

cn331-as2\DjangoWeb\courses\tests.py

```
1  from django.test import TestCase
2  from .models import Course
3  from django.contrib.auth.models import User
4
5  class CourseTestCase(TestCase):
6
7      def setUp(self):
8          course1 = Course.objects.create(c_id="test1", max_seat=2)
9          course2 = Course.objects.create(c_id="test2")
10
```

สร้าง objects course1 โดยกำหนดให้ c_id = "test1" max_seat=2 และ object course2 มีค่า c_id = "test2"

```
11      def test_seat_available(self):
12          """ is_seat_available should be True """
13
14          course = Course.objects.first()
15          self.assertTrue(course.is_seat_available())
16
```

กำหนดให้ course = course1 ใน setUp แล้วทำการเช็ค ผ่าน function is_seat_available ใน class Course ว่าเป็น True หรือไม่

```
16
17      def test_seat_not_available(self):
18          """ is_seat_available should be False """
19
20          student1 = User.objects.create(
21              username="harry", password="potter")
22          student2 = User.objects.create(
23              username="hermione", password="granger")
24          course = Course.objects.first()
25          course.attend.add(student1)
26          course.attend.add(student2)
27          self.assertFalse(course.is_seat_available())
28
```

ทำการสร้าง user student1 โดยกำหนดให้ username = "harry" และ password = "potter" และ student2 โดยกำหนดให้ username = "hermione" และ password = "granger" แล้วทำการเพิ่ม user1 และ user2 เข้าไปใน course1 ที่มี max_seat = 2 ทำให้ที่นั่งเต็ม แล้วทำการเช็ค ผ่าน function is_seat_available ใน class Course ว่าเป็น False หรือไม่

```

29     def test_tostring_model(self):
30         course = Course.objects.get(c_id='test1')
31         self.assertEqual(str(course), "code:test1 seat count:0 seat max:2 quota:True")

```

กำหนดให้ course = course1 ผ่านค่า c_id แล้วทำการเช็ค function __str__ ว่ามีค่าตรงกับที่กำหนดให้ course1 หรือไม่

cn331-as2\DjangoWeb\courses\test_views.py

```

1  from django.test import TestCase, Client
2  from django.urls import reverse
3  from django.db.models import Max
4  from .models import Course
5  from django.contrib.auth.models import User
6  from django.contrib.auth.hashers import make_password
7
8
9  class CourseViewTestCase(TestCase):
10
11     def setUp(self):
12         self.client = Client()
13         password_k = make_password('kritpassword')
14         password_p = make_password('pangpassword')
15         user_k = User.objects.create(username='krit', password=password_k)
16         user_p = User.objects.create(username='pang', password=password_p)
17         course1 = Course.objects.create(c_id="test1")
18         course2 = Course.objects.create(c_id="test2")
19         course1.attend.add(user_k)
20         course2.attend.add(user_p)
21

```

กำหนดค่า self.client ให้เป็น function ของ Client

ทำการ make_password() แล้ว สร้าง User ชื่อว่า user_k และ user_p โดยกำหนด username password สร้าง object course1 และ course2 โดย กำหนด c_id = "test1" และ c_id = "test2" ตามลำดับ แล้วเพิ่ม users เข้าไปใน course1 และ course2 ตามลำดับ

```

22     def test_index_view_status_code(self):
23         self.client.login(username='krit', password='kritpassword')
24         response = self.client.get(reverse('c_index'))
25         self.assertEqual(response.status_code, 200)

```

ล็อกอินด้วย username ชื่อ krit เมื่อทำการล็อกอินเข้าไปแล้ว เช็คว่าเจอ c_index page หรือไม่

```

26
27     def test_index_view_context(self):
28         """ context is correctly set """
29
30         self.client.login(username='krit', password='kritpassword')
31         response = self.client.get(reverse('c_index'))
32         self.assertEqual(
33             response.context['courses'].count(), 2)
34

```

ล็อกอินด้วย username ชื่อ krit แล้วเช็คว่ามีจำนวน course = 2 ตามที่ได้กำหนดไว้ใน setUp หรือไม่

```

34
35     def test_valid_course_page(self):
36         """ valid course page should return status code 200 """
37
38         self.client.login(username='krit', password='kritpassword')
39         f = Course.objects.first()
40         response = self.client.get(reverse('course', args=(f.id,)))
41         self.assertEqual(response.status_code, 200)

```

ล็อกอินด้วย username ชื่อ krit และ สร้าง object f = course1 แล้วเช็คว่าสามารถเข้าไปดูหน้ารายละเอียด course1 ได้หรือไม่

```

43
44     def test_invalid_course_page(self):
45         """ invalid flight page should return status code 404 """
46
47         self.client.login(username='krit', password='kritpassword')
48         max_id = Course.objects.all().aggregate(Max("id"))['id__max']
49         response = self.client.get(
50             Course, pk=reverse('course', args=(max_id+1,)))
51         self.assertEqual(response.status_code, 404)
52

```

ล็อกอินด้วย username ชื่อ krit และเมื่อเข้าหน้ารายละเอียด course ที่ไม่มีอยู่ จะไม่พบหน้าเพจ

```

53 def test_cannot_book_nonavailable_seat_course(self):
54     """ cannot book full max seat course"""
55
56     self.client.login(username='pang', password='pangpassword')
57     f = Course.objects.first()
58     f.max_seat = 1
59     f.save()
60     self.client.post(reverse('book', args=(f.id,)))
61     self.assertEqual(f.attend.count(), 1)
62

```

ล็อกอินด้วย username ชื่อ pang สร้าง object f = course1 กำหนดให้มี max_seat = 1 ซึ่งมี attend เป็น 1 อยู่แล้วในตอน setUp แล้วเมื่อทำการ book course จะทำไม่สำเร็จ แล้วเช็คว่า course จะมี attend เป็น 1 เหมือนเดิมหรือไม่

```

63 def test_success_book_available_seat_course(self):
64
65
66     self.client.login(username='krit', password='kritpassword')
67     f = Course.objects.get(c_id="test2")
68     self.client.post(reverse('book', args=(f.id,)))
69     self.assertEqual(f.attend.count(), 2)
70

```

ล็อกอินด้วย username ชื่อ krit สร้าง object f = course2 แล้วเมื่อทำการ book course เช็คว่า course จะมี attend เพิ่มขึ้นเป็น 2 หรือไม่

```

70
71 def test_success_book_available_seat_course_full(self):
72
73     self.client.login(username='pang', password='pangpassword')
74     course = Course.objects.get(c_id='test1')
75     course.max_seat=2
76     self.client.get(reverse('book', args=(course.id,)))
77     course.refresh_from_db()
78     self.assertEqual(course.quota, False)

```

ล็อกอินด้วย username ชื่อ pang สร้าง object course = course1 กำหนด max_seat = 2 แล้วทำการ book course1 ทำให้ที่นั่งเต็ม แล้วเช็คว่า quota เปลี่ยนเป็น False หรือไม่

cn331-as2\DjangoWeb\students\tests.py

```
1 from django.contrib.auth.hashers import make_password
2 from django.contrib.auth.models import User
3 from django.test import TestCase, Client
4 from django.urls import reverse
5 from courses.models import Course
6
7 # Create your tests here.
8
9 class UserTestCase(TestCase):
10     def setUp(self):
11         self.client = Client()
12         password = make_password('kritpassword')
13         user = User.objects.create(username='krit', password=password)
14         Course.objects.create(c_id="test1")
15
```

กำหนดค่า self.client ให้เป็น function ของ Client

ทำการ make_password() แล้ว สร้าง User ชื่อว่า user โดยกำหนด username password

สร้าง object course โดยที่ c_id = "test1"

```
15
16     def test_redirect_reverse_login(self):
17         response = self.client.get(reverse('index'))
18         self.assertRedirects(response, reverse('login'))
19
```

เชื่อว่าเมื่อไม่ได้ล็อกอินแล้วเข้าหน้า homepage จะทำให้กลับไปหน้า login

```
20     def test_authenticated_user_index_view(self):
21         self.client.login(username='krit', password='kritpassword')
22         response = self.client.get(reverse('index'))
23         self.assertEqual(response.status_code, 200)
24
```

เมื่อทำการ login แล้วเชื่อว่าจะไปหน้า homepage หรือไม่

```

25     def test_login_view(self):
26         response = self.client.get(reverse('login'))
27         self.assertEqual(response.status_code, 200)
28

```

เช็คว่าสามารถเข้าหน้า login ได้หรือไม่

```

28
29     def test_can_login(self):
30         someone = {'username': 'krit', 'password': 'kritpassword'}
31         response = self.client.post(reverse('login'), someone, follow=True)
32         self.assertEqual(response.status_code, 200)
33

```

เช็คว่าสามารถล็อกอินด้วย username และ password ที่ถูกต้องได้หรือไม่

```

34     def test_logout_view(self):
35         self.client.login(username='krit', password='kritpassword')
36         response = self.client.post(reverse('logout'))
37         self.assertEqual(response.status_code, 200)
38

```

ล็อกอินด้วย username ชื่อ krit แล้วทำการ logout เช็คว่าจะไปที่หน้า logout หรือไม่

```

38
39     def test_warning_message_login(self):
40         someone = {'username': 'krit', 'password': 'wrongpassword'}
41         response = self.client.post(reverse('login'), someone, follow=True)
42         messages = list(response.context['message'])
43         self.assertEqual(len(messages), 15)
44         self.assertEqual("".join(messages), 'Invalid member.')
45

```

ล็อกอินด้วย password ที่ผิด เช็คว่าสามารถล็อกอินได้สำเร็จหรือไม่ ถ้าไม่สำเร็จจะขึ้นว่า Invalid member

```

46     def test_success_message_logout(self):
47         self.client.login(username='krit', password='kritpassword')
48         response = self.client.post(reverse('logout'))
49         messages = list(response.context['message'])
50         self.assertEqual(len(messages), 10)
51         self.assertEqual("".join(messages), 'Logged out')
52

```

ล็อกอินด้วย username ชื่อ krit แล้วทำการ logout เช็คค่า message ขึ้นว่า Logged out หรือไม่

```

53     def test_cancle_view(self):
54         response = self.client.get(reverse('quota'))
55         self.assertEqual(response.status_code, 200)
56

```

เช็คค่าสามารถเข้าหน้า quota ได้หรือไม่

```

57     def test_withdraw_course(self):
58
59         self.client.login(username='krit', password='kritpassword')
60         course = Course.objects.first()
61         course.attend.add(User.objects.get(username='krit'))
62         response = self.client.get(reverse('cancel', args=(course.id,)))
63         self.assertEqual(course.attend.count(), 0)
64

```

ล็อกอินด้วย username ชื่อ krit แล้วเพิ่ม krit เข้าไปใน course จากนั้น withdraw แล้วเช็คจำนวน attend ใน course เป็น 0 หรือไม่

```

65     def test_redirect_cancel_course(self):
66         self.client.login(username='krit', password='kritpassword')
67
68         course = Course.objects.first()
69         response = self.client.get(reverse('cancel', args=(course.id,)))
70         self.assertRedirects(response, reverse('quota'))
71

```

ล็อกอินด้วย username ชื่อ krit แล้วกด withdraw เช็คค่าเมื่อทำการถอนวิชา จะกลับไปหน้า quota ใหม่

ผลการรัน

```
ADVICE@tu /d/CN331/cn331-as4/DjangoWeb
$ coverage run manage.py test
Found 20 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 20 tests in 3.999s

OK
Destroying test database for alias 'default'...
(venv)
```


\$ coverage report

Name	Stmts	Miss	Cover
<hr/>			
DjangoWeb__init__.py	0	0	100%
DjangoWeb\settings.py	21	0	100%
DjangoWeb\urls.py	3	0	100%
courses__init__.py	0	0	100%
courses\admin.py	6	0	100%
courses\apps.py	4	0	100%
courses\migrations\0001_initial.py	6	0	100%
courses\migrations\0002_course_attendance_alter_course_max_seat.py	5	0	100%
courses\migrations\0003_remove_course_attendance.py	4	0	100%
courses\migrations\0004_alter_course_max_seat.py	4	0	100%
courses\migrations\0005_alter_course_c_id.py	4	0	100%
courses\migrations__init__.py	0	0	100%
courses\models.py	16	0	100%
courses\test_views.py	54	0	100%
courses\tests.py	20	0	100%
courses\urls.py	3	0	100%
courses\views.py	27	0	100%
manage.py	12	2	83%
students__init__.py	0	0	100%
students\admin.py	1	0	100%
students\apps.py	4	0	100%
students\migrations__init__.py	0	0	100%
students\models.py	1	0	100%
students\tests.py	55	0	100%
students\urls.py	3	0	100%
students\views.py	37	0	100%
<hr/>			
TOTAL	290	2	99%