```
def test_mfa_email_code_fetched_via_imap(monkeypatch):
    """
    Ensure handle_email_by_imap extracts an MFA code from a mocked IMAP email correctly.
    """
    from signIn import handle_email_by_imap
    mfa_token_input = type('FakeInput', (), {'clear': lambda self: None, 'send_keys': lambda self, code: setattr(self, 'sent_code', code
    mfa_token_button = type('FakeButton', (), {'click': lambda self: setattr(self, 'clicked', True)})()
    state = {}
    mfa_token_input.send_keys = lambda code: state.update({'code': code})
    mfa_token_button.click = lambda: state.update({'clicked': True})

    # Patch get_email_code to return a fixed code
    monkeypatch.setattr('signIn.get_email_code', lambda *a, **kw: '123456')
    handle_email_by_imap(
        mfa_token_input,
        mfa_token_button,
        None,
        'user@example.com', 'pass', 'imap.example.com', 'INBOX'
    )
    assert state['code'] == '123456'
    assert state['clicked']
```

---Extra Details---
Filename: test_mfa_email_code_fetched_via_imap.py
Description: Verifies that the code path for handling an IMAP-based email MFA retrieves, fills, and submits the 6-digit code.
Score: 96%
Alignment: 93%
Validation Notes: Mocks IMAP fetch method and button/input. Tests glue logic for email MFA scenario.