



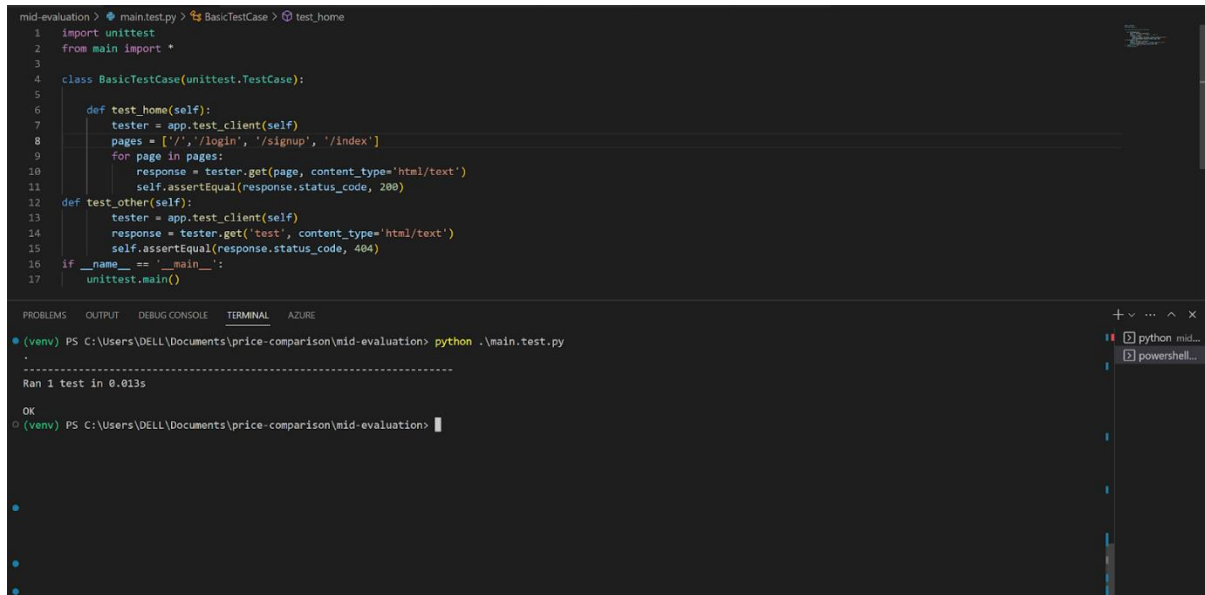
Group No: 28

Project Name: E-commerce price comparisons

Unit Testing

Identify the Test case: Index routing test

Prepare the test environment:



```
mid-evaluation > main.test.py > BasicTestCase > test_home
1 import unittest
2 from main import *
3
4 class BasicTestCase(unittest.TestCase):
5
6     def test_home(self):
7         tester = app.test_client(self)
8         pages = ['/', '/login', '/signup', '/index']
9         for page in pages:
10             response = tester.get(page, content_type='html/text')
11             self.assertEqual(response.status_code, 200)
12
13     def test_other(self):
14         tester = app.test_client(self)
15         response = tester.get('test', content_type='html/text')
16         self.assertEqual(response.status_code, 404)
17
18 if __name__ == '__main__':
19     unittest.main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE

```
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
.
Ran 1 test in 0.013s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

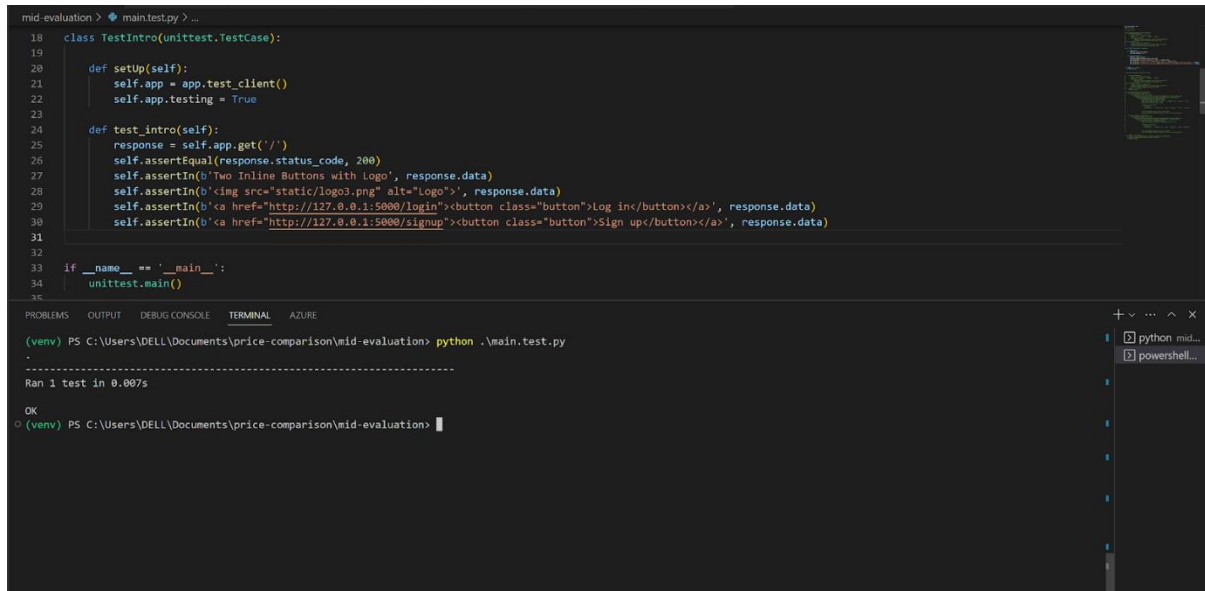
Status Code: OK - Passed

Taken Time: 0.013s

Number of Test: 1

Identify the Test case: Intro Function Test

Prepare the test environment:



```
mid-evaluation > main.test.py > ...
18 class TestIntro(unittest.TestCase):
19
20     def setUp(self):
21         self.app = app.test_client()
22         self.app.testing = True
23
24     def test_intro(self):
25         response = self.app.get('/')
26         self.assertEqual(response.status_code, 200)
27         self.assertIn(b'Two Inline Buttons with Logo', response.data)
28         self.assertIn(b'', response.data)
29         self.assertIn(b'<a href="http://127.0.0.1:5000/login"><button class="button">Log in</button></a>', response.data)
30         self.assertIn(b'<a href="http://127.0.0.1:5000/signup"><button class="button">Sign up</button></a>', response.data)
31
32
33 if __name__ == '__main__':
34     unittest.main()
35
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE

```
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
Ran 1 test in 0.007s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

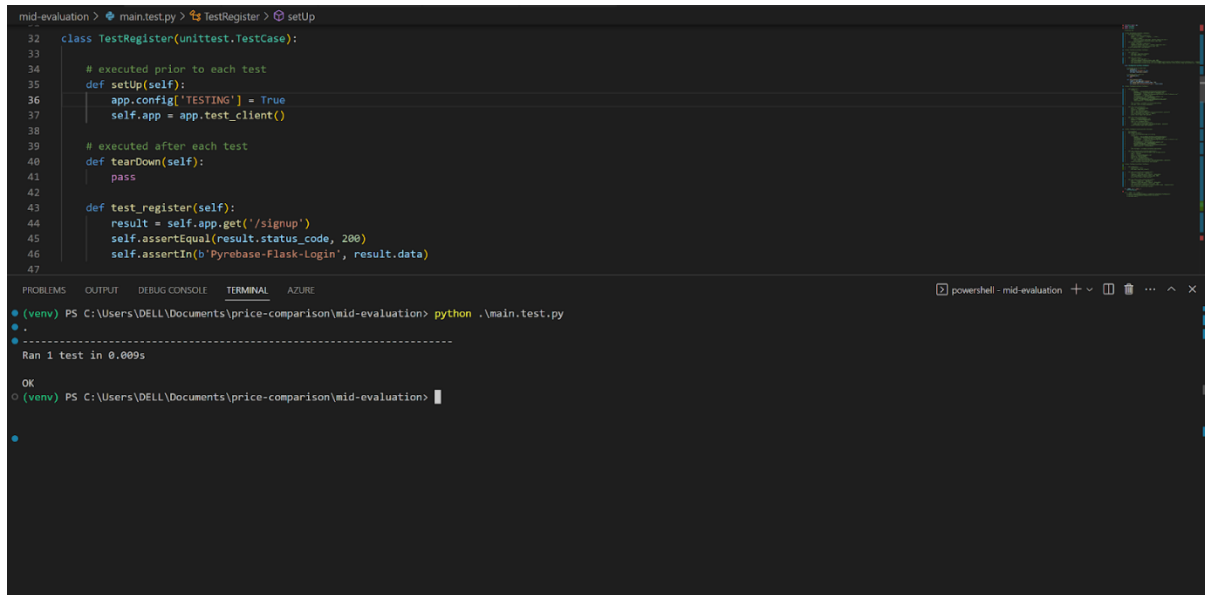
Status Code: OK - Passed

Taken Time: 0.007s

Number of Test: 1

Identify the Test case: Register function test

Prepare the test environment:



```
mid-evaluation > main.test.py > TestRegister > setUp
32 class TestRegister(unittest.TestCase):
33
34     # executed prior to each test
35     def setUp(self):
36         app.config['TESTING'] = True
37         self.app = app.test_client()
38
39     # executed after each test
40     def tearDown(self):
41         pass
42
43     def test_register(self):
44         result = self.app.get('/signup')
45         self.assertEqual(result.status_code, 200)
46         self.assertIn(b'Pyrebase-Flask-Login', result.data)
47
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
powershell - mid-evaluation + - - - - -
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
.
Ran 1 test in 0.009s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

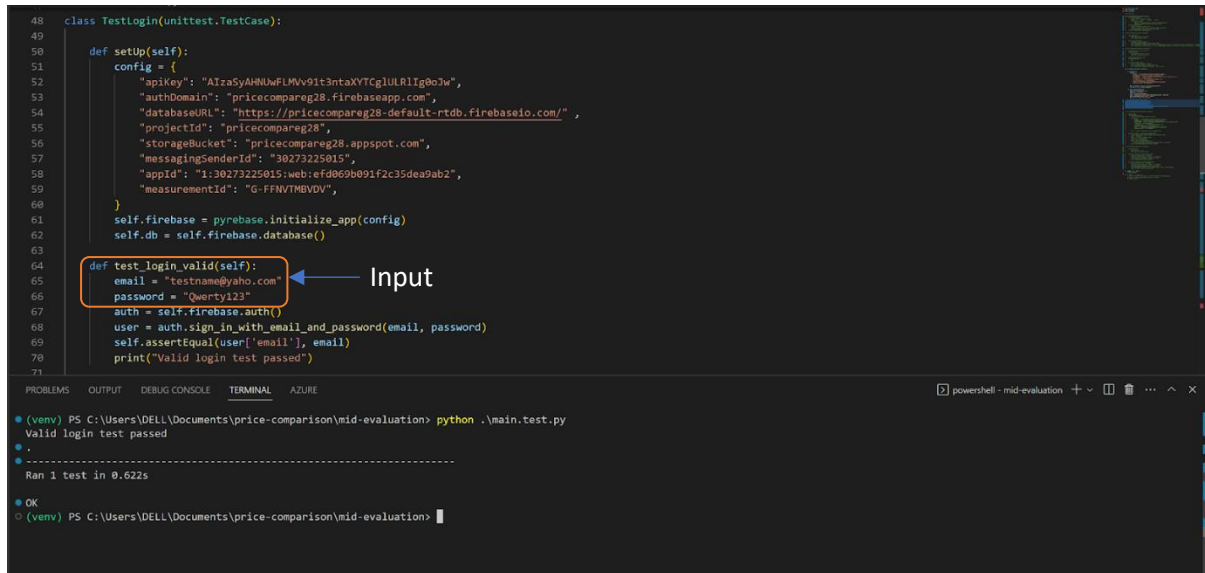
Status Code: OK - Passed

Taken Time: 0.009s

Number of Test: 1

Identify the Test case: Login functionality with valid credentials as test case

Prepare the test environment:



The screenshot displays a code editor with a Python test case and a terminal window showing the execution results. In the code editor, the `test_login_valid` function is highlighted with a red box, and a blue arrow labeled "Input" points to the email and password variables. The terminal window shows the command `python .\main.test.py` being executed, resulting in the output "Valid login test passed" and a status code of "OK".

```
48 class TestLogin(unittest.TestCase):
49
50     def setUp(self):
51         config = {
52             "apiKey": "AIzaSyAHNuwFLMvV9it3ntaKYTCgLUlRlIg0o3w",
53             "authDomain": "pricecompareg28.firebaseio.com",
54             "databaseURL": "https://pricecompareg28-default-rtdb.firebaseio.com/",
55             "projectId": "pricecompareg28",
56             "storageBucket": "pricecompareg28.appspot.com",
57             "messagingSenderId": "30273225015",
58             "appId": "1:30273225015:web:efd069b091f2c35dea9ab2",
59             "measurementId": "G-FFNV7MBVDV",
60         }
61         self.firebase = pyrebase.initialize_app(config)
62         self.db = self.firebase.database()
63
64     def test_login_valid(self):
65         email = "testname@yahoo.com"
66         password = "Qwerty123"
67         auth = self.firebase.auth()
68         user = auth.sign_in_with_email_and_password(email, password)
69         self.assertEqual(user['email'], email)
70         print("Valid login test passed")
71
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE

(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
Valid login test passed
.

Ran 1 test in 0.622s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>

Execute and record the test results:

Output: Valid login test passed

Status Code: OK - Passed

Taken Time: 0.622s

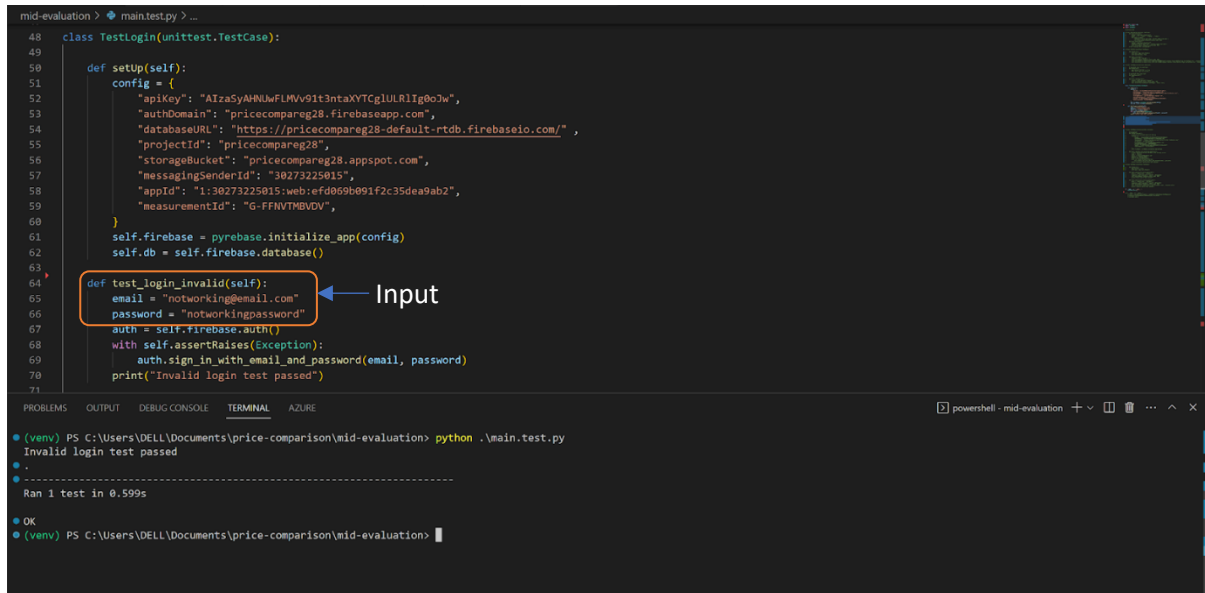
Number of Test: 1

Analyze the results:

Here, we can see that Email contain '@' and Password length is 9 which is lies between 8 to 15.

Identify the Test case: Login functionality with Invalid credentials as test case

Prepare the test environment:



The screenshot shows a VS Code editor with a Python file named `main.test.py`. The code defines a `TestLogin` class with a `setUp` method and a `test_login_invalid` test method. The `test_login_invalid` method is highlighted with a red box and an arrow pointing to it with the label "Input". The test method sets `email = "notworking@email.com"` and `password = "notworkingpassword"`, then calls `auth.sign_in_with_email_and_password(email, password)` and asserts that an exception is raised. The terminal output shows the command `python .\main.test.py` being executed, resulting in the output `Invalid login test passed` and a run time of `0.599s`.

```
48 class TestLogin(unittest.TestCase):
49
50     def setUp(self):
51         config = {
52             "apiKey": "AIzaSyAHuJwFLMvV9it3ntaXYTCgLUlRIg0o3w",
53             "authDomain": "pricecompareg28.firebaseio.com",
54             "databaseURL": "https://pricecompareg28-default-rtdb.firebaseio.com/",
55             "projectId": "pricecompareg28",
56             "storageBucket": "pricecompareg28.appspot.com",
57             "messagingSenderId": "30273225015",
58             "appId": "1:30273225015:web:efd069b091f2c35dea9ab2",
59             "measurementId": "G-FFNVMTMBVDV",
60         }
61         self.firebase = pyrebase.initialize_app(config)
62         self.db = self.firebase.database()
63
64     def test_login_invalid(self):
65         email = "notworking@email.com"
66         password = "notworkingpassword"
67         auth = self.firebase.auth()
68         with self.assertRaises(Exception):
69             auth.sign_in_with_email_and_password(email, password)
70         print("Invalid login Test passed")
71
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE

(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py

Invalid login test passed

Ran 1 test in 0.599s

OK

(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>

Execute and record the test results:

Output: Invalid login test passed

Status Code: OK - Passed

Taken Time: 0.599s

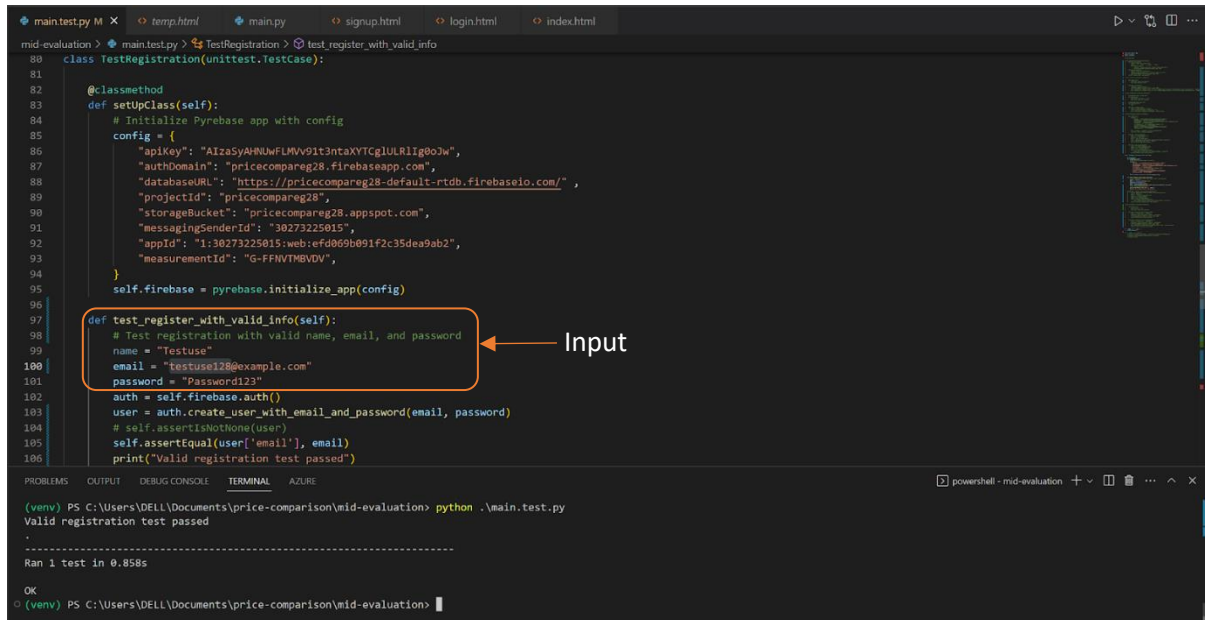
Number of Test: 1

Analyze the results:

Here, we can see that Email contain '@' and Password length is 18 which is not lies between 8 to 15.

Identify the Test case: Registration functionality with new credentials as test case

Prepare the test environment:



The screenshot shows a code editor with a file named `main.test.py`. The code defines a `TestRegistration` class that inherits from `unittest.TestCase`. It includes a `setUpClass` method to initialize a Pyrebase app with a configuration dictionary. The configuration includes fields like `apiKey`, `authDomain`, `databaseURL`, `projectId`, `storageBucket`, `messagingSenderId`, `appId`, and `measurementId`. The `test_register_with_valid_info` method is highlighted with an orange box and labeled "Input". It sets `name = "Testuser"`, `email = "testuser128@example.com"`, and `password = "Password123"`, then uses the Pyrebase auth service to create a user and asserts that the user's email is correct. The terminal output at the bottom shows the command `python .\main.test.py` being executed, resulting in "Valid registration test passed" and a runtime of 0.858s.

```
mid-evaluation > main.test.py > test_registration > test_register_with_valid_info
class TestRegistration(unittest.TestCase):
80
81
82 @classmethod
83 def setUpClass(self):
84     # Initialize Pyrebase app with config
85     config = {
86         "apiKey": "AIzaSyA#WUFLWV9it3ntaXYTCgLUlRlIg0o3w",
87         "authDomain": "pricecompare28.firebaseio.com",
88         "databaseURL": "https://pricecompare28-default-rtdb.firebaseio.com/",
89         "projectId": "pricecompare28",
90         "storageBucket": "pricecompare28.appspot.com",
91         "messagingSenderId": "30273225015",
92         "appId": "1:30273225015:web:efd069b091f2c35dea9ab2",
93         "measurementId": "G-FFNVMTMBVDV",
94     }
95     self.firebase = pyrebase.initialize_app(config)
96
97 def test_register_with_valid_info(self):
98     # Test registration with valid name, email, and password
99     name = "Testuser"
100     email = "testuser128@example.com"
101     password = "Password123"
102     auth = self.firebase.auth()
103     user = auth.create_user_with_email_and_password(email, password)
104     # self.assertIsNone(user)
105     self.assertEqual(user['email'], email)
106     print("Valid registration test passed")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
powershell - mid-evaluation + - - - - -
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
Valid registration test passed
-----
Ran 1 test in 0.858s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

Output: Valid registration test passed

Status Code: OK - Passed

Taken Time: 0.858s

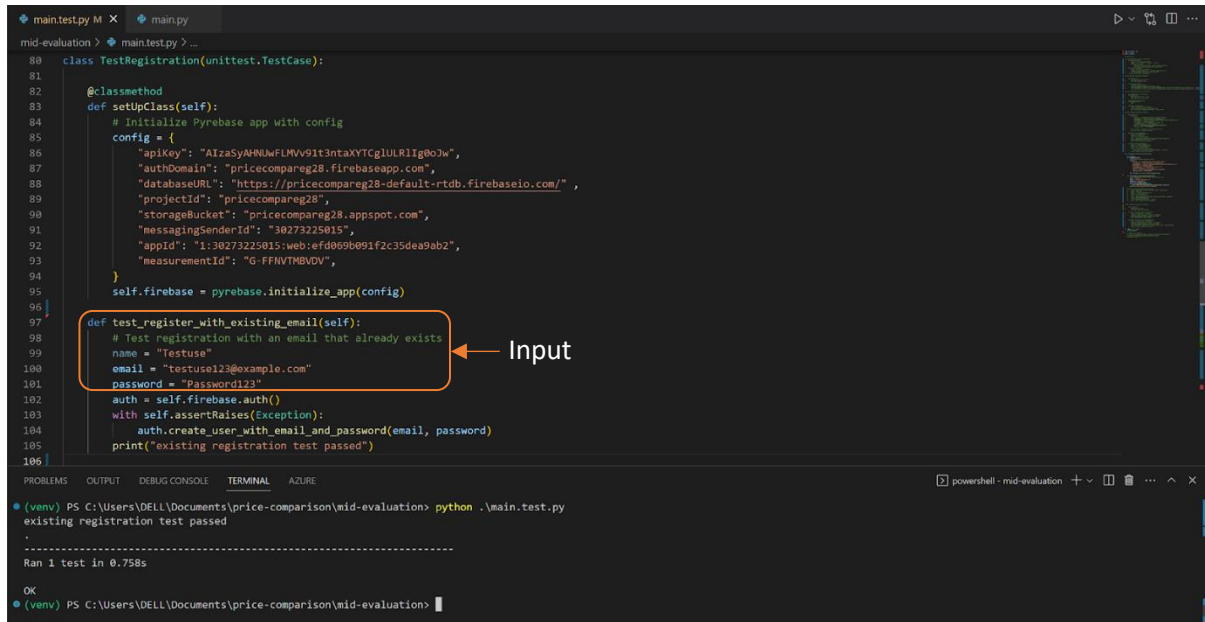
Number of Test: 1

Analyze the results:

Here, credentials are name, email, password is not registered in our system. So, this is valid registered credentials.

Identify the Test case: Registration functionality with existing credentials as test case

Prepare the test environment:



The screenshot shows a VS Code editor with a file named `main.test.py`. The code defines a `TestRegistration` class that inherits from `unittest.TestCase`. It includes a `setUpClass` method to initialize Firebase configuration and a test method `test_register_with_existing_email`. The test method uses the email `testuse123@example.com` and password `Password123`. An orange box highlights the test method, with an arrow pointing to it labeled "Input". Below the editor, the terminal shows the command `python .\main.test.py` and the output `existing registration test passed`. The terminal also shows the execution time `Ran 1 test in 0.758s` and the status `OK`.

```
80 class TestRegistration(unittest.TestCase):
81
82     @classmethod
83     def setUpClass(self):
84         # Initialize Pyrebase app with config
85         config = {
86             "apiKey": "AIzaSyA4MwF1MwV91t3ntaxYTCg1ULR1lg0oJw",
87             "authDomain": "pricecompare28.firebaseio.com",
88             "databaseURL": "https://pricecompare28-default-rtdb.firebaseio.com/",
89             "projectId": "pricecompare28",
90             "storageBucket": "pricecompare28.appspot.com",
91             "messagingSenderId": "30273225015",
92             "appId": "1:30273225015:web:efd069b091f2c35dea9ab2",
93             "measurementId": "G-FFNVTM6VDV",
94         }
95         self.firebase = pyrebase.initialize_app(config)
96
97     def test_register_with_existing_email(self):
98         # Test registration with an email that already exists
99         name = "Testuse"
100         email = "testuse123@example.com"
101         password = "Password123"
102         auth = self.firebase.auth()
103         with self.assertRaises(Exception):
104             auth.create_user_with_email_and_password(email, password)
105         print("existing registration test passed")
106
```

Output:

```
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
existing registration test passed
.
Ran 1 test in 0.758s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

Output: Existing registration test passed

Status Code: OK - Passed

Taken Time: 0.758s

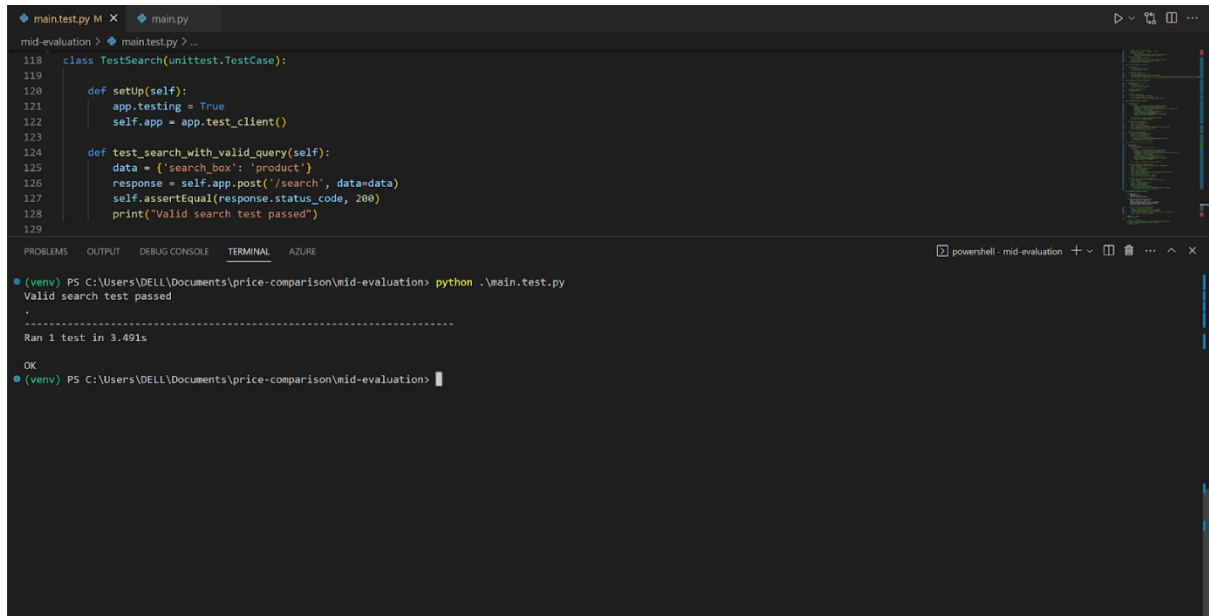
Number of Test: 1

Analyze the results:

In this case, the user credentials are already exists (actually we registered the same credentials in above case).

Identify the Test case: Search functionality with valid query as test case

Prepare the test environment:



The screenshot shows a VS Code editor with a file named `main.test.py` open. The code defines a `TestSearch` class that inherits from `unittest.TestCase`. It includes a `setUp` method and a `test_search_with_valid_query` method. The `test_search_with_valid_query` method sends a POST request to `/search` with data `{'search_box': 'product'}` and asserts that the status code is 200. The terminal output shows the command `python .\main.test.py` being executed, resulting in the message `Valid search test passed`, a separator line, and the execution time `Ran 1 test in 3.491s`. The status `OK` is displayed at the bottom of the terminal output.

```
118 class TestSearch(unittest.TestCase):
119
120     def setUp(self):
121         app.testing = True
122         self.app = app.test_client()
123
124     def test_search_with_valid_query(self):
125         data = {'search_box': 'product'}
126         response = self.app.post('/search', data=data)
127         self.assertEqual(response.status_code, 200)
128         print("Valid search test passed")
129
```

```
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
Valid search test passed
-----
Ran 1 test in 3.491s
OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

Output: Valid Search test passed

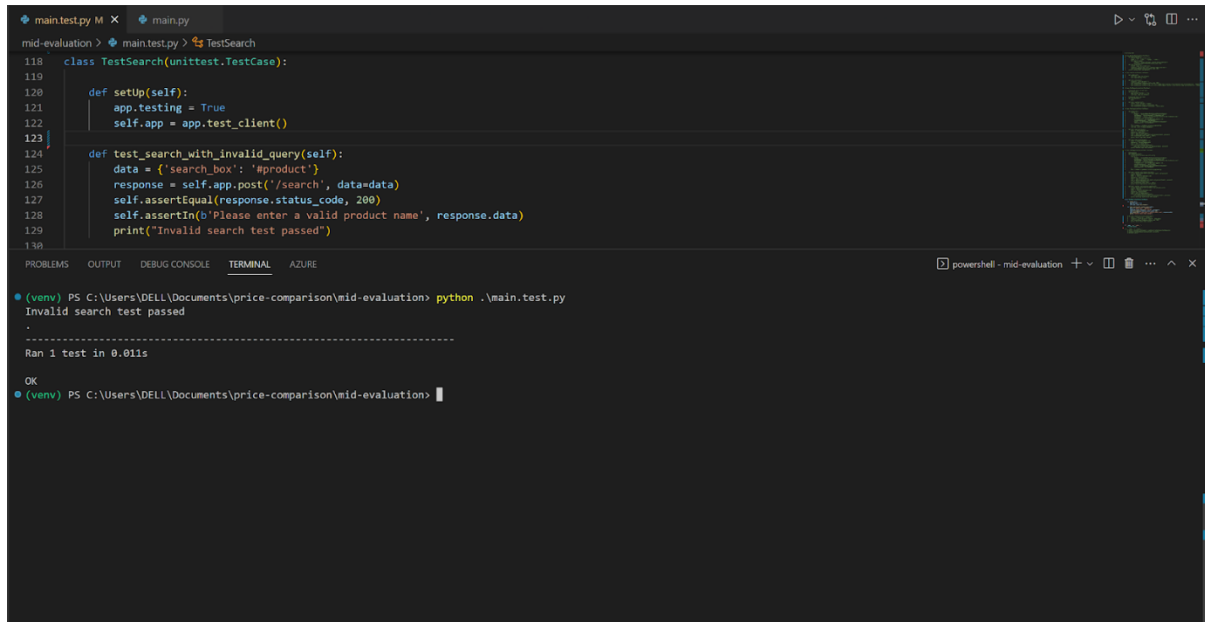
Status Code: OK - Passed

Taken Time: 3.491s

Number of Test: 1

Identify the Test case: Search functionality with Invalid query as test case

Prepare the test environment:



The screenshot shows a VS Code editor with a file named `main.test.py` open. The code defines a `TestSearch` class that inherits from `unittest.TestCase`. It includes a `setUp` method and a `test_search_with_invalid_query` method. The `test_search_with_invalid_query` method sends a POST request to `/search` with data `{'search_box': '#product'}` and asserts that the status code is 200 and the response data contains the string `'Please enter a valid product name'`. Below the editor, the terminal shows the command `python .\main.test.py` being executed, resulting in the output `Invalid search test passed`, `OK`, and `Ran 1 test in 0.011s`.

```
main.test.py M X main.py
mid-evaluation > main.test.py > TestSearch
118 class TestSearch(unittest.TestCase):
119
120     def setUp(self):
121         app.testing = True
122         self.app = app.test_client()
123
124     def test_search_with_invalid_query(self):
125         data = {'search_box': '#product'}
126         response = self.app.post('/search', data=data)
127         self.assertEqual(response.status_code, 200)
128         self.assertIn('Please enter a valid product name', response.data)
129         print("Invalid search test passed")
130
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
Invalid search test passed
OK
Ran 1 test in 0.011s
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

Output: Invalid Search test passed

Status Code: OK - Passed

Taken Time: 0.011s

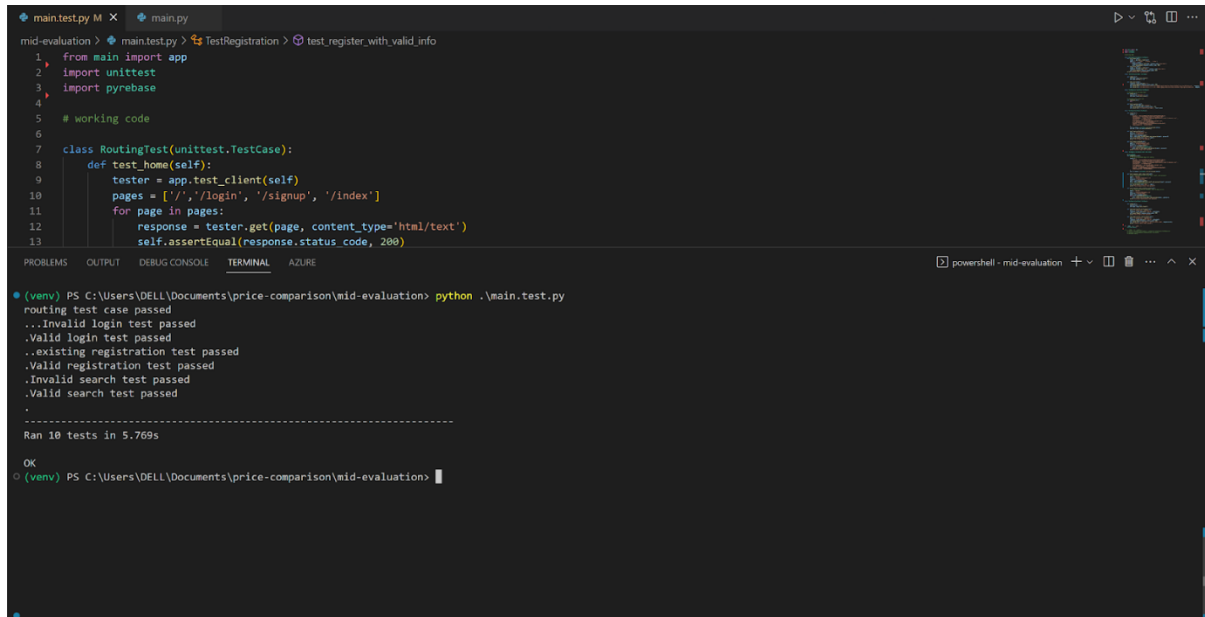
Number of Test: 1

Analyze the results:

Here, search name contains special character.

Identify the Test case: Running all test case functions together

Prepare the test environment:



The screenshot shows a VS Code editor with a file named `main.test.py` open. The code defines a `RoutingTest` class that inherits from `unittest.TestCase`. It includes a `test_home` method that uses `app.test_client` to test various pages. The terminal output shows the results of running the tests, indicating that all 10 tests passed successfully.

```
1 from main import app
2 import unittest
3 import pyrebase
4
5 # working code
6
7 class RoutingTest(unittest.TestCase):
8     def test_home(self):
9         tester = app.test_client(self)
10         pages = ['/', '/login', '/signup', '/index']
11         for page in pages:
12             response = tester.get(page, content_type='html/text')
13             self.assertEqual(response.status_code, 200)
```

```
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation> python .\main.test.py
routing test case passed
...Invalid login test passed
.Valid login test passed
..existing registration test passed
.Valid registration test passed
.Invalid search test passed
.Valid search test passed
.
-----
Ran 10 tests in 5.769s

OK
(venv) PS C:\Users\DELL\Documents\price-comparison\mid-evaluation>
```

Execute and record the test results:

Output: Routing test case passed, Invalid login test passed, Valid login test passed, etc.

Status Code: OK - Passed

Taken Time: 5.769s

Number of Test: 10