```
import time


from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected_conditions as EC

from selenium.webdriver.support.ui import WebDriverWait




class BasePage:

    def __init__(self, driver, wait):

        self.driver = driver

        self.wait = wait


    def click_button(self, by_locator):

        button = WebDriverWait(self.driver, 15).until(EC.element_to_be_clickable(by_locator))

        button.click()


    def click_element(self, by_locator):

        element = WebDriverWait(self.driver, 15).until(EC.element_to_be_clickable(by_locator))

        element.click()


    def fill_box(self, by_locator, text):

        box = WebDriverWait(self.driver, 15).until(EC.element_to_be_clickable(by_locator))

        box.clear()

        box.send_keys(text)


    def get_text(self, by_locator):

        message = WebDriverWait(self.driver, 15).until(EC.visibility_of_element_located(by_locator))
```

```
    return message.text


def get_value(self, by_locator):
    message = WebDriverWait(self.driver, 15).until(EC.visibility_of_element_located(by_locator))
    return message.get_attribute("value")


def scroll_up(self):
    # Scroll to top of the page
    self.driver.execute_script("window.scrollTo(0, 300)")
    time.sleep(10)


def get_cart_content(self):
    self.wait.until(
        EC.visibility_of_element_located((By.XPATH, "//span[contains(text(),'Shopping cart')]"))).click()


    tr_elements = self.driver.find_elements(By.XPATH, "//table[contains(@class, "
                                            "'cart')]/thead/following-sibling::tbody/tr")
    cart_contents = []


    for tr_element in tr_elements:
        sku = tr_element.find_element(By.XPATH, ".//td[contains(@class, 'sku')]").text
        image = tr_element.find_element(
            By.XPATH, ".//td[contains(@class, 'product-picture')]/a/img"
        ).get_attribute("src")
        product = tr_element.find_element(
            By.XPATH, ".//td[contains(@class, 'product')]/a[@class='product-name']"
        ).text
```

```
        price = tr_element.find_element(
            By.XPATH,
            ".//td[contains(@class, 'unit-price')]/span[@class='product-unit-price']",
        ).text
        quantity = tr_element.find_element(
            By.XPATH, ".//td[contains(@class, 'quantity')]/input"
        ).get_attribute("value")
        total = tr_element.find_element(
            By.XPATH,
            ".//td[contains(@class, 'subtotal')]/span[@class='product-subtotal']",
        ).text

        product_data = {
            "sku": sku,
            "image": image,
            "product": product,
            "price": price,
            "quantity": quantity,
            "total": total,
        }

        cart_contents.append(product_data)

    return cart_contents
    # print(products[0]['product'])

def clean_cart(self):
```

```
button_locator = (By.XPATH,
                  "//td[@class='remove-from-cart']//button[contains(@class, 'remove-btn') and contains("
                  "@onclick, 'removefromcart')]")
wait_time = 5



button_elements = WebDriverWait(self.driver, wait_time).until(
    EC.presence_of_all_elements_located(button_locator))



for element in button_elements:
    WebDriverWait(self.driver, wait_time).until(EC.element_to_be_clickable(button_locator)).click()
    time.sleep(2)
```