TDD(Test Driven Development)

Test Driven Development is an approach in which we build a test first, then fail the test and finally refactor our code to pass the test.

As the name suggests, we should first add the test before adding the functionality in our code. Now our target is to make the test pass by adding new code to our program. So we refactor our code to pass the written test. This uses the following process –

- Write a failing unit test
- Make the unit test pass
- Repeat

TDD promotes a design-first approach to development. By thinking through requirements and writing corresponding tests beforehand, you naturally gravitate toward more modular, maintainable, and loosely coupled code. This approach reduces the likelihood of writing monolithic and tightly integrated components.

Test Driven Development (TDD) with Pytest

Here we exploring the Test-driven development approach with Python. Python official interpreter comes with the unit test module. While the unittest library is feature-rich and effective at its task, we'll be using **pytest** as our weapon.

Pytest is one of the most popular frameworks for testing python code. There are many advantages that make people enjoy using Pytest. Definitely, one of them is the ease of use, among extra plugins and very well-written documentation. Pytest supports unit tests and allows you to write simple scalable test sets. Moreover, Pytest lets programmers use fixtures, parametrization, and gives an opportunity to skip selected tests during execution.

Getting Started with Pytest

- Installation of pyTest:

```
pip install pytest
```

```
pip install pytest-django
```

- In root directory (where manage.py is located), create a file named pytest.ini.

pytest.ini
```
[pytest]
DJANGO_SETTINGS_MODULE = HospitalManagement.settings
```

- Create a tests folder in the app folder.

```
(base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> mkdir .\App_DischargePatient\tests


    Directory: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement\App_DischargePatient


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/22/2024   9:16 PM                tests
```

- Now in tests directory, create a file named *test_models.py* to write tests for models:

```
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> pytest
============================= test session starts =============================
platform win32 -- Python 3.11.5, pytest-8.0.1, pluggy-1.4.0
django: version: 5.0.2, settings: HospitalManagement.settings (from ini)
rootdir: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement
configfile: pytest.ini
plugins: django-4.8.0
collected 1 item

App_DischargePatient\tests\test_models.py E                              [100%]

================================== ERRORS ==================================
..\my_env\Lib\site-packages\django\db\backends\sqlite3\base.py:329: OperationalError
=========================== short test summary info ===========================
ERROR App_DischargePatient/tests/test_models.py::test_create_patient - django.db.utils.OperationalError: no such column: App_DischargePatient_patient.i
d
=============================== 1 error in 1.10s ===============================
```

- Now create necessary models for Patient in *models.py*:

```
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> pytest
============================= test session starts =============================
platform win32 -- Python 3.11.5, pytest-8.0.1, pluggy-1.4.0
django: version: 5.0.2, settings: HospitalManagement.settings (from ini)
rootdir: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement
configfile: pytest.ini
plugins: django-4.8.0
collected 1 item

App_DischargePatient\tests\test_models.py .                              [100%]

============================== 1 passed in 0.45s ==============================
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement>
```

- Now in tests directory, create a file named *test_admin.py* to write tests for admin:

```
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> pytest
=============================================================== test session starts ===============================================================
platform win32 -- Python 3.11.5, pytest-8.0.1, pluggy-1.4.0
django: version: 5.0.2, settings: HospitalManagement.settings (from ini)
rootdir: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement
configfile: pytest.ini
plugins: django-4.8.0
collected 2 items / 1 error

===================================================================== ERRORS ======================================================================
_____________________________________________ ERROR collecting App_DischargePatient/tests/test_admin.py ____________________________________________
ImportError while importing test module 'C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement\App_DischargePatient\tests\test_adm
in.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
C:\Users\jucse\anaconda3\Lib\importlib\__init__.py:126: in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
App_DischargePatient\tests\test_admin.py:5: in <module>
    from ..models import Patient
E   ImportError: attempted relative import with no known parent package
============================================================= short test summary info =============================================================
ERROR App_DischargePatient/tests/test_admin.py
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
================================================================= 1 error in 0.49s ================================================================
```

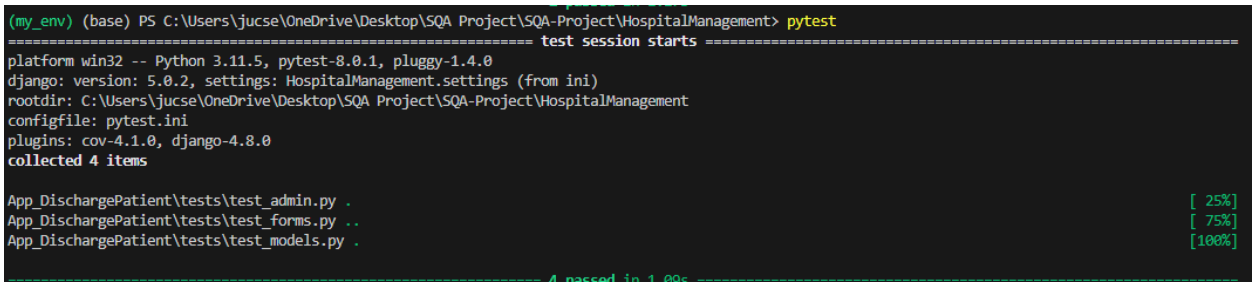
- Now create necessary models for Patient in *admin.py*:

```
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> pytest
=============================================================== test session starts ===============================================================
platform win32 -- Python 3.11.5, pytest-8.0.1, pluggy-1.4.0
django: version: 5.0.2, settings: HospitalManagement.settings (from ini)
rootdir: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement
configfile: pytest.ini
plugins: cov-4.1.0, django-4.8.0
collected 2 items

App_DischargePatient\tests\test_admin.py .                                                                                                   [ 50%]
App_DischargePatient\tests\test_models.py .                                                                                                  [100%]

================================================================ 2 passed in 1.17s ================================================================
```

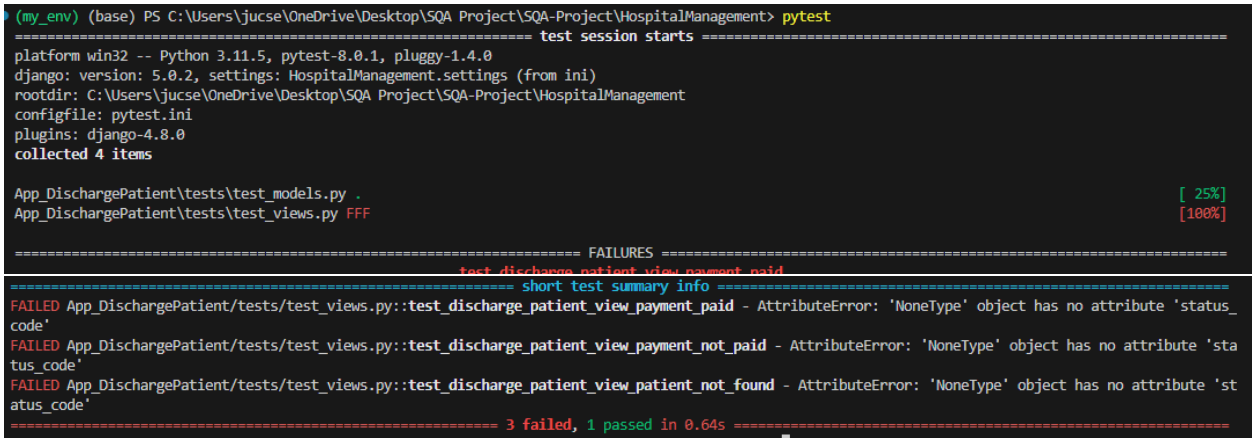
- Now in tests directory, create a file named *test_forms.py* to write tests for form:

```
(my_env) (base) PS C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement> pytest
=============================================================== test session starts ===============================================================
platform win32 -- Python 3.11.5, pytest-8.0.1, pluggy-1.4.0
django: version: 5.0.2, settings: HospitalManagement.settings (from ini)
rootdir: C:\Users\jucse\OneDrive\Desktop\SQA Project\SQA-Project\HospitalManagement
configfile: pytest.ini
plugins: cov-4.1.0, django-4.8.0
collected 2 items / 1 error

============================================================= short test summary info =============================================================
ERROR App_DischargePatient/tests/test_forms.py
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
================================================================= 1 error in 0.46s ================================================================
```

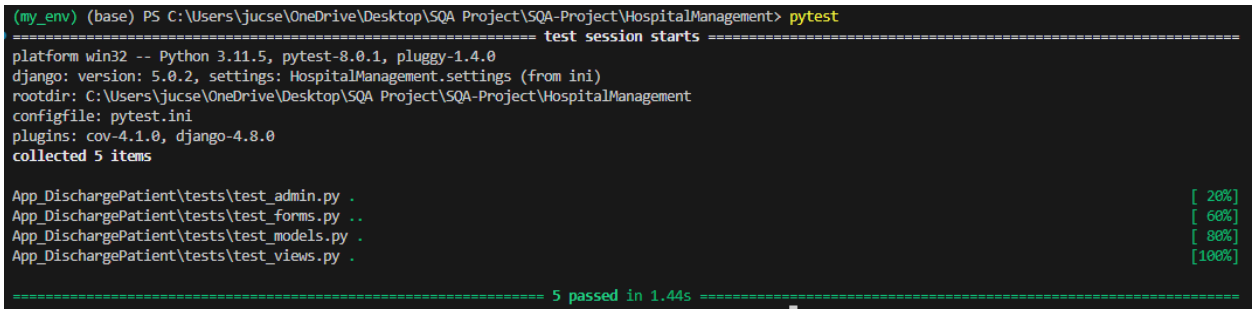
- Now create necessary models for Patient in *forms.py*:



- Now in tests directory, create a file named *test_views.py* to write tests for views:



- Now create necessary models for Patient in *views.py*:



Fix the error for this failed test case if any found and run again.

To Generate Coverage Report:

- Write the command in terminal:

```
pip install pytest-cov
```

```
pytest --cov=HospitalManagement
```

```
App_DischargePatient\tests\test_admin.py .
App_DischargePatient\tests\test_forms.py ..
App_DischargePatient\tests\test_models.py .
App_DischargePatient\tests\test_views.py .

---------- coverage: platform win32, python 3.11.5-final-0 -----------
Name                                Stmts   Miss  Cover
-------------------------------------------------------
HospitalManagement\__init__.py          0      0   100%
HospitalManagement\asgi.py              4      4     0%
HospitalManagement\settings.py         21      0   100%
HospitalManagement\urls.py              4      0   100%
HospitalManagement\views.py             3      1    67%
HospitalManagement\wsgi.py              4      4     0%
-------------------------------------------------------
TOTAL                                  36      9    75%
```

References:

1. https://www.xenonstack.com/blog/test-driven-development-python#:~:text=TDD%20is%20nothing%20but%20the,write%20a%20test%20for%20that.
2. https://stackabuse.com/test-driven-development-with-pytest/