```
Selenium----> library------>collection module(component)
module-----> class,def,variable
3rd party library----------> not present--- -> install---->2
install ???? pip
open source
Automation---> UI,Database,API
version-----> 3.x now 4.x
python script---->selenium----->AUT----> application under test
concept----->
webapplication----->???
RR cycel---->Request and response
GUI-----  presentation
database-----> backend
API---->
json----> import json
'key':value        ---->str
dict----> key:value ----> dict   ----> key is immutable
2 methods----->
loads()---------> json object to python object
dumps() -----> python object to json object
webdriver------>???-----> browser
100%----->
what disadvan--->??
barcode ----->manual
payment  ----->manual
capcha -----> manual
Selenium------>
webdriver-----> class---->??
class----> enchapsulation
locator -----> variables
methods -----> actions
desktop ----->
No reporting capacuty------>unittest,pytest
from selenium import webdriver
requirements.txt
pip install requirements.txt-- 200+ ------>
virtual env------>
DeprecationWarning:
4.x ----> 3.x----> 12+
python2 dict   ----> unorder
python3.6+----> order dict

https://www.selenium.dev/documentation/webdriver/getting_started
/upgrade_to_selenium_4/#firefox-legacy

title=??
if title =="Upgrade to Selenium 4 | Selenium" ----> pass/fail
```

```python
#day1
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://www.selenium.dev/documentation/webdriver/getting_started/upgrade_to_selenium_4/#firefox-legacy')
if driver.title=="Upgrade to Selenium 4 | Selenium":
    print('pass')
else:
    print('Fail')
driver.close()
```

Types of frameworks----->selenium----
1 Modular Testing Framework------>used--->resume
every tc consider in sepreate----->test_login.py
----> test_dash.py
-----> test_resig.py
all the modules are independed to each
seprete copy of file(.py)----> is created
from module import class----> user file

2 Data driven framework----> USED
tc data is present inside outside of tc
we are going to use that data--->
test_login.py----> n user-----> test manager
username---->
password---->
used data---->
module--->openpyxl,mysql-connector or pymysql, csv----> pip install pymysql
exel(100),database,csv
mockey testing-----> test engineer

3 Keyword driven framework---->modular and data driver
web elements(locators),description,keyword----> excel,action(optional)

4 Hybrid framwork---->combination of all the framwework----> used
hybrid framwork with POM(page object module)----->yes
page-----> is collection of web elements(locator)
object----->yes----> based on object we call the method
module---->yes----> test_*.py or *_test.py

```
abc.py---->
    import openpyxl
    import login_page
    class Test_login:
        def test_enter_usename():
            locator----------------> page.py(all the locators)
            openpyxl----> usename/ database

        def test_enter_password():
            openpyxl----> usename/database

         def test_click_on_sub():
            click----> function/action

l1=login()
l1.test_enter_name()
100----> 5min----> 500min
```

framwork structure---> directory structure
module----> compont
1---> login--->3 file---->
login_page.py ------> all locator define
login_test.py -------> test cases---->
login_helper.py ------>setup(open my brower) and teardown(close brower) ----> unitest/pytest

driver.close()---->current brower will close
driver.quite()-----> all brower ----> chrome

how finding locators---->
tools----->extention----------> no one takking----->
chropath------>
sectorhub------>id,css,xpath,class

**locators--->web elements----> locator on brower
id ----->strong----> unique in nature----> unique-identify

----

class ---->mutlple on web browsers----> radio button
--->list--->li ul
----> checkbox
----> dropdwon

----

name ----> radio button,checkbox----> can be different in same group

input type="radio" name='Male'
input type="radio" name='Female'

----

linktext---->
full linktext-----> we specfy all the context of link
Register
partial linktext---->we specfy all the substring of link
Reg or ster

----

css---->** in deatils ---->
customise css selector
tag and id
tag and class
tag and attribute
tag class and attribute

----

xpath----> slow---->last
DOM---> document object model
abs xpath---> full path---> root node
/html/body/div[6]/div[1]/div[2]/div[2]/form/input

rel xpath---->partial----> midle node
//*[@id="small-searchterms"]
----

types----->user ----> time cosumming

css----> casting style sheet----> web developement

selenium3---->
a=find_element_by_id('')
str--------------->immutable----> new location
b=find_elements_by_class('')
list------>[]-----> python list----> mutable--->CRUD

selenium4----->
from selenium.webdriver.common.by import By
driver.find_element(By.ID,'id of element')
driver.find_element(By.CLASS,'class of element')
driver.find_element(By.NAME,'NAME of element')
driver.find_element(By.XPATH,'XPATH of element')
driver.find_element(By.PARTIAL_TEXT,'id of element')
driver.find_element(By.LINK_TEXT,'id of element')
driver.find_element(By.CSS,'id of element')

sleep()
Chrome()
get()
maximize_window()-----> maximize
close()
send_keys('input')

WAIT()---->
selenium 2 types ---->
implicit wait()---->if wait on brower level
explicit wait()---->if wait at locator or any

how to handle multiple windows----> 3 function
windrow_handler()--->collection of all the windows
current_window_handle()---->unique id of current open window
switch_to_window()----->switching betweens

```
l1=driver.windrow_handler()
l2=current_window_handle()
for l2 in l1:
    if l2.
        swith_to_window()
```

```python
import time
from selenium.webdriver.common.by import By
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://demo.nopcommerce.com/')
driver.maximize_window()
time.sleep(2)
#driver.find_element(By.ID,'small-searchterms').send_keys('Lenovo')
#driver.find_element(By.NAME,'q').send_keys('Lenovo')
#driver.find_element(By.XPATH,"//button[@class='button-1 search-box-button']").click()
#driver.find_element(By.LINK_TEXT,'Register').click()
driver.find_element(By.PARTIAL_LINK_TEXT,'Reg').click()
time.sleep(3)
driver.close()
```

```
MVT-----> django
M----->Model----> model.py-----> DATABASE in application
V----->View-----> bis----> processing the data----> CRUD
T----->Template--->html,css,jquery,javascript----> frontend

Commands in selenium----->
1 application commands----->variable
    get(url)--->
    title----->
    current_url---->
    page_source---->
2 Conditional statement---->***
is_displayed() -----> locator displayed ---> True
is_enabled() ----> if locator is clicable or taking input --->
True
is_selected() -----> radio

3 Browser Commands--->
close()----> single browser close
quite()----> open browser close
submit() ----> enter from keyboard

4 Navigation commands--->
forword() ----> next browser page
back()------>   previous browser page
refresh() -----> page refresh

5 WAIT---->
implicit wait---->sec
            if we want to apply wait for whole application
            all the locator
            driver.implicitly_wait(5)  # 5sec
explicit wait---->
    perticular codition,element
    WebDriverWait----> selenium.webdriver.
                #5sec

from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
locator.until(EC.presence_of_element_located(By.NAME,'q'))
expected wait is 5sec----> defect/bug
konsa wait()----->
time.sleep()  ---->
TC---->componet / Product ----> explication
FAIL-----> elementNOtprsent expection
    ---> elementnotdisploye
    ---> elementnotvisible
```

POM-----> page object module

type='checkbox' and contains(@id,'day')
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

alr--->
3 type ---> accept,dismiss,input
select---> dowpdown
multiple windows---> window,tab how handle--->
*** DATA Driven FRAMEWORk----> mini---> framework worked frame developed
1---> mini project/ Python mini /manual s
database----> 4-5hr
APIS ---> 2hr
realtime projects--->3-4hr
automitive---->
insurance domain/ossbss/photoshop management/hospital automatic system

```python
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://www.facebook.com/')
driver.maximize_window()
#print(driver.title)
#rint(driver.current_url)
var1=driver.current_url #'https://www.facebook.com/
for i in var1:
    print(i)

import sys

sys.exit(0)
import time
from selenium.webdriver.common.by import By
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('http://automationpractice.com/index.php')
driver.maximize_window()
#var=driver.find_elements(By.CLASS_NAME,'homeslider-container')
#print(len(var))
var=driver.find_elements(By.TAG_NAME,'a')

for i in var:
    print(i)

time.sleep(2)

#find_element or find_elements
#a=10 len(a)
#a=[1,2,3,4]--->
```

```python
#day4
import time
from selenium.webdriver.support.wait import WebDriverWait
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://demo.nopcommerce.com/register')
driver.maximize_window()
#elment=driver.find_element(By.XPATH,"//input[@id='small-searchterms']")
#print('display the element',elment.is_displayed()) #True
#print('enable the element',elment.is_enabled()) #True
#driver.close()

print('**No element is selected**')
male_locator=driver.find_element(By.XPATH,"//input[@id='gender-male']")
female_locator=driver.find_element(By.XPATH,"//input[@id='gender-female']")
print(male_locator.is_selected())  #false
print(female_locator.is_selected())  #false

print('**male is selected**')
male_locator.click()   #select
time.sleep(2)
print(male_locator.is_selected())  #True
print(female_locator.is_selected())  #false


print('**female is selected**')
female_locator.click()   #select
time.sleep(2)
print(male_locator.is_selected())  #False
print(female_locator.is_selected())  #True
driver.close()
```

```
import time
from selenium.webdriver.support.wait import WebDriverWait
from selenium import webdriver  #browser
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()

driver.get('https://www.google.com/')

driver.implicitly_wait(6)
#driver.maximize_window()
w1=WebDriverWait(driver,6)

search=driver.find_element(By.NAME,'q')
w1.until(EC.presence_of_element_located(By.NAME,'q'))

search.send_keys('hi msg')
search.submit()
driver.close()
```

```
import time
import pdb
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://itera-qa.azurewebsites.net/home/automation')
driver.maximize_window()
#single checkbox select
#mon=driver.find_element(By.XPATH,"//input[@id='monday']")
#mon.click()

#all the checkbox select
list_checkbox=driver.find_elements(By.XPATH,"//input[@type='checkbox' and contains(@id,'day')]")
#print(len(list_checkbox)) 7
#for i in range(len(list_checkbox)):# 0 to 6
   # list_checkbox[i].click()
#5,6 ---> click
#for i in range(len(list_checkbox)-2,len(list_checkbox)):# 7-2=5 to 7
    #list_checkbox[i].click()

#for i in range(len(list_checkbox)-4,len(list_checkbox)-2):
#range(3,5) ---->3,4
    #list_checkbox[i].click()

#for i in range(0,len(list_checkbox)-5):
    #list_checkbox[i].click()

time.sleep(5)
#for i in list_checkbox:
   # i.click()


driver.close()
```

```
import time
from selenium.webdriver.common.by import By
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://www.google.com/')  #facebook
driver.maximize_window()
driver.find_element(By.NAME,'q').send_keys('facebook')
time.sleep(5)
driver.find_element(By.NAME,'q').submit() #enter from keyboard
time.sleep(5)
```

```
import sys

sys.exit(0)
import time
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://www.facebook.com/')  #facebook
driver.maximize_window()
time.sleep(3)
driver.get('https://www.amazon.in/mobile-phones/b/?ie=UTF8&node=1389401031&ref_=nav_cs_mobiles') #amazon
time.sleep(3)
driver.back() #facebook
time.sleep(3)
driver.forward() #amazon
time.sleep(3)
driver.refresh()
```

```python
import sys



sys.exit(0)
import time
from selenium.webdriver.common.by import By
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://demo.nopcommerce.com/')
driver.maximize_window()
time.sleep(2)
driver.find_element(By.LINK_TEXT,'Register').click()
time.sleep(5)
driver.back()
time.sleep(5)
driver.forward()
time.sleep(5)
driver.refresh()

time.sleep(3)
driver.close()
```

```python
import sys
sys.exit(0)
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()

driver.get('https://www.facebook.com/')
driver.maximize_window()
#print(driver.title)
#rint(driver.current_url)
print(driver.page_source)



import sys


sys.exit(0)
var1=driver.current_url #'https://www.facebook.com/
l1=var1.split('.')
print(l1[1])


import time
from selenium.webdriver.common.by import By
from selenium import webdriver
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('http://automationpractice.com/index.php')
driver.maximize_window()
#var=driver.find_elements(By.CLASS_NAME,'homeslider-container')
#print(len(var))
var=driver.find_elements(By.TAG_NAME,'a')
l1=var.split()
print(l1)

time.sleep(2)
```

```
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
from selenium.webdriver.support.select import Select #dropdown

driver.get('https://www.opencart.com/index.php?route=account/register')
driver.maximize_window()

drp_contry=driver.find_element(By.XPATH,"//select[@id='input-country']")
#print(drp_contry.is_displayed())
drp=Select(drp_contry)  #dropdown is activate

#3 methods---->
#select_by_visible_text(' ')----> visible text on UI
#select_by_index() -----> 0 to len-1 ----> first(0) menu to last menu(len-1)
#select_by_value() -----> value front engineer

#drp.select_by_visible_text('Aruba')#user ko jo visible
#drp.select_by_value("4")

#drp.select_by_index(10)

r=drp.options
#for i in r:
   # print(i)
#print(len(r))
time.sleep(4)
driver.close()
```

```python
'''
Alerts /PopUp---->
driver.switch_to.alert -----> our context we move to alert window
text ----> text of alert
accept()----> ok click
dismiss() ----> cancel click

//button[normalize-space()='Click for JS Prompt']

Authentication alert -----> ???
user/pass ---> alert???


'''

import time
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()

driver.get('https://the-internet.herokuapp.com/javascript_alerts')
driver.maximize_window()

#pop1=driver.find_element(By.XPATH,"//button[normalize-space()='Click for JS Prompt']")
#pop1=driver.find_element(By.XPATH,"//button[normalize-space()='Click for JS Confirm']")
pop1=driver.find_element(By.XPATH,"//button[normalize-space()='Click for JS Alert']")
pop1.click()
time.sleep(5)
var1=driver.switch_to.alert

#var1.send_keys('Hi i am input box')

var1.accept()   # alert window click on OK buttom
#var1.dismiss()

time.sleep(5)
driver.close()
```

```
'''
https://the-internet.herokuapp.com/basic_auth


authencation---->

https://username:password@url


'''

import time
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()

driver.get('https://admin:admin@the-internet.herokuapp.com/basic_auth')
driver.maximize_window()

time.sleep(5)
driver.close()
```

```python
import time
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()

driver.get('http://automationpractice.com/index.php')
driver.maximize_window()
#driver.switch_to.new_window('tab') #to open new tab---> on same browser
driver.switch_to.new_window('window') #new browser open for next urls---> different browser
driver.get('https://www.selenium.dev/selenium/web/web-form.html')


time.sleep(5)

driver.close()
```

```
import time
#ActionChains
from selenium import webdriver
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()

drive=webdriver.Chrome()
drive.maximize_window()
drive.get('http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html')

at=drive.find_element(By.XPATH,"//button[normalize-space()='Automation Tools']")
act=ActionChains(drive)
act.move_to_element(at).perform()
time.sleep(3)
drive.find_element(By.XPATH,"//a[text()='TestNG']").click()
time.sleep(3)
drive.close()

import sys
sys.exit(0)
import time
from selenium import webdriver
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()

driver=webdriver.Chrome()
driver.maximize_window()
driver.get('http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html')
time.sleep(2)
automationTool=driver.find_element(By.XPATH,"//button[normalize-space()='Automation Tools']")
act=ActionChains(driver)
act.move_to_element(automationTool).pause(30).click(driver.find_element(By.XPATH,"//a[text()='Appium']")).perform()


import sys
sys.exit(0)
import time
from selenium import webdriver
```

```python
from selenium.webdriver import ActionChains
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
chromedriver_autoinstaller.install()

driver=webdriver.Chrome()
driver.maximize_window()
driver.get('http://seleniumpractise.blogspot.com/2016/08/how-to-perform-mouse-hover-in-selenium.html')
time.sleep(2)
automationTool=driver.find_element(By.XPATH,"//button[normalize-space()='Automation Tools']")
act=ActionChains(driver)
act.move_to_element(automationTool).perform()
time.sleep(2)
driver.find_element(By.XPATH,"//button[normalize-space()='Automation Tools']").click()
time.sleep(2)
print(driver.current_url)
```

```
''' AUtomation/framweork developement
How to handle multiple windows---->
window----> unique id

1. window_handles ----->
        all the windows id in 1 list--->[A,B,C,D]
        list[0]----> parent window ----> automation context
        list[1:]----> child windows

2. current_window_handle ---->
    unique id of current window----> parent----> l1[0]

3. switch_to.window(windowID) ---->
    from one window to next window
    context will switch from 1 to provide windowID

Programs----->A,B,C,D
l1[0]--->A
l1[1:]--->B,C,D

l1=driver.window_handles
c_window_id=driver.current_window_handle(id)
c_window_id=driver.current_window_handle(id_id_67832D)

l1=[id_0109A,id_78978B,id_67816C,id_67832D]
c_window_id=id_id_67832D

for w in l1:
    if w==c_window_id: #last
        driver.switch_to.window(c_window_id)
        driver.close()


parent window--->
child window1---->
child window 2--->
child window3---->

context---> is located on parent window
driver.close()--->single browser close--->
konsa window close hoga??----> automation through----> webdriver

driver.quite()----->all the open browsers
konsa all windows close hoga??-----> automation through ---->
webdriver
```

```
Selenium all senois---->
locatr=driver.find_element(By.XPATH,'')
locator.send_keys('file_path') # start to end (file name)


screen shot--->
                save_screenshot()
                get_screenshot_as_file()

data driven---> openpyxl
database----> pymysql---> sqlworkbench
api ----> postman,requests
framework----> unitest,pytest
```

```
import openpyxl

# we need to install 3rd party library----> python present but
we use
#file---> workbook--->sheets--->rows--->cells(r,c)

file1="E:\Book1.xlsx"
workbook=openpyxl.load_workbook(file1)
sheet=workbook.active #parent sheet

row=sheet.max_row #5
column=sheet.max_column #3

for r in range(2,row+1):   #range(6)--->   #0(title) to 5
    for c in range(1,column+1):
        if r==2 and c==2:
            sheet.cell(r,c).value='Mumbai'

workbook.save(file1)



import sys
sys.exit(0)
#all row and column se data read
for r in range(2,row+1):   #range(6)--->   #0(title) to 5
    for c in range(1,column+1):
        print(sheet.cell(r,c).value,end='   ')
    print()




import sys

sys.exit(0)
v=sheet.cell(4,2).value #Mumbai
print(v)
v=sheet.cell(1,2).value  #city
print(v)
v=sheet.cell(1,1).value #name
print(v)
v=sheet.cell(2,3).value #89545614
print(v)
```

```
#ddf functions
import openpyxl
def get_row_count(file,sheetname):
    workbook=openpyxl.load_workbook(file)
    sheet=workbook[sheetname]
    return sheet.max_row

def get_column_count(file,sheetname):
    workbook = openpyxl.load_workbook(file)
    sheet = workbook[sheetname]
    return sheet.max_column

def read_Data(file,sheetname,rowno,colno):
    workbook=openpyxl.load_workbook(file)
    sheet=workbook[sheetname]
    return sheet.cell(rowno,colno).value

def write_data(file,sheetname,rowno,colno,data):
    workbook=openpyxl.load_workbook(file)
    sheet=workbook[sheetname]
    sheet.cell(rowno,colno).value=data
    workbook.save(file)
```

```python
import time
import function_for_ddl
from selenium import webdriver
import chromedriver_autoinstaller
from selenium.webdriver.common.by import By
chromedriver_autoinstaller.install()
driver=webdriver.Chrome()
driver.get('https://www.selenium.dev/selenium/web/web-form.html')
driver.maximize_window()

file1='E:\VB.xlsx'
row=function_for_ddl.get_row_count(file1,'Sheet1')

for r in range(1,row+1):
    usrname=function_for_ddl.read_Data(file1,'Sheet1',r,1)
    password=function_for_ddl.read_Data(file1,'Sheet1',r,2)
    print(usrname,password)
    #inputbox=driver.find_element(By.XPATH,"//input[@id='my-text-id']")
    #inputbox.send_keys(usrname)
    #time.sleep(5)

#passwordbox=driver.find_element(By.XPATH,"//input[@name='my-password']")
    #passwordbox.send_keys(password)
    #time.sleep(5)
    #driver.find_element(By.XPATH,"//button[normalize-space()='Submit']").click()
    #time.sleep(5)
```

```
2 framework----
unittest or pytest -----> integration for all framework
import unittest -------->
import pytest ------>
pytest,unitest----> OS----> linux
*** working people---->**

__name__ -----> present modulename in python
name==main ----> current module execute
other module name==modulename

if __name__ == '__main__' :
    current module code

__init__.py -----> interpreted--->compiler

unittest ----->inbuild framework
import unittest
unitest.TestCase -----> class
    D/I-----> unittest

test case name unittest-----> aplh

assert statement---->actual expected
assertequal() ---->pass
assertnotequal() ----> pass

defect----> developer
fixtures in unittest and pytest---->

UI testing----->
setup--->open brower
testcases
teardown---->close brower,quite

database testing----->
setup--->connection to data
testcases
teardown---->connection close
data driver testing/keyword driven----->
setup--->file-->woorkbook--->sheets--->row--->cell
testcases ----> samplefile.writeadd()
teardown---->save()

API testing
setup---> r=request.get()
testcases----> v=r.response
```

assertequal(v,200)
teardown---->

samplefile----> 4 functions
maxrow,maxcolumn,read,write

fixtures----> setup(prerequist) and cleanup activity
setup(self)
teardown(self)
@classmethod
setUpClass(cls)
@classmethod
tearDownClass(cls)

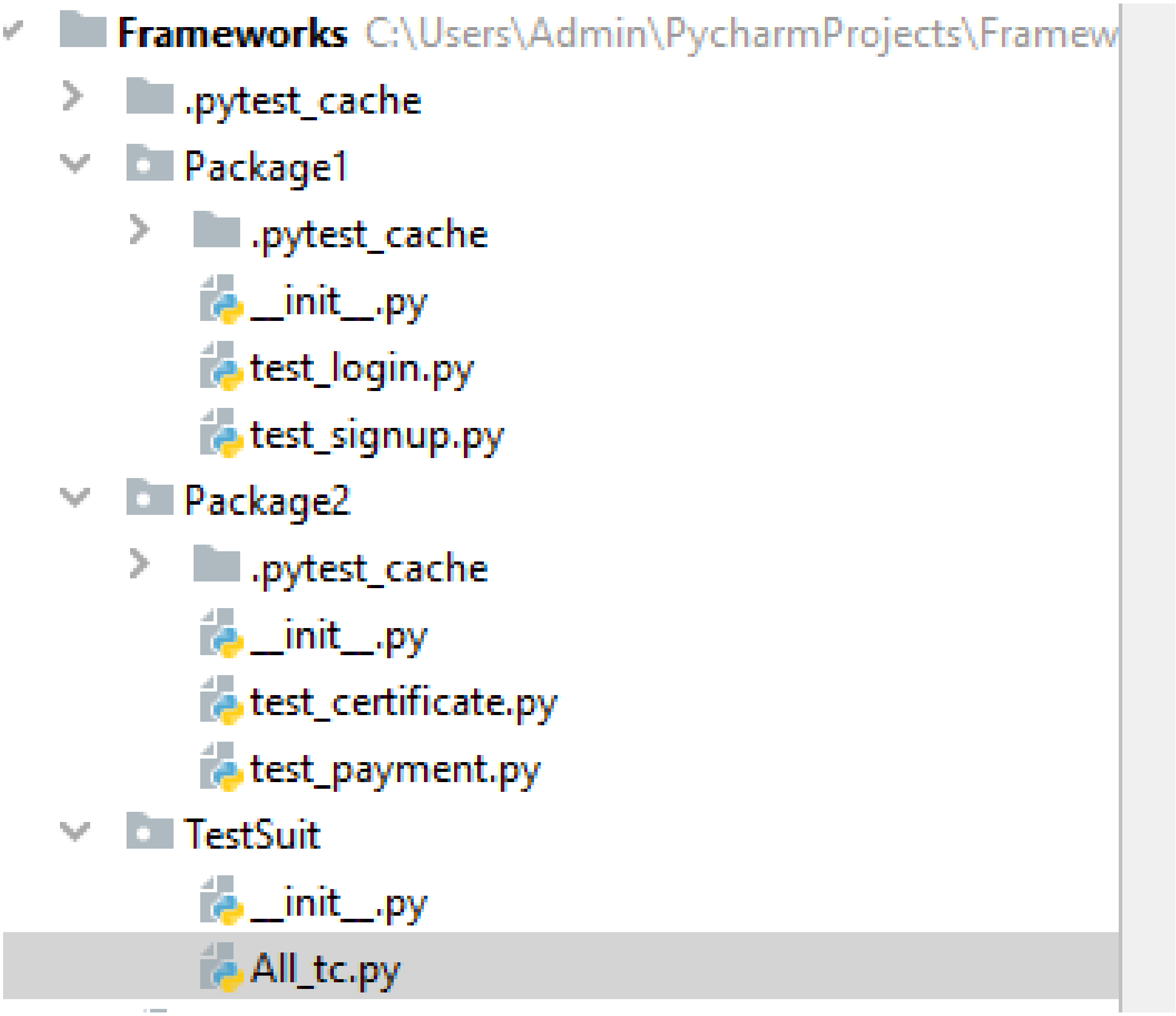
skip---->
@unittest.skip ----> decorator ---> perticular tc will skip

Ok
failure
error

1 display our result on console we can our result in log file
2 all the Case will be execute based on Aplhabical
3 as part of batch(testsuit,job) execution, all the tc from testcase class will be execute
4 if tc name is same----> python support---> only recent tc
method overloading----> in same class method name same
5 fixture---> 4method---> setup,teardown,setupclass,teardownclass
6 grouping TC---is not supported in Unittest

Test Suit--->Unittest
suit ---> collection of all the tc
Master suit--->(700+)---->all tc type
stable master suite------> 90 %   10%---> payment,bar,scanner,rect or angular
proctator tool----> agular js,react

5 tc---->
failed at 3??



```
Frameworks C:\Users\Admin\PycharmProjects\Framew
    .pytest_cache
    Package1
        .pytest_cache
        __init__.py
        test_login.py
        test_signup.py
    Package2
        .pytest_cache
        __init__.py
        test_certificate.py
        test_payment.py
    TestSuit
        __init__.py
        All_tc.py
```

```python
#test_signup.py
import unittest
class SignUpTest(unittest.TestCase):
    def test_signupbygmail(self):
        print('this tc is for gmail signup-')
    def test_signupbyphone(self):
        print('this tc is for phone signup')
    def test_signupbyfacebook(self):
        print('this tc is for facebook signup')
if __name__ == "__main__":
    unittest.main()
```

```python
#test_login.py
import unittest
class LoginTest(unittest.TestCase):
    def test_loginbygmail(self):
        print('this tc is for gmail login')
    def test_loginbyphone(self):
        print('this tc is for phone login')
    def test_loginbyfacebook(self):
        print('this tc is for facebook login')
if __name__== "__main__":
    unittest.main()
```

```python
#test_cerrtificate
import unittest
class CerficateTest(unittest.TestCase):
    def test_certificatebyname(self):
        print('this tc is for name of vendor')
    def test_certificatebyformate(self):
        print('this tc is for pdf formate')

if __name__ == "__main__":
    unittest.main()
```

```python
#test_payment
import unittest
class PaymentTest(unittest.TestCase):
    def test_paymentbydollor(self):
        print('this tc is for payment dollor')
    def test_paymentbyrupees(self):
        print('this tc is for payment by rupees')

if __name__ == "__main__":
    unittest.main()
```

```python
#test_suit file
import unittest
from Package1.test_login import LoginTest
from Package1.test_signup import SignUpTest
from Package2.test_payment import PaymentTest
from Package2.test_certificate import CerficateTest

t1=unittest.TestLoader().loadTestsFromTestCase(LoginTest)
t2=unittest.TestLoader().loadTestsFromTestCase(SignUpTest)
t3=unittest.TestLoader().loadTestsFromTestCase(PaymentTest)
t4=unittest.TestLoader().loadTestsFromTestCase(CerficateTest)

santitysuit=unittest.TestSuite([t1,t2])
functionalsuit=unittest.TestSuite([t3,t4])
mastersuit=unittest.TestSuite([t1,t2,t3,t4])

unittest.TextTestRunner().run(santitysuit)
```