```
# Used to access user info
from django.contrib.auth import get_user_model
# Used when you're working with a database
from django.test import TestCase
# If provided the URL pattern name, which is assigned in urls.py
# reverse provides the URL
from django.urls import reverse
from .models import Article

# Run tests --> python manage.py test

class TestSite(TestCase):
  # Create a user and a sample post
  # This is run before any other tests below
  def setUp(self):
    # Create a user
    self.user = get_user_model().objects.create_user(
    username = 'sampleuser',
    email = 'sample@aol.com',
    password = 'password'
    )

    # Create a sample article
    self.article = Article.objects.create(
    title = 'Axehandle Hounds Ravage Minnesota',
    text = 'The axehandle hound is an American fearsome critter of Minnesota and Wisconsin.',
    author = self.user,
    )

  # Test value of an article title
  def test_article_title(self):
    article = Article(title='Axehandle Hounds Ravage Minnesota')
    self.assertEqual(str(article), article.title)

  # Test all expected content in an article
  def test_setting_all(self):
    self.assertEqual(f'{self.article.title}', 'Axehandle Hounds Ravage Minnesota')
    self.assertEqual(f'{self.article.author}', 'sampleuser')
    self.assertEqual(f'{self.article.text}', 'The axehandle hound is an American fearsome critter of
Minnesota and Wisconsin.')

  # Verify a 200 code, expected content and correct template
  def test_response(self):
    response = self.client.get(reverse('home'))
    self.assertEqual(response.status_code, 200)
    self.assertContains(response, 'Axehandle Hounds Ravage Minnesota')
    self.assertTemplateUsed(response, 'home.html')

  # Verify expected results from both a real and nonexistent
  # articles
  def test_article_detail_view(self):
    response = self.client.get('/article/1/')
    bad_response = self.client.get('/article/100/')
    self.assertEqual(response.status_code, 200)
```

```
self.assertEqual(bad_response.status_code, 404)
self.assertContains(response, 'Axehandle Hounds Ravage Minnesota')
self.assertTemplateUsed(response, 'article_detail.html')
```