

Peerapat Ngamsanga 64106985223

Peekkan Piankan 6410615097

## Coverage report

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ coverage report
```

Name	Stmts	Miss	Cover
Students\__init__.py	0	0	100%
Students\admin.py	3	0	100%
Students\apps.py	4	0	100%
Students\migrations\0001_initial.py	5	0	100%
Students\migrations\__init__.py	0	0	100%
Students\models.py	9	0	100%
Students\tests.py	1	0	100%
Students\urls.py	0	0	100%
Students\views.py	0	0	100%
course\__init__.py	0	0	100%
course\admin.py	9	0	100%
course\apps.py	4	0	100%
course\forms.py	5	0	100%
course\migrations\0001_initial.py	6	0	100%
course\migrations\0002_alter_class_of_students_course.py	4	0	100%
course\migrations\0003_alter_class_of_students_course.py	4	0	100%
course\migrations\__init__.py	0	0	100%
course\models.py	28	0	100%
course\tests.py	141	0	100%
course\urls.py	3	0	100%
course\views.py	60	0	100%
login\__init__.py	0	0	100%
login\admin.py	1	0	100%
login\apps.py	4	0	100%
login\migrations\0001_initial.py	5	0	100%
login\migrations\__init__.py	0	0	100%
login\models.py	7	0	100%
login\tests.py	29	0	100%
login\urls.py	3	0	100%
login\views.py	16	0	100%
manage.py	12	2	83%
register\__init__.py	0	0	100%
register\asgi.py	4	4	0%
register\settings.py	21	0	100%
register\urls.py	3	0	100%
register\wsgi.py	4	4	0%
TOTAL	395	10	97%
(. venv)			

# Login app

```
def home_view(request,*args, **kwargs):  
    return HttpResponseRedirect(reverse('login'))
```

```
# create your tests here.  
class HomeViewTest(TestCase):  
  
    def setUp(self) -> None:  
        self.response = self.client.get(reverse('home'))  
  
    def test_Found(self):  
        self.assertEqual(self.response.status_code,302)  
  
    def test_reverse_login(self):  
        self.assertRedirects(self.response, reverse('login'))
```

ทำ 2 เทส โดย function test\_found(self) ใช้ ตรวจสอบ status code redirect ส่วน function test\_reverse\_login ใช้  
ตรวจสอบว่า reverse('login') มาถูกต้องมั้ย

ผลการรันเทส

```
USER@tu ~/Documents/work/jack/cn331/register (testing)  
$ python manage.py test login.tests.HomeViewTest  
Found 2 test(s).  
Creating test database for alias 'default'...  
System check identified no issues (0 silenced).  
..  
-----  
Ran 2 tests in 0.037s  
  
OK  
Destroying test database for alias 'default'...  
(.venv)
```

```
def login_view(request):
    if request.method == 'POST':
        # Perform authentication
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:
            login(request, user)
            return HttpResponseRedirect(reverse('course_list'))
        return render(request, 'registration/login.html', {'error_message': 'Invalid login credentials'})
    return render(request, 'registration/login.html') # Display the login page
```

```
class LoginViewTest(TestCase):

    def setUp(self) -> None:
        self.user = User.objects.create_user(username="6410685223",password="kiw123456789")
        self.login_url = reverse('login')

    def test_status_code(self):
        response = self.client.get(self.login_url)
        self.assertEqual(response.status_code,200)
        self.assertTemplateUsed(response, 'registration/login.html')

    def test_login_redirect(self):
        response = self.client.post(self.login_url,{'username':'6410685223','password':'kiw123456789'})
        self.assertEqual(response.status_code,302)
        self.assertRedirects(response, reverse('course_list'))

    def test_login_invalid(self):
        response = self.client.post(self.login_url, {'username': 'testuser', 'password': 'wrongpassword'})
        self.assertEqual(response.status_code, 200) # Expecting a re-render of the login page
        self.assertTemplateUsed(response, 'registration/login.html')
        self.assertContains(response, 'Invalid login credentials')
```

ทำเทสทั้ง 3 test โดนผมทำการ setup รหัส User กับ reverse('login')

Test 1: function test\_status\_code โดยฟังก์ชันนี้ทำการเทส status หลังจากทำการ get reverse('login') และทำการเช็ค template ที่ตอบสนองกลับมากับมา

Test 2: function test\_login\_redirect โดยฟังก์ชันนี้ทำการเทส status code redirect == 302 และ check response Reverse('course\_list') หรือไม่

Test 3: function test\_login\_invalid โดยฟังก์ชันนี้จะเทสทั้งหมด 3 อย่าง คือ Check status code, check template Response == 'registration/login.html', check message by self.assertContains

ผลลัพธ์

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test login.tests.LoginViewTest
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 1.050s

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

# Course App

```
def course_list(request):
    courses = Course.objects.all()
    check = set()
    # print(courses)
    for i in class_of_students.objects.values_list('student_id','course'):
        if str(i[0]) == str(request.user):
            check.add(i[1])
    return render(request, 'course/course_list.html', {
        'courses': courses,
        'subject': check,
    })
```

```
class ClassListTestCase(TestCase):
    def setUp(self) -> None:
        # set user
        self.user = User.objects.create_user(username='6410685249',password='first123456789')
        self.client.login(username='6410685249', password='first123456789')
        self.user_info = student_info.objects.create(student_username='6410685249',student_password='first123456789',
            student_surname='Siriphatson',student_lastname='sringamphong')

        # set class
        self.cn202 = Course(course_id='CN202',course_name='Algorithm 2',course_term='1',course_year='2023',
            available_seat=10,course_is_open=True)
        self.cn203 = Course(course_id='CN203',course_name='Algorithm 3',course_term='1',course_year='2023',
            available_seat=10,course_is_open=True)
        self.cn202.save()
        self.cn203.save()

        self.temp = class_of_students.objects.create(student_id='6410685249',student_name='Siriphatson',student_lastname='sringamphong')
        self.temp.course.add('CN202')
        # set up response
        self.response = self.client.get(reverse('course_list'))

    def test_add_course(self):
        self.assertEqual(self.temp.course.first(), self.cn202)

    def test_course_list_response(self):
        self.assertEqual(self.response.status_code, 200)
```

```
def test_render_template(self):
    self.assertTemplateUsed(self.response, 'course/course_list.html')

def test_response_message(self):
    self.assertQuerysetEqual(set(self.response.context['courses']), set(Course.objects.all()))
    self.assertQuerysetEqual(''.join(self.response.context['subject']), Course.objects.get(course_id='CN202').course_id)
```

ทำการทดสอบทั้งหมด 4 function

setup ทำการ add user สำหรับ login และ ทำการ login by self.client.login และ ทำการ add student information

ได้ทำการเพิ่มรายวิชา 2 วิชา CN202, CN203 หลังจากนั้นทำการผูก student กับวิชานั้นๆ

สุดท้ายทำการ set self.response = self.client.get(reverse('course\_list'))

Function test\_add\_course : ทำการทดสอบว่าวิชาแรกใช่ cn202 หรือไม่

Function test\_course\_list\_response: ทดสอบ status code == 200

Function test\_render\_template: เช็ค template ของ response == 'course/course\_list.html'

Function test\_response\_message: message courses (object),subject (course id) ที่ทำการใช้กับ html

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test course.tests.ClassListTestCase
Found 4 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
....
-----
Ran 4 tests in 1.742s

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

```
def custom_logout(request):
    if request.method == 'POST':
        auth_logout(request)
        return redirect(reverse('login'))
    return redirect(reverse('course_list'))
```

```
class custom_logout(TestCase):

    def test_redirect_without_post(self):
        response = self.client.get(reverse('logout'))
        self.assertEqual(response.status_code, 302)
        self.assertRedirects(response, reverse('course_list'))

    def test_redirect_logout_post(self):
        user = User.objects.create_user(username='testuser', password='testpassword')
        self.client.login(username='testuser', password='testpassword')
        response = self.client.post(reverse('logout'))
        self.assertRedirects(response, reverse('login'))
```

Function test\_redirect\_without\_post: check status redirects code,

check redirects response == reverse('course\_list')

Function test\_redirects\_logout\_post: ให้ user login เข้ามาก่อน แล้วทำการ check response reverse == 'login'

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test course.tests.custom_logout
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.453s

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

```
def register(request):
    if request.method == 'POST':
        return redirect(reverse('course_list'))
    return redirect(reverse('mycourse'))
```

```
class RegisterViewTest(TestCase):
    def setUp(self) -> None:
        self.user = User.objects.create_user(username='6410685249', password='first123456789')
        self.client.login(username='6410685249', password='first123456789')
        self.user_info = student_info.objects.create(student_username='6410685249', student_password='first123456789',
                                                    student_surname="Siriphatson", student_lastname="sringamphong")

        self.cn202 = Course(course_id="CN202", course_name="Algorithm 2", course_term="1", course_year="2023",
                            available_seat=10, course_is_open=True)
        self.cn203 = Course(course_id="CN203", course_name="Algorithm 3", course_term="1", course_year="2023",
                            available_seat=10, course_is_open=True)
        self.cn202.save()
        self.cn203.save()
        self.temp = class_of_students.objects.create(student_id="6410685249", student_name="Siriphatson", student_lastname="sringamphong")
        self.temp.course.add("CN202")
    def test_redirect_mycourse_status(self):
        response = self.client.get(reverse('register'))
        self.assertEqual(response.status_code, 302)
        self.assertRedirects(response, reverse('mycourse'))
    def test_redirect_course_list_status(self):
        response = self.client.post(reverse('register'))
        self.assertEqual(response.status_code, 302)
        self.assertRedirects(response, reverse('course_list'))
```

setup: set user และทำการ login, set student information, set course CN202 and CN203, set student link กัน

Course CN202

Function test\_redirect\_mycourse\_status: check status code == 302 และ check response reverse ==

'mycourse'

Function test\_redirect\_course\_list\_status: check status code == 302 และ check response reverse ==

'course\_list'

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test course.tests.RegisterViewTest
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.889s

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

```

@login_required
def add_to_DB(request):
    if request.method == 'POST':
        try:
            student_info_obj = student_info.objects.get(student_username=request.user)
        except student_info.DoesNotExist:
            return HttpResponse("Student information not found for this user.")

        course_id = request.POST.get('course_id')
        course = Course.objects.get(course_id=course_id)

        # Create a new Student instance
        student_instance = class_of_students.objects.create(
            student_id=str(student_info_obj.student_username),
            student_name=str(student_info_obj.student_surname),
            student_lastname=str(student_info_obj.student_lastname),
        )
        student_instance.course.add(course_id)
        student_instance.save()
        course = Course.objects.get(course_id=course_id)
        course.available_seat -= 1
        course.save()
        # Redirect to the course list page
        return redirect(reverse('course_list'))

    return HttpResponseRedirect(reverse('course_list'))

```

```

class Add_to_dbViewTest(TestCase):

    def setUp(self):
        self.cn202 = Course(course_id="CN202", course_name="Algorithm 2", course_term="1",
                           course_year="2023", available_seat=10, course_is_open=True)
        self.cn203 = Course(course_id="CN203", course_name="Algorithm 3", course_term="1",
                           course_year="2023", available_seat=10, course_is_open=True)
        self.cn202.save()
        self.cn203.save()
        user_info = student_info.objects.create(student_username="6410685249", student_password="first123456789",
                                                student_surname="Siriphatson", student_lastname="sringamphong")
        self.temp = class_of_students.objects.create(student_id="6410685249",
                                                    student_name="Siriphatson", student_lastname="sringamphong")
        self.temp.course.add("CN202")

    def test_redirect_course_list(self):
        user = User.objects.create_user(username='testuser', password='testpassword')
        self.client.login(username='testuser', password='testpassword')
        response = self.client.get(reverse('add'))
        self.assertRedirects(response, reverse('course_list'))
        self.assertEqual(response.status_code, 302)

    def test_student_info_not_exists(self):
        user = User.objects.create_user(username='6418522350', password='xxx123456789')
        self.client.login(username='6418522350', password='xxx123456789')
        response = self.client.post(reverse('add'))
        self.assertContains(response, "Student information not found for this user.")
        self.assertEqual(response.status_code, 200)
        self.assertEqual(Course.objects.get(course_id='CN202').available_seat, 10)

```

```

def test_student_reverse_course_list(self):
    user = User.objects.create_user(username='6410685249', password='first123456789')
    self.client.login(username='6410685249', password='first123456789')
    response = self.client.post(reverse('add'), {'course_id': 'CN202'})

    self.assertRedirects(response, reverse('course_list'))
    self.assertEqual(response.status_code, 302)
    self.assertEqual(Course.objects.get(course_id='CN202').available_seat, 9)

```

setup: set user และทำการ login, set student information, set course CN202 and CN203, set student link กัน

Course CN202

Function test\_redirect\_course\_list: check reverse == 'course\_list' และ check status code == 302

Function test\_student\_info\_not\_exists: check status code == 200 check response message == 'Student information not found for this user.', check available\_seat หลังจากที่ไม่มื่อนักศึกษาอยู่ใน student info models

Functions test\_student\_reverse\_course\_list: ทำการเช็คถ้าเกิดมีชื่อนักศึกษาใน student info Model ถ้าทำการลบ แล้วจำนวนเก้าอี้ที่ลงได้จะลดลงมัย ซึ่งในนี้จำกัทดสอบ test response reverse == 'course\_list', test status code == 302, test available\_seat == 9

ผลลัพธ์

```
$ python manage.py test course.tests.Add_to_dbViewTest
Found 3 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
...
-----
Ran 3 tests in 1.362s

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



```
def get_in_my_course(request):
    if request.method == 'POST':
        return redirect(reverse('mycourse'))
    return HttpResponseRedirect(reverse('course_list'))
```

```
class GetMyCourseTest(TestCase):
    def test_HttpResponseRedirect(self):
        reponse = self.client.get(reverse('get_in_my_course'))
        self.assertRedirects(reponse, reverse('course_list'))
        self.assertEqual(reponse.status_code, 302)

    def test_redirect_my_course(self):
        self.user = User.objects.create_user(username='6410685249', password='first123456789')
        self.client.login(username='6410685249', password='first123456789')
        self.user_info = student_info.objects.create(student_username='6410685249', student_password='first123456789',
                                                    , student_surname="Siriphatson", student_lastname="sringamphong")

        reponse = self.client.post(reverse('get_in_my_course'))
        self.assertRedirects(reponse, reverse('mycourse'))
        self.assertEqual(reponse.status_code, 302)
```

Function test\_HttpResponseRedirect: test status code 302, test response reverse == 'course\_list'

Function test\_redirect\_my\_course: test status code, test response reverse == 'mycourse'

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test course.tests.GetMyCourseTest
Found 2 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.458s

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

```
def delete_subject(request):
    if request.method == 'POST':

        class_of_students.objects.get(course=request.POST['course_id'][:5]).delete()
        course = Course.objects.get(course_id=request.POST['course_id'][:5])
        course.available_seat += 1
        course.save()
        return redirect(reverse('mycourse'))
    return redirect(reverse('mycourse'))
```

```
class DeleteCourseTest(TestCase):
    def setUp(self):
        self.cn202 = Course(course_id="CN202",course_name="Algorithm 2",course_term="1",course_year="2023",
                             avilable_seat=10,course_is_open=True)
        self.cn203 = Course(course_id="CN203",course_name="Algorithm 3",course_term="1",course_year="2023",
                             avilable_seat=0,course_is_open=True)
        self.cn202.save()
        self.cn203.save()
        self.temp = class_of_students.objects.create(student_id="6410685249", student_name="Siriphatson", student_lastname="sringamphong")
        self.temp.course.add("CN202")
        self.temp2 = class_of_students.objects.create(student_id="6410685249", student_name="Siriphatson", student_lastname="sringamphong")
        self.temp2.course.add("CN203")

    def test_redirect_mycourse(self):
        user = User.objects.create_user(username='6410685249', password='first123456789')
        self.client.login(username='6410685249', password='first123456789')
        user_info = student_info.objects.create(student_username='6410685249',student_password='first123456789',
                                                student_surname="Siriphatson",student_lasname="sringamphong")
        reponse = self.client.get(reverse('delete_subject'))
        self.assertRedirects(reponse,reverse('mycourse'))
        self.assertEqual(reponse.status_code,302)
    def test_post_redirect_mycourse(self):
        user = User.objects.create_user(username='6410685249', password='first123456789')
        self.client.login(username='6410685249', password='first123456789')
        user_info = student_info.objects.create(student_username='6410685249',student_password='first123456789',
                                                student_surname="Siriphatson",student_lasname="sringamphong")
        reponse = self.client.post(reverse('delete_subject'),{'course_id':'CN202'})
        self.assertRedirects(reponse,reverse('mycourse'))
        self.assertEqual(reponse.status_code,302)
```

```
def test_course_not_open(self):
    cn201 = Course(course_id="CN201",course_name="java",course_term="1",course_year="2023",avilable_seat=10,course_is_open=False)
    self.assertEqual(cn201.course_is_open,False)

def test_string_course(self):
    cn201 = Course(course_id="CN201",course_name="java",course_term="1",course_year="2023",avilable_seat=10,course_is_open=False)
    self.assertEqual(str(cn201),'CN201 java 1/2023 False')

def test_class_of_student(self):
    self.assertEqual(str(self.temp),'6410685249 Siriphatson sringamphong')

def test_avliable_set(self):
    user = User.objects.create_user(username='6410685249', password='first123456789')
    self.client.login(username='6410685249', password='first123456789')
    user_info = student_info.objects.create(student_username='6410685249',student_password='first123456789',
                                            student_surname="Siriphatson",student_lasname="sringamphong")

    self.assertEqual(self.cn203.avilable_seat,0)
def test_admin(self):
    course_admin = CourseAdmin(Course, None)
    students = course_admin.get_students(self.cn202)
    self.assertListEqual(students,['Siriphatson'])

def test_student(self):
    user_info = student_info.objects.create(student_username='6410685249',student_password='first123456789',
                                            student_surname="Siriphatson",student_lasname="sringamphong")
    self.assertEqual(str(user_info),'Siriphatson sringamphong: 6410685249 first123456789')
```

setUp add course CN202,CN203 หลังจากนั้น add student to that course

Function test\_redirect\_mycourse: test redirect = 'mycourse', test status code== 302

Function test\_post\_redirect\_mycourse: test redirect = 'mycourse', test status code== 302

Function test\_course\_not\_open: test course CN201 is open ==False

Function test\_string\_course: Test string Course model

Function test\_available\_seat: Test seat of CN203 == 0

Function test\_admin: test List of student

Function test\_student: test string student info == 'Siriphaatson sringamphong: 6410685249  
first123456789'

```
USER@tu ~/Documents/work/jack/cn331/register (testing)
$ python manage.py test course.tests.DeleteCourseTest
Found 8 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 8 tests in 1.312s

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