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 > ...

from .settings import *

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.sqlite3",
        "NAME": ":memory",
    }
}

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 > F 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

```
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
```

Models.py

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

- 4. Write a command in terminal for testing models.py: py.test
- 5. Checking Coverage Report For Models. After opening index.html from htmlcov folder



6. test_view.py

```
CSE-455_SQA > base > test > 🏓 test_views.py > ધ TestLogoutAdmin > ᠪ 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
class DeleteStudentTest(TestCase):
```

```
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



9. test_forms.py

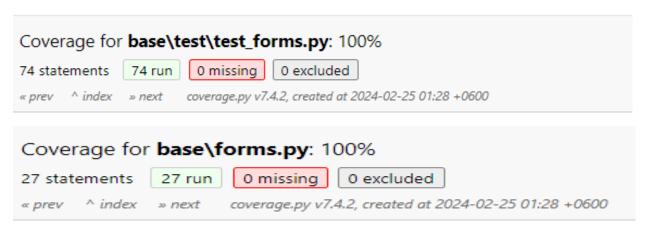
```
CSE-455_SOA > base > test > test_forms.py > test_create_user_form_valid_data

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
     from .models import *
     from django.forms.widgets import Widget
     from django.contrib.auth.forms import UserCreationForm
      from django import forms
     from django.contrib.auth.models import User
      CreateUserForm
  10
  11
      .. autoclass:: yourapp.forms.CreateUserForm
  12
  13
         :members:
         :undoc-members:
  14
         :show-inheritance:
  15
```

- 10. Then run a command in terminal : py.test
- 11. Checking Coverage Report For forms. After opening index.html from htmlcov folder



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')
```

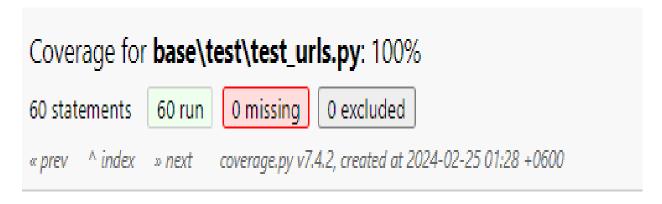
Urls.py

```
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



15. filters.py

```
Import django_filters
from django_filters import CharFilter, NumberFilter
from .models import *

from django import forms

class StudentFilter(django_filters.FilterSet):

"""

Filter class for the Student model.
```

Test filters.py

```
CSE-455_SQA > base > test > test_filters.py > test_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.
```

- 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_initpy	0	0	100%
SQA_SMS\test_settings.py	3	8	100%
base\initpy	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\initpy	0	0	100%
<pre>base\test\test_filters.py</pre>	15	0	100%
<pre>base\test\test_forms.py</pre>	74	0	100%
<pre>base\test\test_models.py</pre>	57	1	98%
base\test\test_urls.py	60	0	100%
<pre>base\test\test_views.py</pre>	327	10	97%
base\views.py	153	10	93%
TOTAL	793	22	97%