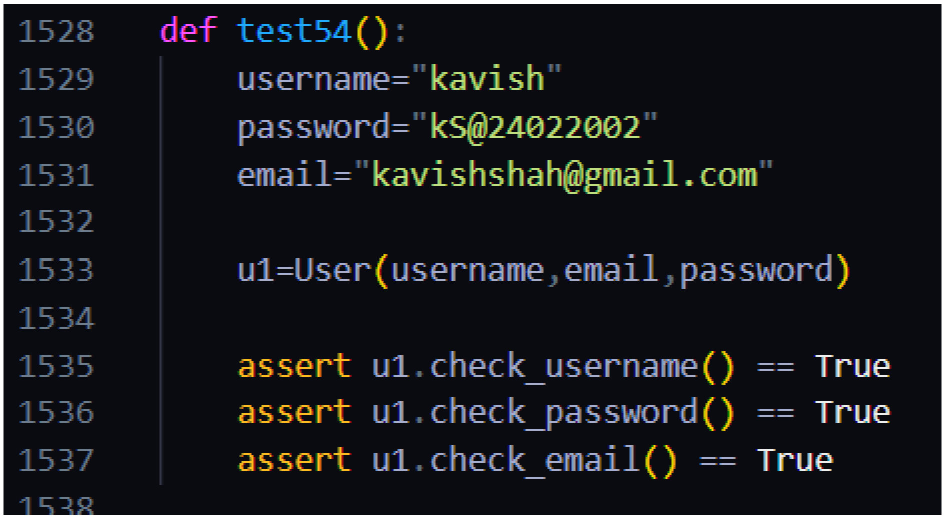
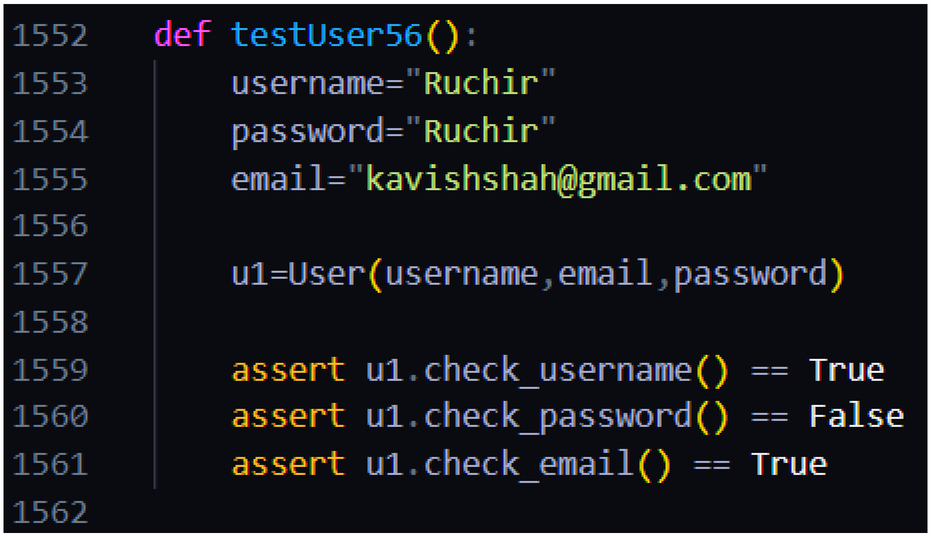
# Unit Testing for Crossword Creation

## Performed using pyTest

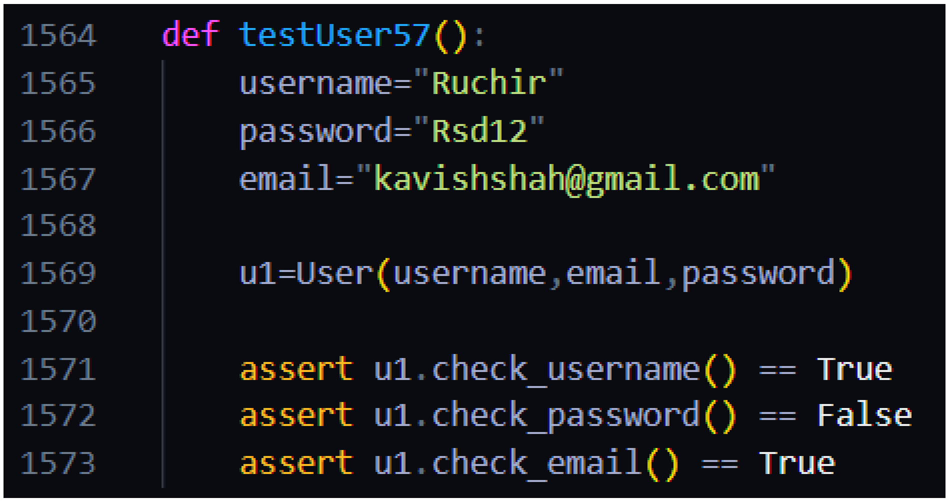
### Few Test Case:



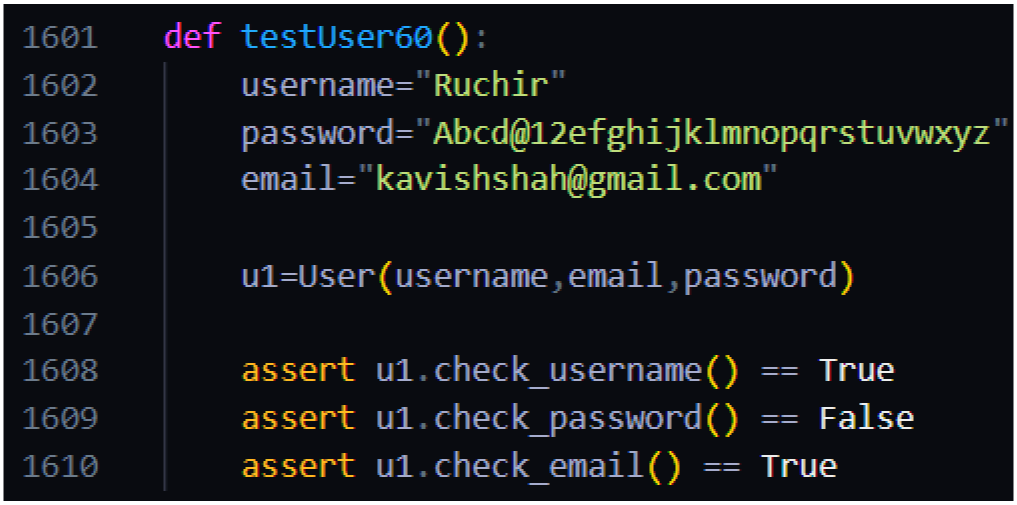
```python
def test54():
    username="kavish"
    password="kS@24022002"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```



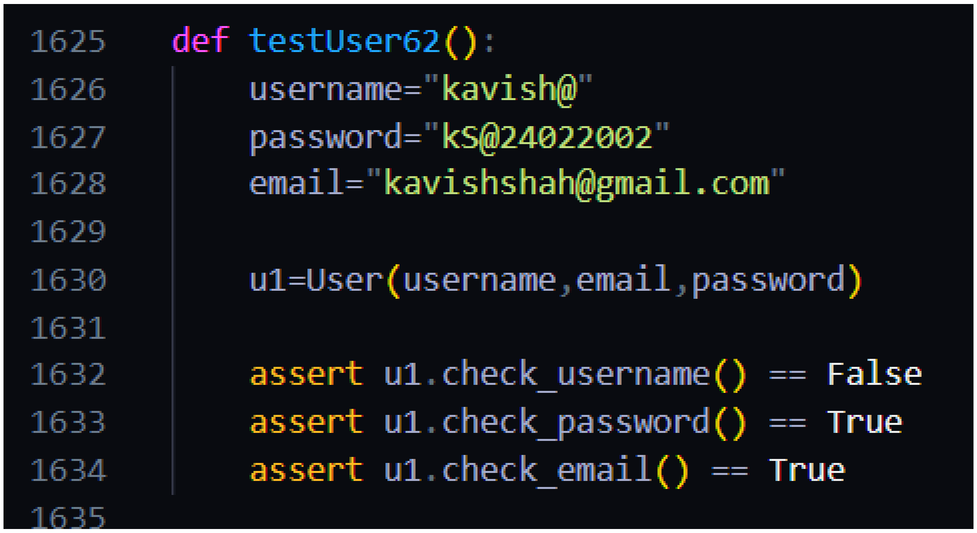
```python
def testUser56():
    username="Ruchir"
    password="Ruchir"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == False
    assert u1.check_email() == True
```



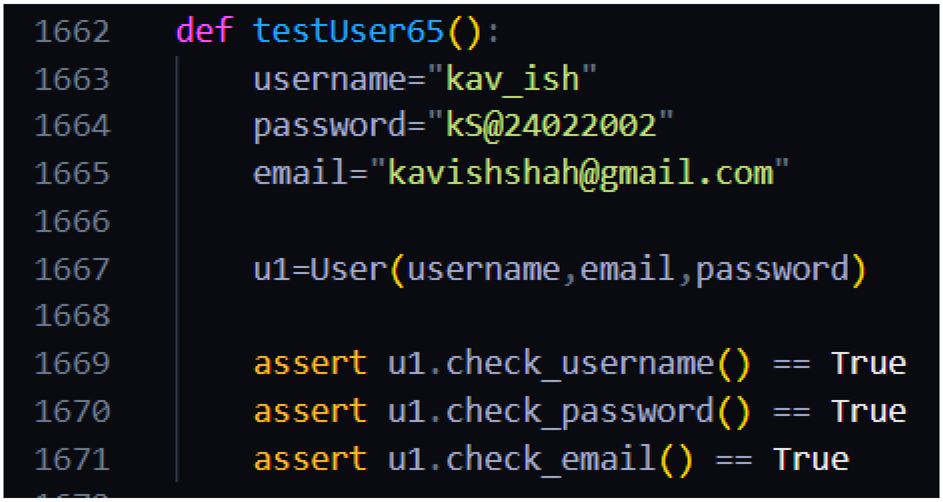
```python
def testUser57():
    username="Ruchir"
    password="Rsd12"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == False
    assert u1.check_email() == True
```



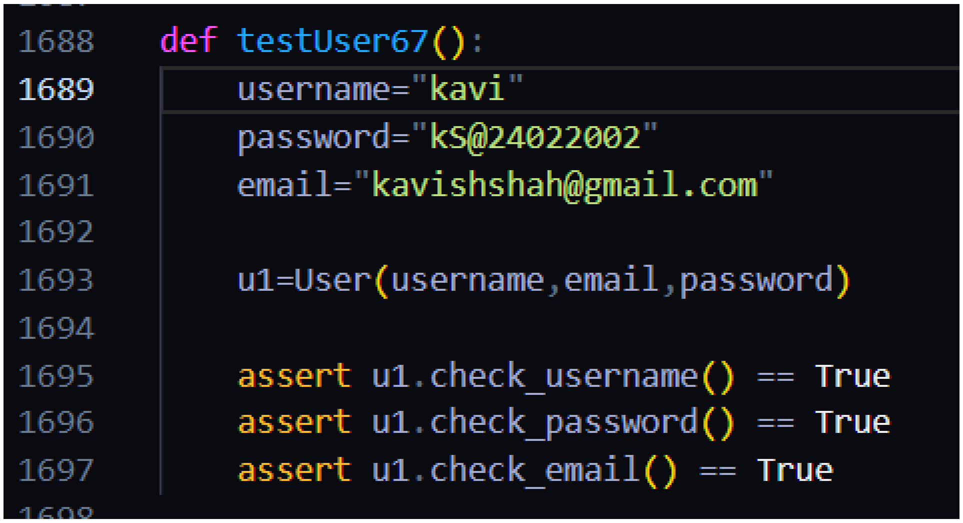
```python
def testUser60():
    username="Ruchir"
    password="Abcd@12efghijklmnopqrstuvwxyz"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == False
    assert u1.check_email() == True
```



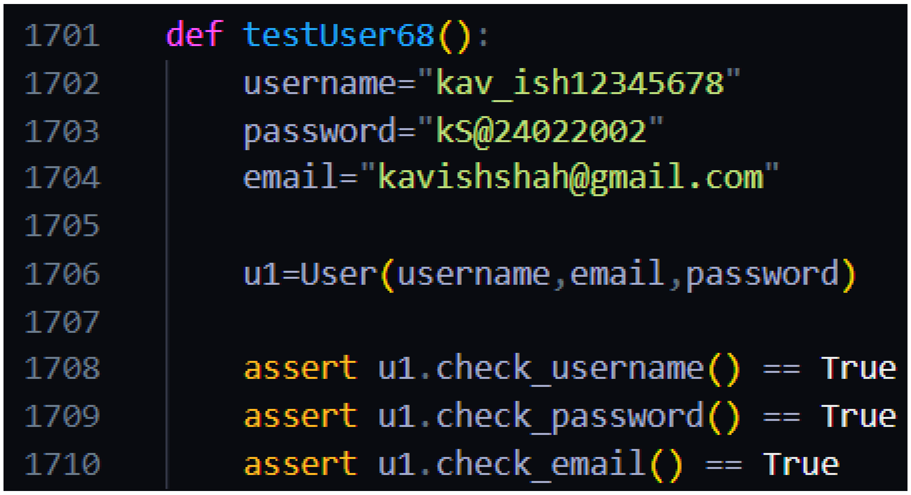
```python
def testUser62():
    username="kavish@"
    password="kS@24022002"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == False
    assert u1.check_password() == True
    assert u1.check_email() == True
```



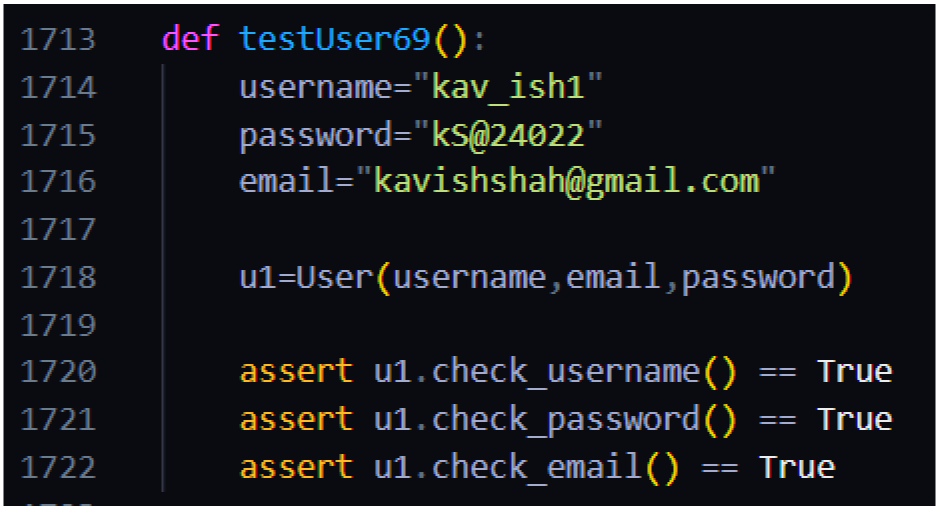
```python
def testUser65():
    username="kav_ish"
    password="kS@24022002"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```



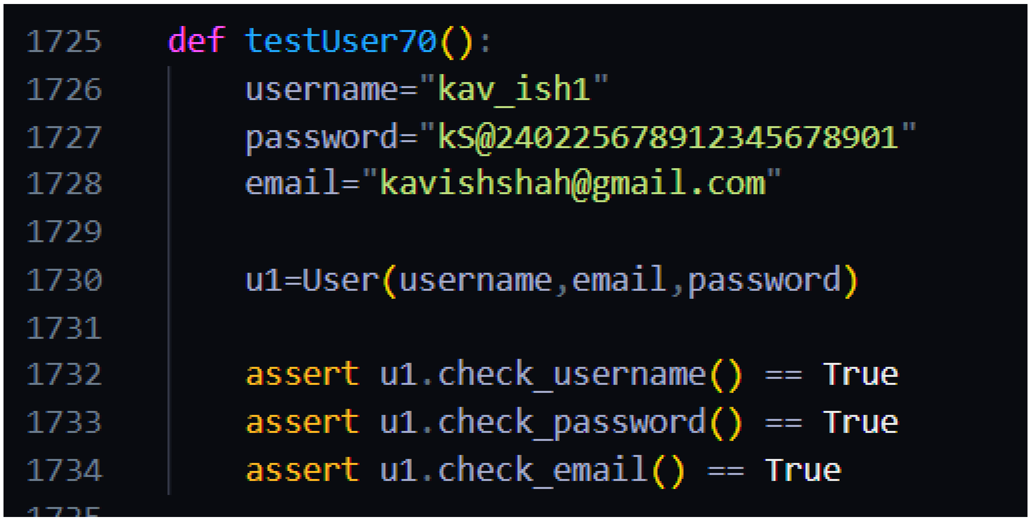
```python
def testUser67():
    username="kavi"
    password="kS@24022002"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```



```python
def testUser68():
    username="kav_ish12345678"
    password="kS@24022002"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```

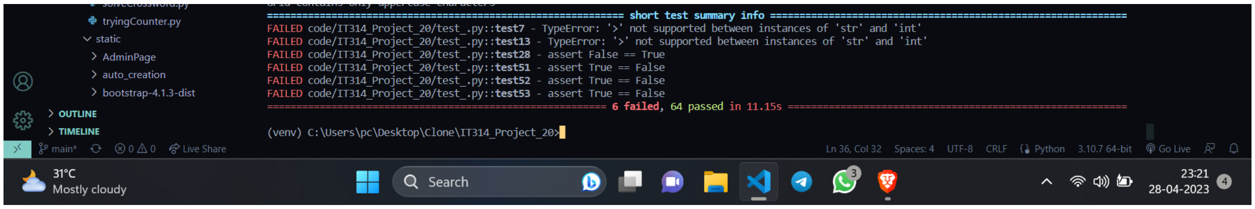


```python
def testUser69():
    username="kav_ish1"
    password="kS@24022"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```



```python
def testUser70():
    username="kav_ish1"
    password="kS@240225678912345678901"
    email="kavishshah@gmail.com"

    u1=User(username,email,password)

    assert u1.check_username() == True
    assert u1.check_password() == True
    assert u1.check_email() == True
```

- We have integrated Test Case with GitHub actions.
- Every push of Test Cases is run on it automatically.



```
================================================================ short test summary info ================================================================
FAILED code/IT314_Project_20/test_.py::test7 - TypeError: '>' not supported between instances of 'str' and 'int'
FAILED code/IT314_Project_20/test_.py::test13 - TypeError: '>' not supported between instances of 'str' and 'int'
FAILED code/IT314_Project_20/test_.py::test28 - assert False == True
FAILED code/IT314_Project_20/test_.py::test51 - assert True == False
FAILED code/IT314_Project_20/test_.py::test52 - assert True == False
FAILED code/IT314_Project_20/test_.py::test53 - assert True == False
============================================================= 6 failed, 64 passed in 11.15s =============================================================

(venv) C:\Users\pc\Desktop\Clone\IT314_Project_20>
```