Q2:Python application using pytest

Created all required files

```
master@master-vm:-/Desktop/q2$ cd retest-jenkins-1/
master@master-vm:~/Desktop/q2/retest-jenkins-1$ ls
app Jenkinsfile README.md requirements.txt tests
master@master-vm:~/Desktop/q2/retest-jenkins-1$ cd app
master@master-vm:~/Desktop/q2/retest-jenkins-1/app$ nano calculator.py
master@master-vm:~/Desktop/q2/retest-jenkins-1/app$ cd ..
master@master-vm:~/Desktop/q2/retest-jenkins-1$ cd tests/
master@master-vm:~/Desktop/q2/retest-jenkins-1/tests$ nano test_calculator.py
master@master-vm:~/Desktop/q2/retest-jenkins-1/tests$ cd ..
master@master-vm:~/Desktop/q2/retest-jenkins-1$ nano Jenkinsfile
master@master-vm:~/Desktop/q2/retest-jenkins-1$ nano requirements.txt
master@master-vm:~/Desktop/q2/retest-jenkins-1$
```

Calculator.py

```
GNU nano 6.2
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        raise ValueError("Division by zero is not allowed.")
    return a / b
```

Test_calculator.py

```
GNU nano 6.2
Import pytest
from app import calculator

def test_add():
    assert calculator.add(2, 3) == 5

def test_subtract():
    assert calculator.subtract(5, 2) == 3

def test_multiply():
    assert calculator.multiply(3, 4) == 12

def test_divide():
    assert calculator.divide(10, 2) == 5

def test_divide_by_zero():
    with pytest.raises(ValueError):
        calculator.divide(10, 0)
```

Jenkinsfile

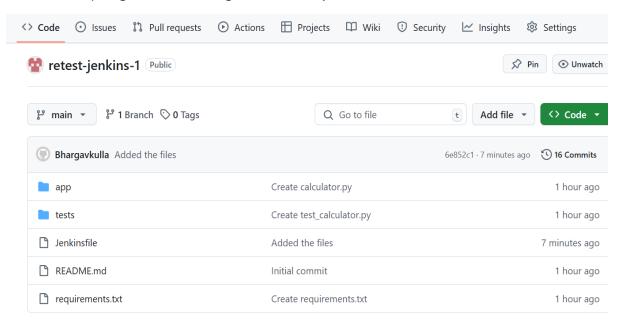
```
master@master-vm: ~/Desktop/q2/retes
  GNU nano 6.2
                                                                                                                                                                                                                                Jenkinsfile
pipeline {
agent any
     environment {
    VENV = 'venv'
     stages {
    stage('Checkout') {
                steps {
    git url: 'https://github.com/Bhargavkulla/retest-jenkins-1.git', branch: 'main'
           stage('Install virtualenv and Set Up Virtual Environment') { steps { sh '''}}
                            # Install virtualenv if it's not installed python3 -m pip install --user virtualenv
                            # Create the virtual environment using virtualenv python3 -m virtualenv $VENV
                            # Upgrade pip inside the virtual environment $VENV/bin/pip install --upgrade pip
                      # Install dependencies from requirements.txt
$VENV/bin/pip install -r requirements.txt
           stage('Run Pytest') {
    steps {
        sh '''
                            # Run pytest and generate a JUnit XML report
$VENV/bin/python -m pytest --junitxml=pytest-report.xml
           stage('Publish Test Results') {
    steps {
        junit '**/pytest-report.xml' // Look for the generated XML report
    post {
    always {
        // Clean up by removing the virtual environment after the pipeline finishes
        sh 'rm -rf $VENV || true'
}
```

Requirements.txt

```
GNU nano 6.2
pytest
coverage
```

Lets move to Github

Githuburl: https://github.com/Bhargavkulla/retest-jenkins-1



Output

