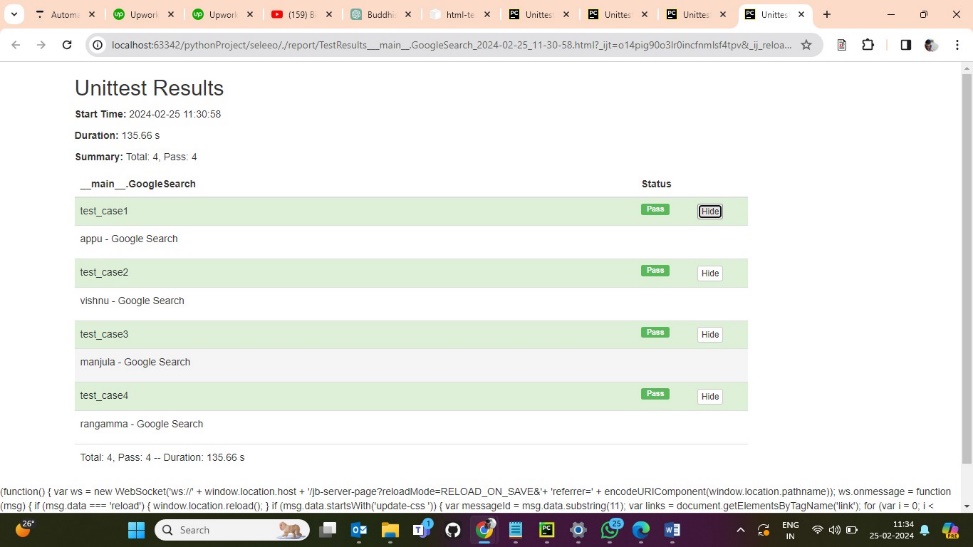
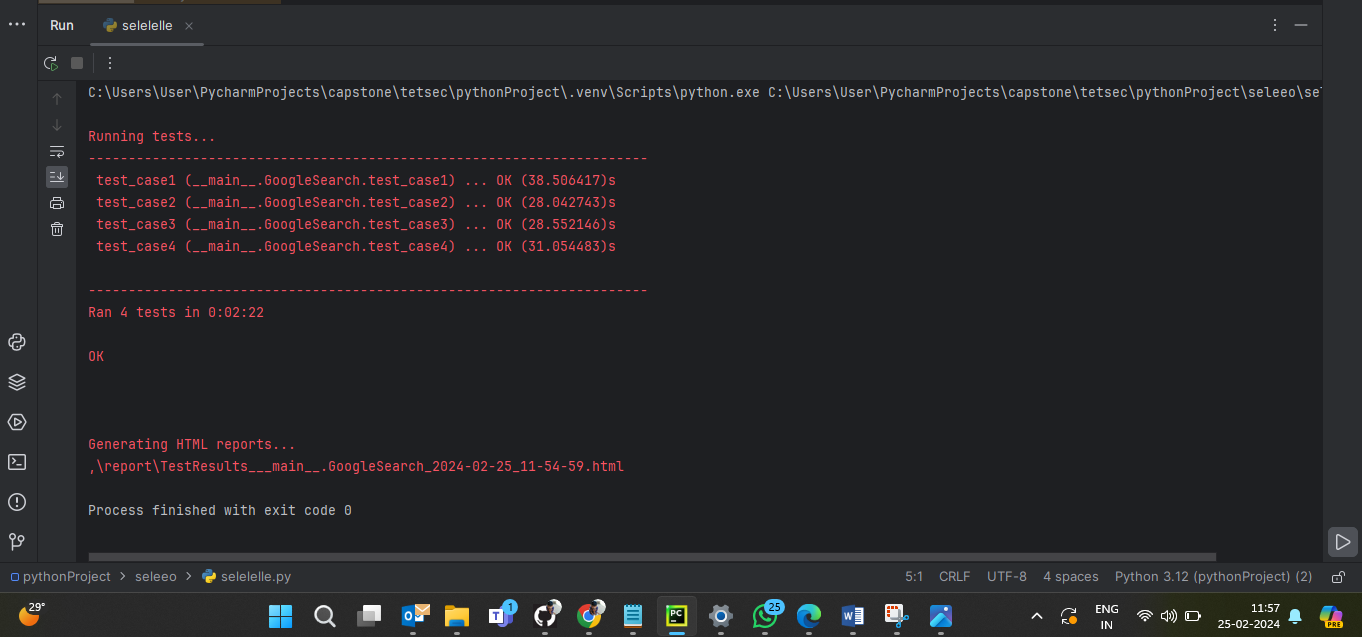
from selenium import webdriver  
from selenium.webdriver.common.by import By  
import HtmlTestRunner  
import unittest  
class GoogleSearch(unittest.TestCase):  
 def setUp(self):  
 self.browser = webdriver.Chrome ( )  
 self.browser.get("https://www.google.com/webhp?hl=en&sa=X&ved=0ahUKEwjDxfrt6JGEAxVNwjgGHcFVCGcQPAgJ")  
 self.browser.implicitly\_wait(5)  
 def tearDown(self):  
 print(self.browser.title)  
 self.browser.quit()  
 def test\_case1(self):  
 self.browser.find\_element(By.NAME, "q").send\_keys("appu")  
 self.browser.find\_element(By.NAME, 'btnK').click()  
 def test\_case2(self):  
 self.browser.find\_element(By.NAME, "q").send\_keys("vishnu")  
 self.browser.find\_element(By.NAME, 'btnK').click()  
 def test\_case3 ( self ):  
 self.browser.find\_element ( By.NAME , "q" ).send\_keys ( "manjula" )  
 self.browser.find\_element ( By.NAME , 'btnK' ).click ( )  
 def test\_case4 ( self ):  
 self.browser.find\_element ( By.NAME , "q" ).send\_keys ( "rangamma" )  
 self.browser.find\_element ( By.NAME , 'btnK' ).click ( )  
if \_\_name\_\_ == "\_\_main\_\_":  
 unittest.main(testRunner=HtmlTestRunner.HTMLTestRunner(output=",/report/"))

output details: to check the report the HTMLtestrunner



Console window:



python to scrping the data from any website:

import requests

from bs4 import BeautifulSoup

products\_track = [

{

"product\_url": "https://www.flipkart.com/redmi-10-caribbean-green-64-gb/p/itmbfb8dfcdd6e10?pid=MOBGC9GYQGGKWEU3&lid=LSTMOBGC9GYQGGKWEU3SLBNSD&marketplace=FLIPKART&q=mobiles&store=tyy%2F4io&srno=s\_1\_10&otracker=search&otracker1=search&fm=organic&iid=e7a15dcc-cc0f-47d8-bf6f-891cdc6284ab.MOBGC9GYQGGKWEU3.SEARCH&ppt=hp&ppn=homepage&ssid=xi23af1i7k0000001675351739831&qH=eb4af0bf07c16429",

"name": "REDMI 10 (Caribbean Green, 64 GB) (4 GB RAM)",

"taeget price": (12000)

},

{

"product\_url": "https://www.flipkart.com/motorola-g62-5g-frosted-blue-128-gb/p/itm914ff5c9b9aaf?pid=MOBGEDT5BYG68GNS&lid=LSTMOBGEDT5BYG68GNSVKARNI&marketplace=FLIPKART&q=mobiles&store=tyy%2F4io&srno=s\_1\_5&otracker=search&otracker1=search&fm=organic&iid=e7a15dcc-cc0f-47d8-bf6f-891cdc6284ab.MOBGEDT5BYG68GNS.SEARCH&ppt=hp&ppn=homepage&ssid=xi23af1i7k0000001675351739831&qH=eb4af0bf07c16429",

"name": "MOTOROLA G62 5G (Frosted Blue, 128 GB) (8 GB RAM)",

"taeget price": (8000)

},

{

"product\_url": "https://www.flipkart.com/cellecor-r50/p/itmad8912485f34b?pid=MOBG8P9EYHG6GD9A&lid=LSTMOBG8P9EYHG6GD9AWZNHNB&marketplace=FLIPKART&q=mobiles&store=tyy%2F4io&srno=s\_1\_4&otracker=search&otracker1=search&fm=organic&iid=en\_uiPrbKYHfX0mzM4Npt9CBGfJXTwrSniLvW81uWT%2FyWuVHkjkLeG82yOUuflYtntALMwIB8mSda6b7urhkoNrbA%3D%3D&ppt=hp&ppn=homepage&ssid=xi23af1i7k0000001675351739831&qH=eb4af0bf07c16429",

"name": "Cellecor R50 (ROYAL BLACK)",

"taeget price": (1200)

},

{

"product\_url": "https://www.flipkart.com/motorola-e40-carbon-gray-64-gb/p/itm0ca635007c9e2?pid=MOBG2EMWUMUFGSZE&lid=LSTMOBG2EMWUMUFGSZEJNGZMU&marketplace=FLIPKART&q=mobiles&store=tyy%2F4io&spotlightTagId=BestsellerId\_tyy%2F4io&srno=s\_1\_1&otracker=search&otracker1=search&fm=Search&iid=e7a15dcc-cc0f-47d8-bf6f-891cdc6284ab.MOBG2EMWUMUFGSZE.SEARCH&ppt=sp&ppn=sp&ssid=xi23af1i7k0000001675351739831&qH=eb4af0bf07c16429",

"name": "MOTOROLA e40 (Carbon Gray, 64 GB) (4 GB RAM)",

"taeget price": (12000)

},

{

"product\_url": "https://www.flipkart.com/motorola-g62-5g-midnight-gray-128-gb/p/itm37da299ffb2d0?pid=MOBGEDT5ZZMZQZSJ&lid=LSTMOBGEDT5ZZMZQZSJFXDZSQ&marketplace=FLIPKART&q=mobiles&store=tyy%2F4io&srno=s\_1\_2&otracker=search&otracker1=search&fm=organic&iid=e7a15dcc-cc0f-47d8-bf6f-891cdc6284ab.MOBGEDT5ZZMZQZSJ.SEARCH&ppt=hp&ppn=homepage&ssid=xi23af1i7k0000001675351739831&qH=eb4af0bf07c16429",

"name": " MOTOROLA G62 5G (Midnight Gray, 128 GB) (6 GB RAM)",

"taeget price": (16000)

}

]

def give\_product\_price(url):

headers = {

"user\_agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36"

}

page = requests.get(url, headers=headers)

soup = BeautifulSoup(page.content, "html.parser")

product\_price = soup.find("div", {"class": "\_30jeq3 \_16Jk6d"})

return product\_price.string

results\_file = open('my\_result\_file\_txt', 'w')

try:

for every\_product in products\_track:

product\_price\_returned = give\_product\_price(every\_product.get("product\_url"))

print(product\_price\_returned + ' ' + every\_product.get("name"))

deal\_price = product\_price\_returned = product\_price\_returned[1:]

deal\_price = deal\_price.replace(",", "")

deal\_price = int(deal\_price)

target\_price= int (every\_product.get("taeget price"))

if deal\_price < target\_price:

print("product on deal price")

results\_file.write(every\_product.get(

"name") + ' - \t' + ' Available at Target Price ' + ' Current Price - ' + str(deal\_price) + '\n')

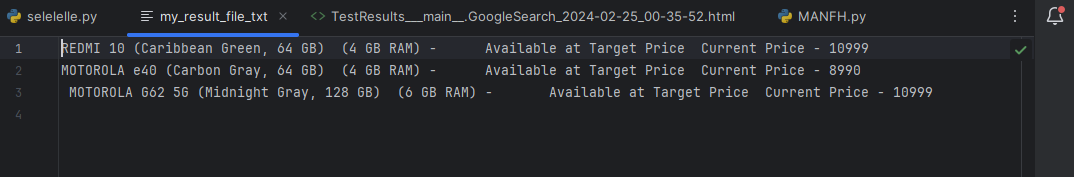
else:

print("sorry its in same price")

finally:

results\_file.close()

Result file extracted in .txt file :



output result file:

