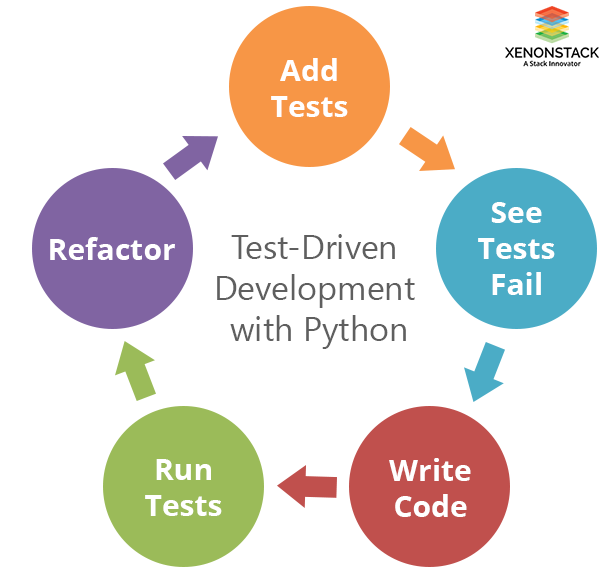
|  |
| --- |
| **TDD(Test Driven Development)** |

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.

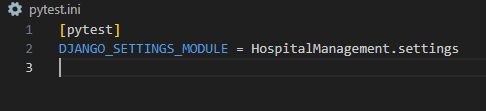
|  |
| --- |
| **Getting Started with Pytest** |

* Installation of pyTest:

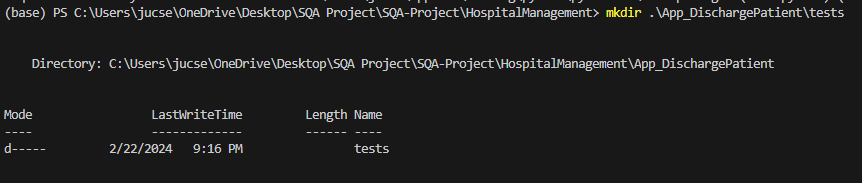




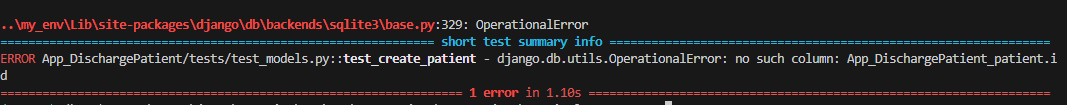
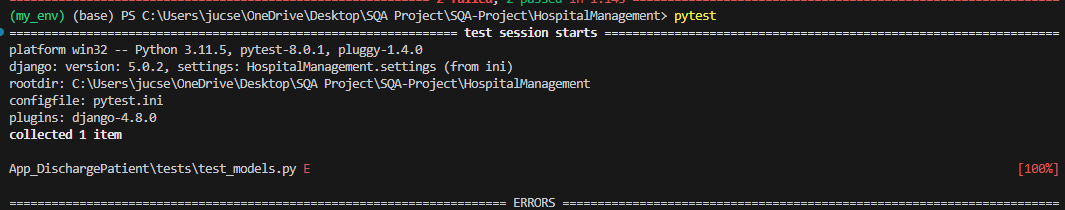
* In root directory (where manage.py is located), create a file named pytest.ini.



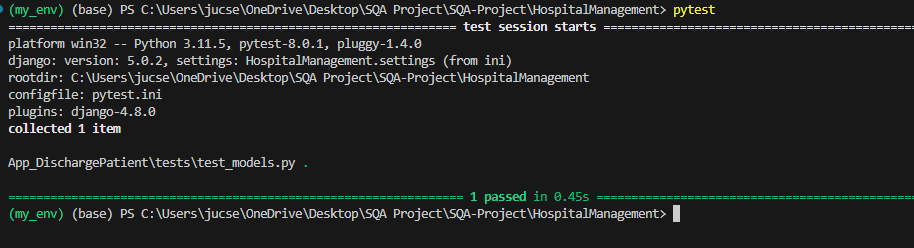
* Create a tests folder in the app folder.



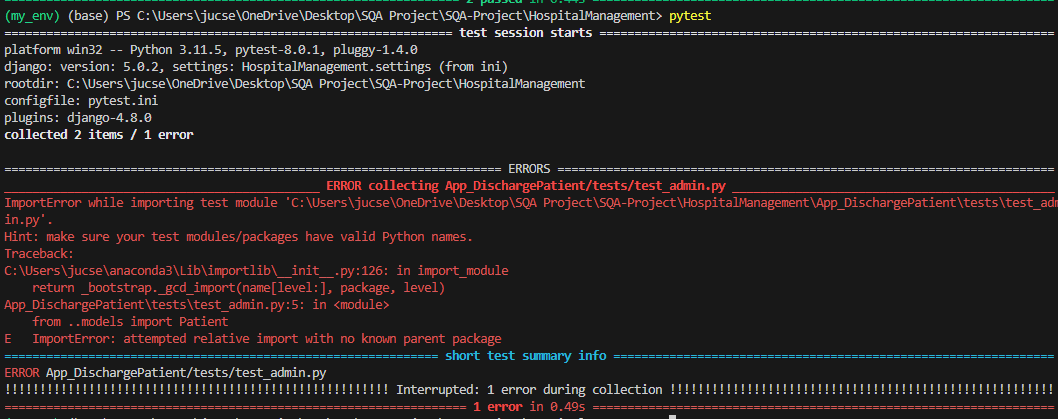
* Now in tests directory, create a file named *test\_models.py* to write tests for models:



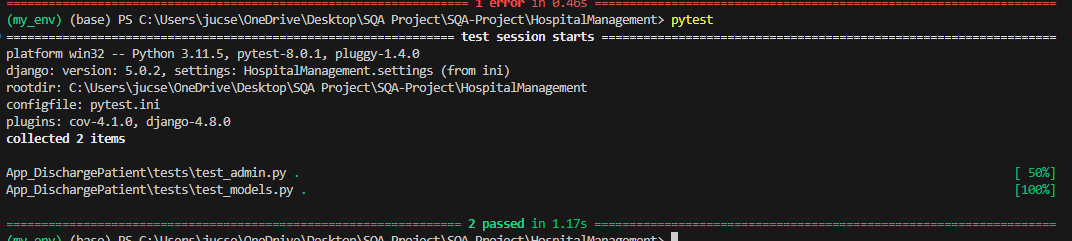
* Now create necessary models for Patient in *models.py*:



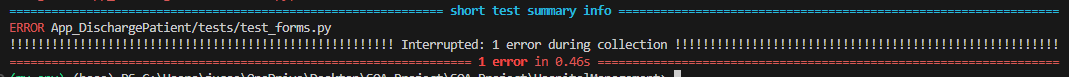
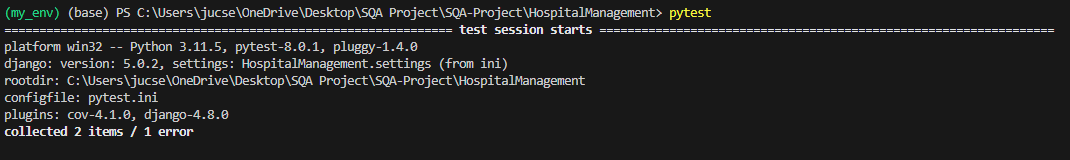
* Now in tests directory, create a file named *test\_admin.py* to write tests for admin:



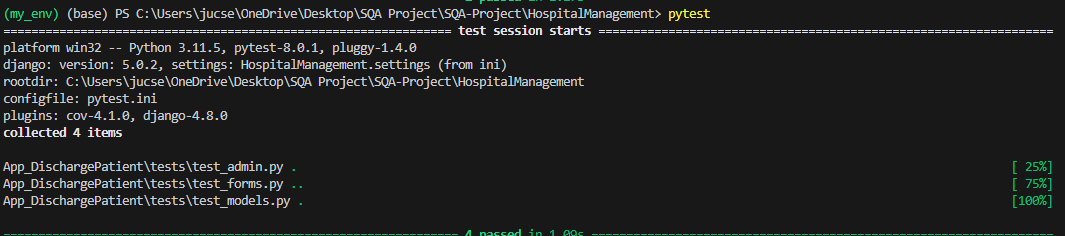
* Now create necessary models for Patient in *admin.py*:



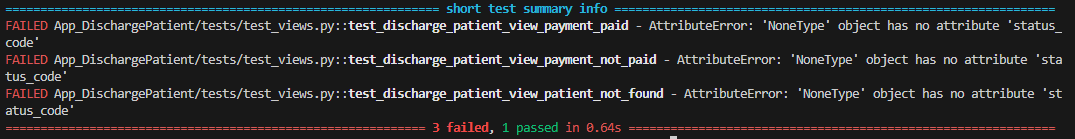
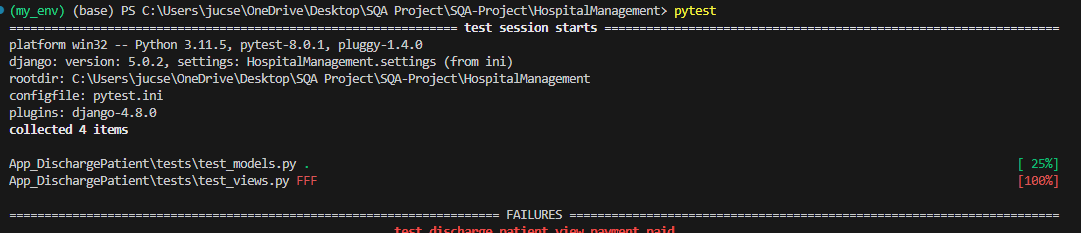
* Now in tests directory, create a file named *test\_forms.py* to write tests for form:



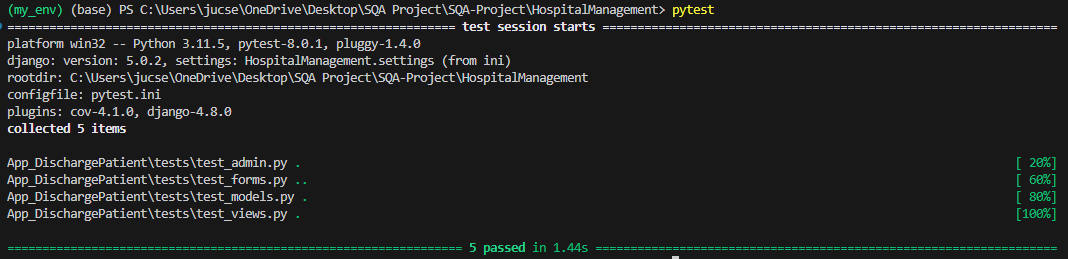
* Now create necessary models for Patient in *forms.py*:



* 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*:



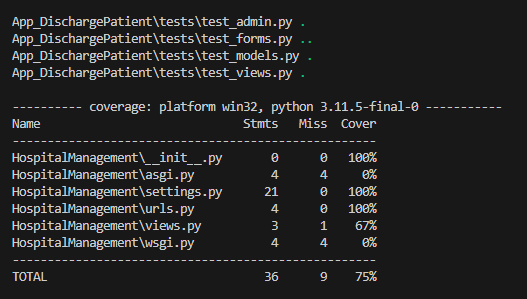
Fix the error for this failed test case if any found and run again.

|  |
| --- |
| **To Generate Coverage Report:** |

* Write the command in terminal:







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/>