```
def test_sign_in_successful_flow(monkeypatch, mocker):
    """
    Test a successful sign-in with correct credentials, soft token MFA, and successful post-login sync.
    """
    # Arrange
    driver = mocker.Mock()
    driver.current_url = 'https://mint.intuit.com/'
    driver.implicitly_wait = mocker.Mock()
    driver.get = mocker.Mock()

    # Simulate all page transitions
    monkeypatch.setattr('signIn.home_page', lambda d: None)
    monkeypatch.setattr('signIn.user_selection_page', lambda d: None)
    monkeypatch.setattr('signIn.handle_same_page_username_password', lambda d, e, p: None)
    monkeypatch.setattr('signIn.handle_login_failures', lambda d: None)
    monkeypatch.setattr('signIn.bypass_verified_user_page', lambda d: True)
    monkeypatch.setattr('signIn.bypass_passwordless_login_page', lambda d: None)
    monkeypatch.setattr('signIn.account_selection_page', lambda d, a: None)
    monkeypatch.setattr('signIn.password_page', lambda d, p: None)
    monkeypatch.setattr('signIn.handle_wait_for_sync', lambda d, t, f: "Account refresh complete")

    result = sign_in(
        email='test@example.com',
        password='securepass',
        driver=driver,
        mfa_method='SOFT_TOKEN',
        mfa_token='FAKESECRET',
        mfa_input_callback=lambda prompt: '123456',
        wait_for_sync=True,
        wait_for_sync_timeout=300,
        fail_if_stale=False
    )
    # Assert
    assert result == "Account refresh complete"

---Extra Details---
Filename: signIn.py
Description: Test the happy-path flow of sign_in with all success: valid credentials, SOFT_TOKEN MFA, and a successful accoun
Score: 100%
Alignment: 100%
Validation Notes: Mocks critical path functions to simulate the ideal end-to-end login scenario. This checks all central integration p
```