Poniżej znajdziesz bibliotekę w Pythonie, która implementuje wzorzec projektowy **Page Object Pattern** dla podstrony **Elements > Text Box** i **Check Box** ze strony [demoqa.com](https://demoqa.com/).

**Struktura projektu**

bash

Skopiuj kod

/demoqa\_page\_objects

\_\_init\_\_.py

base\_page.py

elements\_page.py

text\_box\_page.py

check\_box\_page.py

/tests

test\_text\_box.py

test\_check\_box.py

requirements.txt

**1. Plik base\_page.py**

Klasa bazowa dostarczająca metody do obsługi elementów strony.

python

Skopiuj kod

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

class BasePage:

def \_\_init\_\_(self, driver):

self.driver = driver

self.wait = WebDriverWait(driver, 10)

def open(self, url):

"""Otwiera stronę o podanym URL."""

self.driver.get(url)

def find\_element(self, locator):

"""Znajduje pojedynczy element."""

return self.wait.until(EC.visibility\_of\_element\_located(locator))

def click\_element(self, locator):

"""Kliknięcie w element."""

element = self.find\_element(locator)

element.click()

def enter\_text(self, locator, text):

"""Wpisuje tekst do pola."""

element = self.find\_element(locator)

element.clear()

element.send\_keys(text)

**2. Plik elements\_page.py**

Klasa reprezentująca sekcję **Elements**.

python

Skopiuj kod

from demoqa\_page\_objects.base\_page import BasePage

from selenium.webdriver.common.by import By

class ElementsPage(BasePage):

TEXT\_BOX\_MENU = (By.ID, "item-0")

CHECK\_BOX\_MENU = (By.ID, "item-1")

def open\_text\_box(self):

"""Przejście do sekcji Text Box."""

self.click\_element(self.TEXT\_BOX\_MENU)

def open\_check\_box(self):

"""Przejście do sekcji Check Box."""

self.click\_element(self.CHECK\_BOX\_MENU)

**3. Plik text\_box\_page.py**

Klasa obsługująca sekcję **Text Box**.

python

Skopiuj kod

from demoqa\_page\_objects.base\_page import BasePage

from selenium.webdriver.common.by import By

class TextBoxPage(BasePage):

FULL\_NAME\_FIELD = (By.ID, "userName")

EMAIL\_FIELD = (By.ID, "userEmail")

CURRENT\_ADDRESS\_FIELD = (By.ID, "currentAddress")

PERMANENT\_ADDRESS\_FIELD = (By.ID, "permanentAddress")

SUBMIT\_BUTTON = (By.ID, "submit")

OUTPUT\_SECTION = (By.ID, "output")

def fill\_form(self, name, email, current\_address, permanent\_address):

"""Wypełnia formularz Text Box."""

self.enter\_text(self.FULL\_NAME\_FIELD, name)

self.enter\_text(self.EMAIL\_FIELD, email)

self.enter\_text(self.CURRENT\_ADDRESS\_FIELD, current\_address)

self.enter\_text(self.PERMANENT\_ADDRESS\_FIELD, permanent\_address)

self.click\_element(self.SUBMIT\_BUTTON)

def get\_output\_text(self):

"""Zwraca zawartość sekcji wyników."""

return self.find\_element(self.OUTPUT\_SECTION).text

**4. Plik check\_box\_page.py**

Klasa obsługująca sekcję **Check Box**.

python

Skopiuj kod

from demoqa\_page\_objects.base\_page import BasePage

from selenium.webdriver.common.by import By

class CheckBoxPage(BasePage):

TOGGLE\_HOME = (By.CSS\_SELECTOR, ".rct-icon.rct-icon-expand-close")

CHECK\_DOCUMENTS = (By.XPATH, "//label[@for='tree-node-documents']//span[@class='rct-checkbox']")

OUTPUT\_SECTION = (By.ID, "result")

def expand\_home\_directory(self):

"""Rozwija katalog główny."""

self.click\_element(self.TOGGLE\_HOME)

def select\_documents(self):

"""Zaznacza opcję Documents."""

self.click\_element(self.CHECK\_DOCUMENTS)

def get\_selected\_items(self):

"""Zwraca wybrane elementy."""

return self.find\_element(self.OUTPUT\_SECTION).text

**5. Testy**

**Testy dla Text Box: test\_text\_box.py**

python

Skopiuj kod

import unittest

from selenium import webdriver

from demoqa\_page\_objects.elements\_page import ElementsPage

from demoqa\_page\_objects.text\_box\_page import TextBoxPage

class TextBoxTests(unittest.TestCase):

@classmethod

def setUpClass(cls):

cls.driver = webdriver.Chrome()

cls.driver.maximize\_window()

cls.driver.get("https://demoqa.com/elements")

@classmethod

def tearDownClass(cls):

cls.driver.quit()

def test\_text\_box\_submission(self):

elements\_page = ElementsPage(self.driver)

text\_box\_page = TextBoxPage(self.driver)

elements\_page.open\_text\_box()

text\_box\_page.fill\_form(

name="John Doe",

email="johndoe@example.com",

current\_address="123 Main St",

permanent\_address="456 Another St"

)

output = text\_box\_page.get\_output\_text()

self.assertIn("John Doe", output)

self.assertIn("johndoe@example.com", output)

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**Testy dla Check Box: test\_check\_box.py**

python

Skopiuj kod

import unittest

from selenium import webdriver

from demoqa\_page\_objects.elements\_page import ElementsPage

from demoqa\_page\_objects.check\_box\_page import CheckBoxPage

class CheckBoxTests(unittest.TestCase):

@classmethod

def setUpClass(cls):

cls.driver = webdriver.Chrome()

cls.driver.maximize\_window()

cls.driver.get("https://demoqa.com/elements")

@classmethod

def tearDownClass(cls):

cls.driver.quit()

def test\_check\_box\_selection(self):

elements\_page = ElementsPage(self.driver)

check\_box\_page = CheckBoxPage(self.driver)

elements\_page.open\_check\_box()

check\_box\_page.expand\_home\_directory()

check\_box\_page.select\_documents()

output = check\_box\_page.get\_selected\_items()

self.assertIn("documents", output.lower())

if \_\_name\_\_ == "\_\_main\_\_":

unittest.main()

**6. Plik requirements.txt**

Zależności:

makefile

Skopiuj kod

selenium==4.10.0

**Uruchamianie testów**

1. Zainstaluj Selenium:

bash

Skopiuj kod

pip install -r requirements.txt

1. Uruchom testy:

bash

Skopiuj kod

python -m unittest tests/test\_text\_box.py

python -m unittest tests/test\_check\_box.py

Ten projekt umożliwia łatwe zarządzanie testami dla podstron **Text Box** i **Check Box**, a dzięki **Page Object Pattern** kod jest modularny i łatwy w utrzymaniu.