

Unit Testing for Crossword Creation

Performed using pytest

Few Test Case:

```
1528 def test54():
1529     username="kavish"
1530     password="ks@24022002"
1531     email="kavishshah@gmail.com"
1532
1533     u1=User(username,email,password)
1534
1535     assert u1.check_username() == True
1536     assert u1.check_password() == True
1537     assert u1.check_email() == True
1538
```

```
1552 def testUser56():
1553     username="Ruchir"
1554     password="Ruchir"
1555     email="kavishshah@gmail.com"
1556
1557     u1=User(username,email,password)
1558
1559     assert u1.check_username() == True
1560     assert u1.check_password() == False
1561     assert u1.check_email() == True
1562
```

```
1564 def testUser57():
1565     username="Ruchir"
1566     password="Rsd12"
1567     email="kavishshah@gmail.com"
1568
1569     u1=User(username,email,password)
1570
1571     assert u1.check_username() == True
1572     assert u1.check_password() == False
1573     assert u1.check_email() == True
```

```
1601 def testUser60():
1602     username="Ruchir"
1603     password="Abcd@12efghijklmnopqrstuvwxyz"
1604     email="kavishshah@gmail.com"
1605
1606     u1=User(username,email,password)
1607
1608     assert u1.check_username() == True
1609     assert u1.check_password() == False
1610     assert u1.check_email() == True
```

```
1625 def testUser62():
1626     username="kavish@"
1627     password="ks@24022002"
1628     email="kavishshah@gmail.com"
1629
1630     u1=User(username,email,password)
1631
1632     assert u1.check_username() == False
1633     assert u1.check_password() == True
1634     assert u1.check_email() == True
1635
```

```
1662 def testUser65():
1663     username="kav_ish"
1664     password="kS@24022002"
1665     email="kavishshah@gmail.com"
1666
1667     u1=User(username,email,password)
1668
1669     assert u1.check_username() == True
1670     assert u1.check_password() == True
1671     assert u1.check_email() == True
```

```
1688 def testUser67():
1689     username="kavi"
1690     password="kS@24022002"
1691     email="kavishshah@gmail.com"
1692
1693     u1=User(username,email,password)
1694
1695     assert u1.check_username() == True
1696     assert u1.check_password() == True
1697     assert u1.check_email() == True
```

```
1701 def testUser68():
1702     username="kav_ish12345678"
1703     password="kS@24022002"
1704     email="kavishshah@gmail.com"
1705
1706     u1=User(username,email,password)
1707
1708     assert u1.check_username() == True
1709     assert u1.check_password() == True
1710     assert u1.check_email() == True
```

```

1713 def testUser69():
1714     username="kav_ish1"
1715     password="kS@24022"
1716     email="kavishshah@gmail.com"
1717
1718     u1=User(username,email,password)
1719
1720     assert u1.check_username() == True
1721     assert u1.check_password() == True
1722     assert u1.check_email() == True

```

```

1725 def testUser70():
1726     username="kav_ish1"
1727     password="kS@240225678912345678901"
1728     email="kavishshah@gmail.com"
1729
1730     u1=User(username,email,password)
1731
1732     assert u1.check_username() == True
1733     assert u1.check_password() == True
1734     assert u1.check_email() == True

```

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

The screenshot shows a VS Code editor window. On the left, the 'OUTLINE' pane is open, showing a file tree with folders like 'static', 'AdminPage', 'auto_creation', and 'bootstrap-4.1.3-dist'. The main editor area displays a 'short test summary info' section with the following content:

```

===== 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 =====

```

At the bottom of the editor, the status bar shows the file path: (venv) C:\Users\pc\Desktop\clone\IT314_Project_20\.