```python
import os
import logging
from contextlib import contextmanager
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,
    StaleElementReferenceException,
    WebDriverException
)
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
import pytest

logging.basicConfig(
    level=logging.DEBUG,
    format='%(asctime)s - %(levelname)s - %(message)s',
    handlers=[
        logging.FileHandler('test_execution.log'),
        logging.StreamHandler()
    ]
)

URL = os.getenv('TEST_URL', "https://www.qaonlineacademy.com/")
try:
    TIMEOUT = int(os.getenv('TIMEOUT', "20"))
    if TIMEOUT <= 0:
        raise ValueError("TIMEOUT must be positive")
except ValueError:
    TIMEOUT = 20
    logging.warning("Invalid TIMEOUT value, using default: 20")

CHROMEDRIVER_PATH = os.getenv('CHROMEDRIVER_PATH', 'chromedriver.exe')

class HomePage:
    def __init__(self, driver):
        self.driver = driver
        self.wait = WebDriverWait(driver, TIMEOUT)

    def wait_for_page_load(self):
        self.wait.until(
            lambda driver: driver.execute_script("return document.readyState") == "complete"
        )

    def print_page_info(self):
        try:
            logging.debug(f"Current URL: {self.driver.current_url}")
            logging.debug(f"Page title: {self.driver.title}")
            buttons = self.driver.find_elements(By.TAG_NAME, "button")
            links = self.driver.find_elements(By.TAG_NAME, "a")
            logging.debug("Buttons found on page:")
            for button in buttons:
                try:
                    logging.debug(f"Button text: '{button.text}', Visible: {button.is_displayed()}")
                except (StaleElementReferenceException, WebDriverException) as e:
                    logging.debug(f"Could not get button information: {e}")
            logging.debug("Links found on page:")
            for link in links:
                try:
                    logging.debug(f"Link text: '{link.text}', Visible: {link.is_displayed()}")
                except (StaleElementReferenceException, WebDriverException) as e:
                    logging.debug(f"Could not get link information: {e}")
        except Exception as e:
            logging.error(f"Error while getting page info: {e}")

    def click_get_started(self):
        self.wait_for_page_load()
        self.print_page_info()
        screenshot_path = "before_search.png"
        try:
            self.driver.save_screenshot(screenshot_path)
            logging.debug(f"Screenshot saved to {screenshot_path}")
        except Exception as e:
            logging.error(f"Failed to save screenshot: {e}")
        selectors = [
            (By.XPATH, "//span[text()='Get Started']/ancestor::*[self::a or self::button]"),
            (By.XPATH, "//button[contains(text(), 'Get Started')]"),
            (By.XPATH, "//a[contains(text(), 'Get Started')]"),
            (By.LINK_TEXT, "Get Started"),
            (By.PARTIAL_LINK_TEXT, "Get Started")
        ]
        last_exception = None
        for selector in selectors:
            try:
                logging.debug(f"Trying selector: {selector}")
                get_started_btn = self.wait.until(
                    EC.element_to_be_clickable(selector)
                )
                logging.info(f"Found button with selector {selector}")
                get_started_btn.click()
                logging.info("Clicked Get Started button")
                modal = self.wait.until(
                    EC.visibility_of_element_located((By.CLASS_NAME, "modal-content"))
                )
                return modal
            except (TimeoutException, NoSuchElementException, WebDriverException) as e:
                logging.debug(f"Selector {selector} failed: {str(e)}")
                last_exception = e
                continue
        raise TimeoutException(f"Could not find or click Get Started button with any selector. Last error: {last_exception}")
```