SOFTWARE TESTING

Group 12

Akhil Balineni AP19110010162

Chaitanya Nitin Suda AP19110010195

Manish Valeti AP19110010204

Sai Suprabhanu Nallapaneni AP19110010232

Testing in Django Framework:

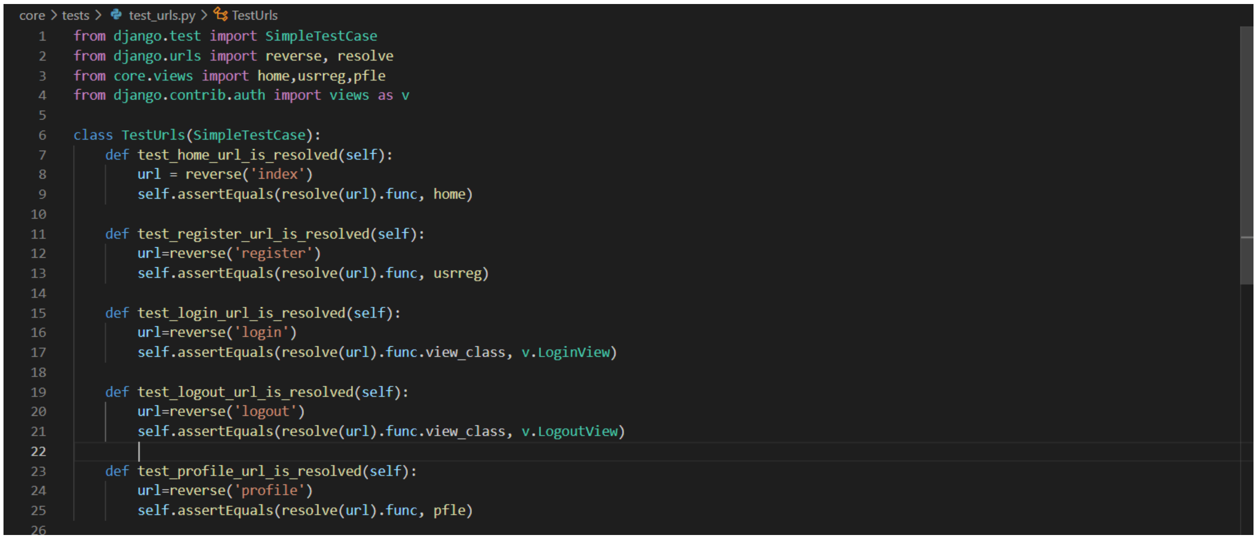
Django provides many useful developer utilities, testing is one of them. We can test the URLs, Models, Views, Forms, and Functionalities of our web application using these testing utilities.

To begin with, we have to create a tests folder in the required Django app. Inside the folder, we have to create *__init__.py*, *test_urls.py*, *test_models.py*, *test_views.py,* and *test_forms.py*. Each of these files corresponds to the test set of each component respectively.

With the help of built-in functions like *assertEquals()* and *assertTemplateUsed()*, we can easily understand and analyze the behavior of various components of the application. This is how we can easily implement both unit and integration tests with the help of Django's built-in modules.

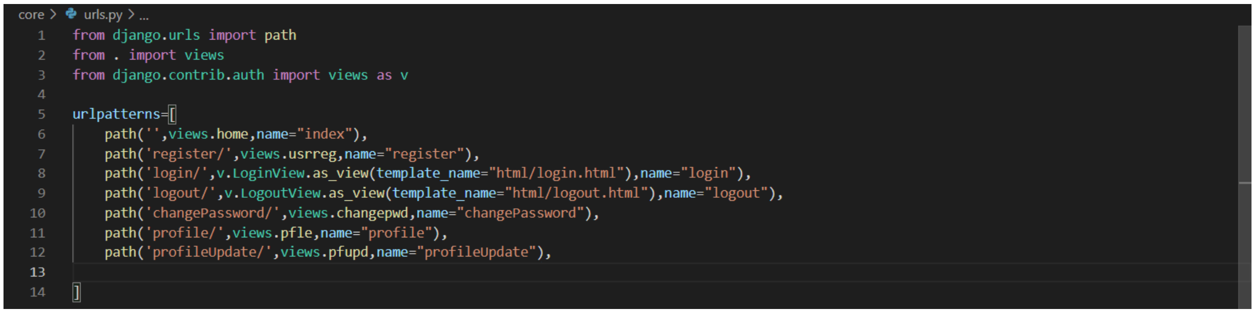
URLs Testing:

test_urls.py



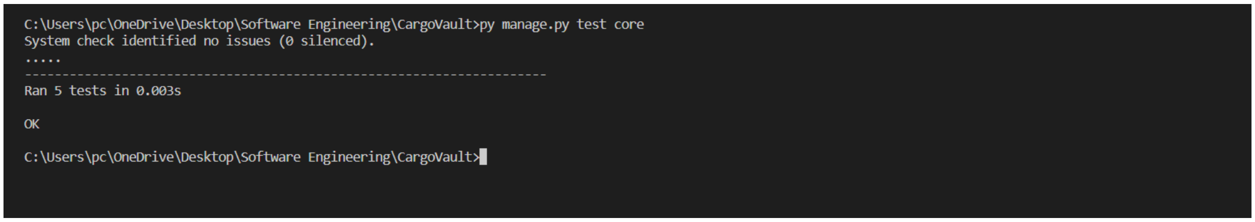
```python
from django.test import SimpleTestCase
from django.urls import reverse, resolve
from core.views import home,usrreg,pfle
from django.contrib.auth import views as v

class TestUrls(SimpleTestCase):
    def test_home_url_is_resolved(self):
        url = reverse('index')
        self.assertEquals(resolve(url).func, home)

    def test_register_url_is_resolved(self):
        url=reverse('register')
        self.assertEquals(resolve(url).func, usrreg)

    def test_login_url_is_resolved(self):
        url=reverse('login')
        self.assertEquals(resolve(url).func.view_class, v.LoginView)

    def test_logout_url_is_resolved(self):
        url=reverse('logout')
        self.assertEquals(resolve(url).func.view_class, v.LogoutView)

    def test_profile_url_is_resolved(self):
        url=reverse('profile')
        self.assertEquals(resolve(url).func, pfle)
```

urls.py



```python
from django.urls import path
from . import views
from django.contrib.auth import views as v

urlpatterns=[
    path('',views.home,name="index"),
    path('register/',views.usrreg,name="register"),
    path('login/',v.LoginView.as_view(template_name="html/login.html"),name="login"),
    path('logout/',v.LogoutView.as_view(template_name="html/logout.html"),name="logout"),
    path('changePassword/',views.changepwd,name="changePassword"),
    path('profile/',views.pfle,name="profile"),
    path('profileUpdate/',views.pfupd,name="profileUpdate"),

]
```

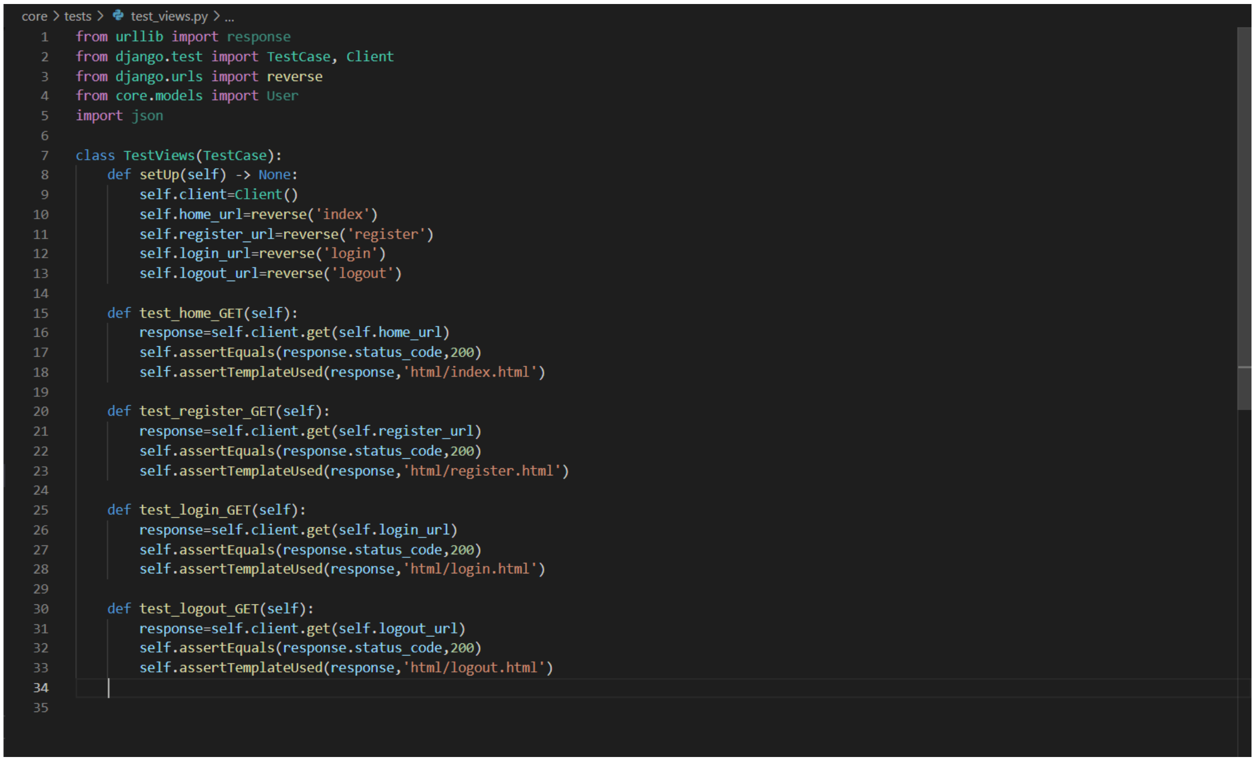
URLs test output



```
C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>py manage.py test core
System check identified no issues (0 silenced).
.....
----------------------------------------------------------------------
Ran 5 tests in 0.003s

OK

C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>
```

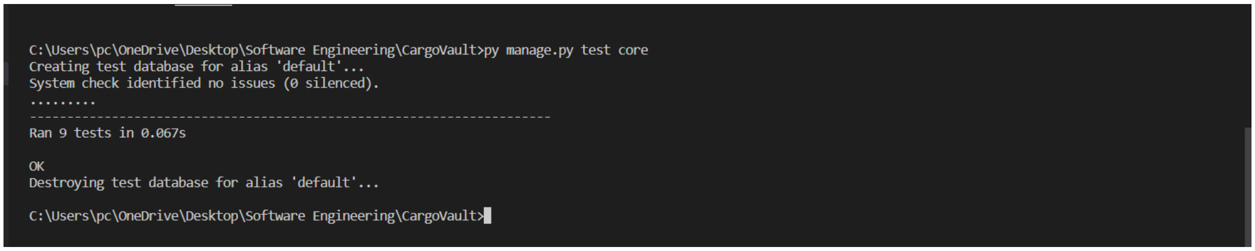
Views Testing:

test_views.py



```python
from urllib import response
from django.test import TestCase, Client
from django.urls import reverse
from core.models import User
import json

class TestViews(TestCase):
    def setUp(self) -> None:
        self.client=Client()
        self.home_url=reverse('index')
        self.register_url=reverse('register')
        self.login_url=reverse('login')
        self.logout_url=reverse('logout')

    def test_home_GET(self):
        response=self.client.get(self.home_url)
        self.assertEquals(response.status_code,200)
        self.assertTemplateUsed(response,'html/index.html')

    def test_register_GET(self):
        response=self.client.get(self.register_url)
        self.assertEquals(response.status_code,200)
        self.assertTemplateUsed(response,'html/register.html')

    def test_login_GET(self):
        response=self.client.get(self.login_url)
        self.assertEquals(response.status_code,200)
        self.assertTemplateUsed(response,'html/login.html')

    def test_logout_GET(self):
        response=self.client.get(self.logout_url)
        self.assertEquals(response.status_code,200)
        self.assertTemplateUsed(response,'html/logout.html')

```

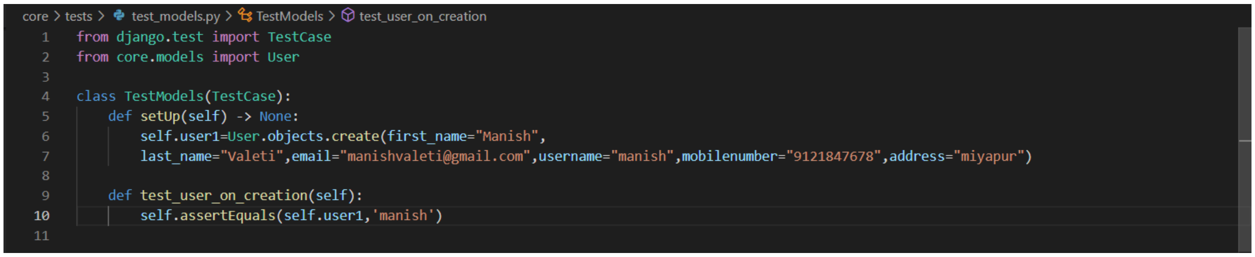
Views test output



```
C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>py manage.py test core
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.........
----------------------------------------------------------------------
Ran 9 tests in 0.067s

OK
Destroying test database for alias 'default'...

C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>
```

Models Testing:

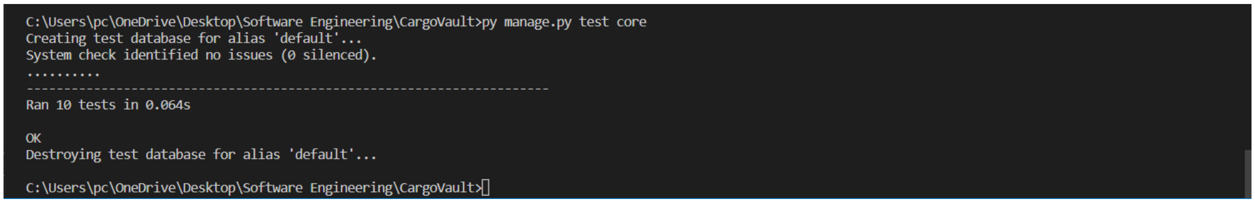
test_models.py



core > tests > test_models.py > TestModels > test_user_on_creation

```python
from django.test import TestCase
from core.models import User

class TestModels(TestCase):
    def setUp(self) -> None:
        self.user1=User.objects.create(first_name="Manish",
        last_name="Valeti",email="manishvaleti@gmail.com",username="manish",mobilenumber="9121847678",address="miyapur")

    def test_user_on_creation(self):
        self.assertEquals(self.user1,'manish')

```

Models test output:



```
C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>py manage.py test core
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..........
----------------------------------------------------------------------
Ran 10 tests in 0.064s

OK
Destroying test database for alias 'default'...

C:\Users\pc\OneDrive\Desktop\Software Engineering\CargoVault>
```