

## Test Driven Development (TDD)

“Product Construction or Implementation Phase”

**Tools:** Visual Studio Code

**Framework:** Python-Django

**Required Installations:**

**For Testing Python Core Code:** pip install pytest

**For Testing Django-Framework:** pip install pytest-django

**For Generating Test Report:** pip install pytest-cov

**For helping to generate instances of Django models:** pip install mixer

### Flows of Process (FOP)

1. In Root Project Repo “SQA\_SMS”: creating a file test\_setting.py

test\_setting.py

```
CSE-455_SQA > SQA_SMS > test_settings.py > ...
1 from .settings import *
2
3 DATABASES = {
4     "default": {
5         "ENGINE": "django.db.backends.sqlite3",
6         "NAME": ":memory",
7     }
8 }
9
10 EMAIL_BACKEND = 'django.core.mail.backends.locmem.EmailBackend'
```

2. In Root Project Repo “SQA\_SMS”: creating a file pytest.ini

pytest.ini

```
CSE-455_SQA > pytest.ini
1 [pytest]
2 DJANGO_SETTINGS_MODULE = SQA_SMS.test_settings
3 addopts = --nomigrations --cov=. --cov-report=html
```

3. In inner root “SQA\_SMS” repo: creating a folder “tests”

in “tests” folder create 2 files “\_\_init\_\_.py” and “test\_models.py”

test\_models.py

CSE-455\_SQA > base > test > test\_models.py > ...

```
1 from django.test import TestCase
2 from base.models import Department, Teacher
3 from django.test import TestCase
4 from base.models import Student
5 from django.utils import timezone
6 from base.models import AdminNotice
7
```

```
class DepartmentModelTest(TestCase):
    """
    Test cases for the Department model.
    """

    @classmethod
    def setUpTestData(cls):
        """
        Set up test data for Department model tests.
        """

        # Create a department object for testing
        cls.department = Department.objects.create(
            name="Marketing",
            location="Block B",
            rank=2,
            phone="555-1234",
```

Models.py

CSE-455\_SQA > base > models.py > ...

```
1 from django.db import models
2 from django.core.validators import MaxValueValidator, MinValueValidator
3
```

```

class Student(models.Model):

    STATUS=(
        ('Regular','Regular'),
        ('Ex-Student','Ex-Student'),
    )

    #user = models.OneToOneField(User,null=True,blank=True, on_delete=models.CASCADE)
    name=models.CharField(max_length=30,null=False)
    hsc_roll=roll=models.CharField(max_length=10,null=True,unique=True)
    hsc_reg=models.CharField(max_length=16,null=True,unique=True)
    reg_no=models.CharField(max_length=16,null=True,unique=True,blank=True)
    roll=models.CharField(max_length=10,null=True,unique=True,blank=True)
    #dept=models.ForeignKey(Department,null=True,on_delete=models.SET_NULL,blank=True)
    session=models.CharField(max_length=10,null=True)
    email=models.EmailField()

```

4. Write a command in terminal for testing models.py : `pytest`

5. Checking Coverage Report For Models. After opening index.html from htmlcov folder

Coverage for **base\models.py**: 100%

60 statements   60 run   0 missing   0 excluded

« prev   ^ index   » next   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

Coverage for **base\test\test\_models.py**: 98%

57 statements   56 run   1 missing   0 excluded

« prev   ^ index   » next   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

## 6. test\_view.py

```
CSE-455_SQA > base > test > test_views.py > TestLogoutAdmin > test_logout_user_view
1  from django.test import TestCase, Client, RequestFactory
2  from django.urls import reverse
3  from django.http import HttpResponse
4  from base.forms import TeacherForm # Import your TeacherForm
5  from base.models import Teacher # Import your Teacher model
6  from django.contrib.auth.models import User
7  from base.models import Student
8  from django.http import HttpRequest
9  from base.forms import StudentForm
10 from django.core.exceptions import ObjectDoesNotExist
11 from django.http.response import HttpResponseRedirectBase
12 from django.http import HttpResponseRedirect
13 from base.views import create_student
14 from base.forms import AdminNoticeForm # Make sure to import your form
15 from base.models import AdminNotice
16 from base.views import update_teacher
17

class DeleteStudentTest(TestCase):
    """
    Test case for the delete_student.

    Inherits:
    - TestCase: Django's built-in test case class.

    Attributes:
    - student1: Sample instance of the Student model for testing.
    """
```

## views.py

```
from django.contrib.auth.decorators import user_passes_test
from django.contrib.auth import authenticate, login, logout
from .forms import StudentForm

def common_page(request):
    """
    View for rendering the common page.
    """
```

7. Then run a command in terminal : `py.test`

8. Checking Coverage Report For Views. After opening `index.html` from `htmlcov` folder

Coverage for **base\test\test\_views.py**: 97%

327 statements   317 run   10 missing   0 excluded

[« prev](#)   [^ index](#)   [» next](#)   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

Coverage for **base\views.py**: 93%

153 statements   143 run   10 missing   0 excluded

[« prev](#)   [^ index](#)   [» next](#)   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

9. `test_forms.py`

```
CSE-455_SQA > base > test > test_forms.py > TestForms > test_create_user_form_valid_data
1  from django.test import TestCase
2  from base.forms import DepartmentForm, TeacherForm
3  from base.models import Department, Teacher
4  from base.forms import StudentForm
5  from base.forms import CreateUserForm, AdminNoticeForm
6  from django.contrib.auth.models import User
7
8
9
10 class TestForms(TestCase):
11     """
12     Test case for form validation.
13
```

## Forms.py

```
CSE-455_SQA > base > forms.py > ...
1  from django.forms import ModelForm
2  from .models import *
3  from django.forms.widgets import Widget
4  from django.contrib.auth.forms import UserCreationForm
5  from django import forms
6  from django.contrib.auth.models import User
7
8  '''
9  CreateUserForm
10  =====
11
12  .. autoclass:: yourapp.forms.CreateUserForm
13     :members:
14     :undoc-members:
15     :show-inheritance:
```

10. Then run a command in terminal : py.test

11. Checking Coverage Report For forms. After opening index.html from htmlcov folder

Coverage for **base\test\test\_forms.py**: 100%

74 statements   74 run   0 missing   0 excluded

[« prev](#)   [^ index](#)   [» next](#)   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

Coverage for **base\forms.py**: 100%

27 statements   27 run   0 missing   0 excluded

[« prev](#)   [^ index](#)   [» next](#)   coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

## 12. test\_urls.py

```
from django.test import TestCase
from django.urls import reverse
from django.contrib.auth.models import User
from base.models import AdminNotice # Replace 'yourapp' with the actual name of your

class TestUrls(SimpleTestCase):

    def test_create_teacher_url_resolves(self):
        url = reverse('create_teacher')
```

## Urls.py

```
from django.urls import path
from . import views

""".. function:: path(route, view, kwargs=None, name=None)

    Defines a URL pattern to match the given route and view function.
```

13. Then run a command in terminal : `py.test`

14. Checking Coverage Report For urls. After opening index.html from htmlcov folder

Coverage for **base\test\test\_urls.py**: 100%

60 statements

60 run

0 missing

0 excluded

« prev ^ index » next coverage.py v7.4.2, created at 2024-02-25 01:28 +0600

## 15. filters.py

```
CSE-455_SQA > base > filters.py > ...
1  import django_filters
2  from django_filters import CharFilter, NumberFilter
3  from .models import *
4
5  from django import forms
6
7
8
9
10 class StudentFilter(django_filters.FilterSet):
11     """
12     Filter class for the Student model.
13
```

## Test\_filters.py

```
CSE-455_SQA > base > test > test_filters.py > StudentFilterTest > test_student_filter
1  from django.test import TestCase
2  from base.models import Student
3  from base.filters import StudentFilter
4
5
6
7  class StudentFilterTest(TestCase):
8      """
9      Test case for the StudentFilter class.
10
```

16. Then run a command in terminal : `py.test`

17. Checking Coverage Report For filters. After opening index.html from htmlcov folder



## Conclusion:

Total Coverage Rate: 97%

Name	Stmts	Miss	Cover
-----			
SQA_SMS\__init__.py	0	0	100%
SQA_SMS\test_settings.py	3	0	100%
base\__init__.py	0	0	100%
base\admin.py	4	0	100%
base\filters.py	13	0	100%
base\forms.py	27	0	100%
base\models.py	60	1	98%
base\test\__init__.py	0	0	100%
base\test\test_filters.py	15	0	100%
base\test\test_forms.py	74	0	100%
base\test\test_models.py	57	1	98%
base\test\test_urls.py	60	0	100%
base\test\test_views.py	327	10	97%
base\views.py	153	10	93%
-----			
TOTAL	793	22	97%