#Question 1

#Import selenium webdriver

from selenium import webdriver

#Import selenium selector

from selenium.webdriver.common.by import By

#Import selenium service in case chromedriver does not work in the testing environment

from selenium.webdriver.chrome.service import Service

#Import chrome driver

from webdriver\_manager.chrome import ChromeDriverManager

#Import os to get current path name

import os

#function login parameters username, password, and the complete path for the output screenshot file

def login(pUsername, pPassword, pFilename):

#Define the chrome driver

driver = webdriver.Chrome('C:\Drivers\chromedriver')

#Uncommend the code below in case the Chromedriver does not work in the testing environment

#s = Service(ChromeDriverManager().install())

#driver = webdriver.Chrome(service=s)

#Open the destination URL

driver.get("https://practicetestautomation.com/practice-test-login/")

#Identify the element username and store it in variable eUsername

eUsername = driver.find\_element(By.ID, "username")

#Identify the element password and store it in variable ePassword

ePassword = driver.find\_element(By.ID, "password")

#Identify the submit button element username and store it in variable eSubmit

eSubmit = driver.find\_element(By.ID, "submit")

#Clear and enter the field username

eUsername.clear()

eUsername.send\_keys(pUsername)

#Clear and enter the field password

ePassword.clear()

ePassword.send\_keys(pPassword)

#similate the click submit button event

eSubmit.click()

#print current URL

print(driver.current\_url)

#Maximize the browser for screenshot

driver.maximize\_window()

#Take screenshot and save as the filename from the pFilename parameter

driver.save\_screenshot(pFilename)

#Close the browser

driver.close()

return

#Define correct username

correctUsername = "student"

#Define correct password

correctPassword = "Password123"

#Define incorrect username

incorrectUsername = "incorrectUser"

#Define incorrect password

incorrectPassword = "incorrectPassword"

#Get the initial input from user

choice = input("Input test case 1, 2, 3, or q for quit: ")

#Prompt until user has input a valid choice: 1, 2, 3, or q

while choice not in ['1', '2', '3', 'q']:

choice = input("Re-enter your choice: ")

#Get current py file path

dirname = os.path.dirname(\_\_file\_\_)

#Construct the output screenshot filename using the current py file path and the choice

screenshot = dirname + "/case" + choice + ".png"

if choice == '1':

#Test case 1, correct username and password

login(correctUsername, correctPassword, screenshot)

elif choice == '2':

#Test case 2, incorrect username

login(incorrectUsername, correctPassword, screenshot)

elif choice == '3':

#Test case 3, correct username and incorrect password

login(correctUsername, incorrectPassword, screenshot)

#Question 2

#Import pandas

import pandas

#Read source file

result = pandas.read\_csv("result.csv")

#Print the highest exam mark

print("Highest Exam Marks:")

print(result["Exam"].max())

#Print average assignment 1 mark

print("Average Assignment1 Marks:")

print(result["Assignment1"].mean())

#Locate student with empty assignment 2 marks and print their studint id and names

print("Without Submitting Assignment2:")

print(result.loc[result["Assignment2"].isnull(), ["Stud\_ID", "Name"]])

#Calculate final mark and assign to new series Final\_Marks

result["Final\_Marks"] = result["Assignment1"] + result["Assignment2"] + result["Class\_Participation"] + result["Exam"]

#Add Final Grade series

result["Final\_Grade"] = ""

#For each row

for i in range(len(result)):

#Get Final\_Marks from the second last series and convert to float for comparison

mark = float(result.iloc[i,-2])

#Grade E if less than 50

if mark < 50:

grade = "E"

#Grade D if between 50 and less than 65

elif mark < 65:

grade = "D"

#Grade C if between 65 and less than 80

elif mark < 80:

grade = "C"

#Grade B if between 80 and less than 90

elif mark < 90:

grade = "B"

#Grade A if more than or equal to 90

elif mark >= 90:

grade = "A"

#Otherwise, final grade is not a number

else:

grade = ""

#Assign grade to the last series Final Grade

result.iloc[i,-1] = grade

#Print the whole dataframe

print(result)

#Save to result.xlsx excel file

with pandas.ExcelWriter("result.xlsx", engine="openpyxl", mode="w") as writer:

result.to\_excel(writer, sheet\_name="Assessment", index=False)