/div[2]/div/
div/div/div[2]/div[2]/div[1]/div/input').send_keys('3')
    # Testing if the order summary modal appears after clicking the "Send"
button, the modal should not appear if we select more than 2 meal quantity
    send_btn = WebDriverWait(driver,
10).until(EC.visibility_of_element_located((By.XPATH, '/html/body/div/div[2]/
div[1]/div/div/div[2]/div/div/div/div[2]/div[4]/div[1]/button')))
    driver.execute_script("arguments[0].click();", send_btn)
    time.sleep(5)
    assert not driver.find_element(By.XPATH, '//*[@id="root"]/div[2]/div[1]/div/
div/div[2]/div/div/div/div[2]').is_displayed()
```