

```

def test_mfa_page_invalid_mfa_method(monkeypatch, caplog):
    """
    Test that mfa_page returns gracefully and logs if the supplied MFA method is not available or optional.
    """
    import signIn
    class DummyInput:
        def clear(self): pass
        def send_keys(self, v): pass
        def click(self): pass
        @property
        def text(self): return ""
    class DummyDriver:
        def find_element(self, by, value):
            raise signIn.NoSuchElementException()
    def always_raise(*args, **kwargs):
        raise signIn.MFAMethodNotAvailableError('not available')
    monkeypatch.setattr('signIn.set_mfa_method', always_raise)
    d = DummyDriver()
    with caplog.at_level('INFO'):
        result = signIn.mfa_page(
            d, mfa_method='NOT_A_METHOD', mfa_token=None, mfa_input_callback=None,
            imap_account=None, imap_password=None, imap_server=None, imap_folder="INBOX"
        )
    assert result is None
    assert any("not available" in msg for msg in caplog.text or "does not require" in caplog.text)

```

---Extra Details---

Filename: signIn.py

Description: Test that mfa\_page logs and returns gracefully if the specified MFA method is not available, covering optional MFA a

Score: 90%

Alignment: 90%

Validation Notes: Ensures MFA error-path is robust and logs correctly without exception propagation.