

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ST JOSEPH ENGINEERING COLLEGE, Mangaluru
2021-2022**



**ASSIGNMENT
ON
APPLICATION DEVELOPMENT USING PYTHON**

Submitted To

Dr.Melwin D'souza

Submitted by

Jyothi Lisha Vas - 4SO19CS070
K Divya Pai - 4SO19CS071
Pooja - 4SO19CS108
Ritvik S Shetty - 4SO19CS128

Problem Statement – 1:

Develop a python program for fetching course information from Coursera. Coursera Inc. is a U.S.-based massive open online course provider founded in 2012 by Stanford University computer science professors Andrew Ng and Daphne Koller. Coursera works with universities and other organizations to offer online courses, certifications, and degrees in a variety of subjects.

It is a time consuming and tedious job to browse through courses on the website and decide if the course is suitable or not. To overcome this problem, we have written a python code with the help of which the user can easily fetch the required data , to decide if the course is suitable for him or not. The user needs to enter the course names correctly ,for which he wants to fetch data.

The required information is fetched using web scraping technique (Beautiful Soup) and selenium framework and generates an excel sheet to insert the information that has been fetched.

The user can compare the different courses and select the appropriate one using its url.

Source Code:

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
import requests
from bs4 import BeautifulSoup
from openpyxl import Workbook

from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException

wb = Workbook()                                     #creates a new workbook
sheet = wb.active                                    #accesses the current working sheet of
that workbook
```

```

sheet.title = "Coursera"                                # first row of the excel
sheet['A1'] = "Course name"
sheet['B1'] = "Rating"
sheet['C1'] = "Institute"
# sheet['D1'] = "What we will learn"
sheet['E1'] = "No of enrollments"
sheet['F1'] = "Start date"
sheet['G1'] = "Instructor(s)"
sheet['H1'] = "Language"
sheet['I1'] = "Time Duration"
sheet['J1'] = "Financial Aid"
sheet['K1'] = "URL"

row = 2                                                 #initializing the row as 2

url=[]                                                 #list that stores all the course URLs
course=[]                                              #list that stores all the entered
course names
num = int(input("Enter number of courses you want: "))

for i in range(num):
    print(f'Enter course name {i+1} :')
    name=input()
    course.append(name)                               #adds the names to the list

try:
    for c in course:
        browser=webdriver.Chrome()                  #chrome webdriver
        browser.get('http://coursera.com')          #accessing coursera
website
        elem= browser.find_element_by_class_name('nostyle') #used to close
the popup
        elem.click()

        elem = WebDriverWait(browser,
5).until(EC.presence_of_element_located((By.CLASS_NAME,
"react-autosuggest__input"))) #selects the searchbar

        elem.send_keys(c)                           # sends the keys entered to the search
bar
        time.sleep(10)

elem=browser.find_element_by_id('react-autowhatever-1-section-0-item-0')
#selects the course which comes first
        elem.click()

```

```

        url.append(browser.current_url)
        # appends the url to the list

    except TimeoutException:

        print("Loading took too much time!")

print('scrapping..')

for link in url:

    res = requests.get(link)                                     #make a
request to a webpage and return the status code

    soup = BeautifulSoup(res.content, "html.parser")           #fetching
and parsing the data using beautiful soup

    title = soup.find('h1', class_="banner-title").text          #storing
all the scraped items
    rating = soup.find('span', class_="rating-text").text
    unvname = soup.find('h3', class_="rc-Partner_title").text
    # learn = soup.find('ul',class_="_jyhj5r list-style-none p-a-0 p-l-1
m-b-0").text
    enroll = soup.find(class_="_1fpiay2").text
    strtdate = soup.find(class_="startdate m-t-1s rc-StartDateString
font-xs").text
    faid = soup.find(class_="button-link finaid-link").text
    lang = soup.findAll("div", class_="_16ni8zai")[-1].text
    time1 = soup.findAll("div", class_="_16ni8zai")[-2].text

    instructors = []
    for i in soup.findAll("h3", class_="instructor-name"):
        instructors.append(i.text)                                #appending all
the instructor names into the list

    sheet['A'+str(row)] = title                               # storing the
scraped values into the excel sheet
    sheet['B'+str(row)] = rating
    sheet['C'+str(row)] = unvname
    # sheet['D'+str(row)] = learn
    sheet['E'+str(row)] = enroll
    sheet['F'+str(row)] = strtdate
    sheet['G'+str(row)] = ",".join(instructors)
    sheet['H'+str(row)] = lang
    sheet['I'+ str(row)] = time1
    sheet['J'+str(row)] = faid
    sheet['K'+str(row)] = link

    row += 1

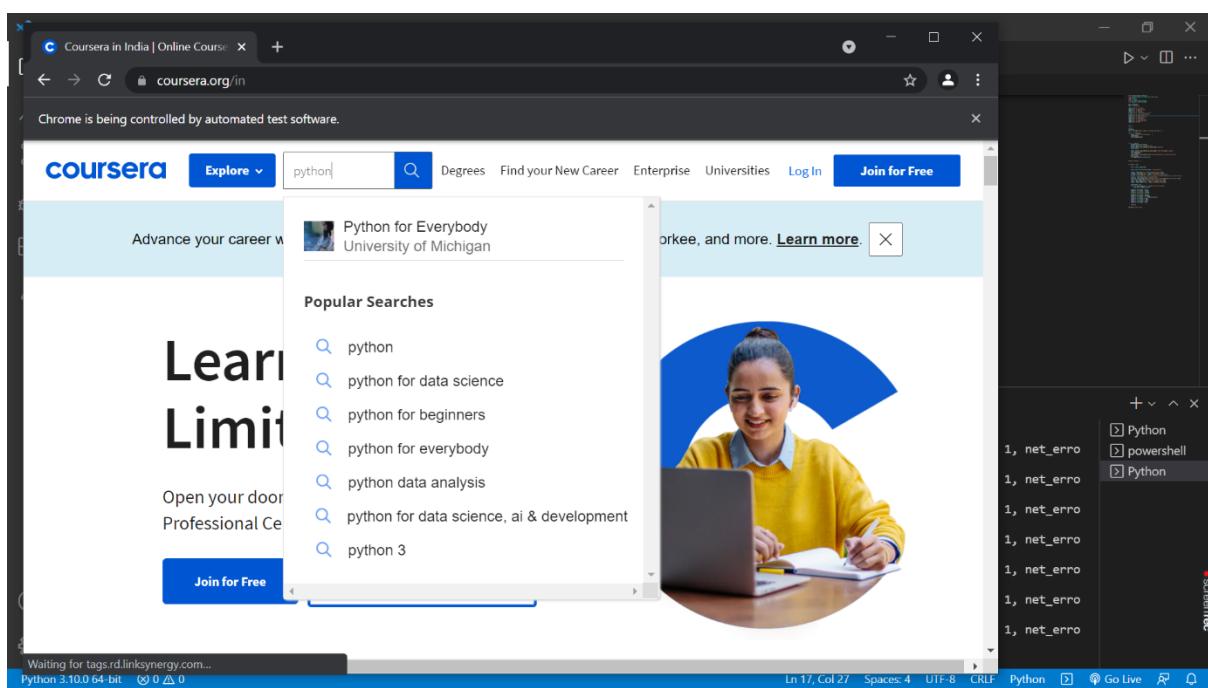
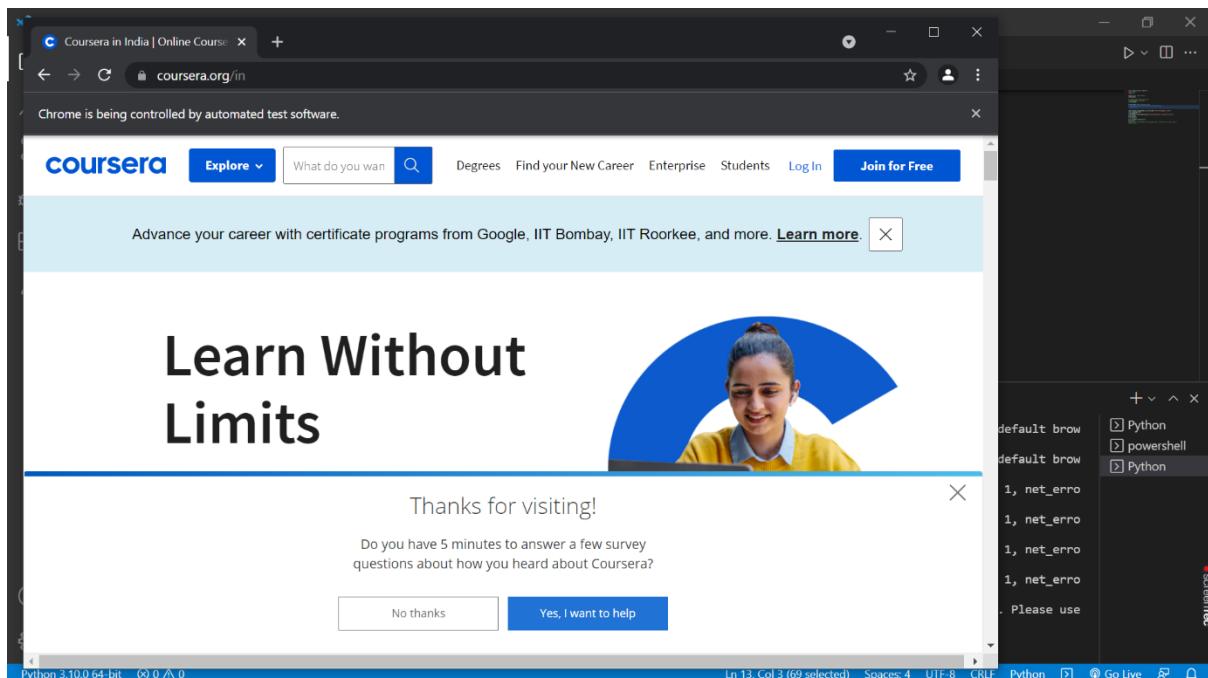
```

```
wb.save("ritvik.xlsx")  
# saving the  
excel
```

Output

The screenshot shows the Visual Studio Code interface. The code editor displays a Python script named `finalmain.py` which uses Selenium and OpenPyXL to interact with an Excel file. The terminal window shows the execution of the script and user input for course details. The file explorer on the left shows other files in the project directory.

```
from selenium import webdriver  
from selenium.webdriver.common.keys import Keys  
import time  
import requests  
from bs4 import BeautifulSoup  
from openpyxl import Workbook  
  
wb = Workbook()  
sheet = wb.active  
  
sheet.title = "Coursera"  
sheet['A1'] = "Course name"  
sheet['B1'] = "Rating"  
sheet['C1'] = "Institute"  
sheet['D1'] = "What we will learn"  
sheet['E1'] = "No of enrollments"  
  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/powershell  
  
PS C:\Users\Ritvik S Shetty\Desktop\adp> & "C:/Users/Ritvik S Shetty/AppData/Local/Programs/Python/Python310/python.exe" "c:/Users/Ritvik S Shetty/Desktop/adp/finalmain.py"  
Enter number of courses you want: 4  
Enter course name1 :  
python  
Enter course name2 :  
java  
Enter course name3 :  
finance  
Enter course name4 :  
marketing
```



Python for Everybody | Coursera

coursera.org/specializations/python?

Chrome is being controlled by automated test software.

coursera Explore What do you want? Degrees Find your New Career Enterprise Universities Log In Join for Free

Browse > Computer Science > Software Development

Offered By **UNIVERSITY OF MICHIGAN**

Python for Everybody Specialization

Learn to Program and Analyze Data with Python. Develop programs to gather, clean, analyze, and visualize data.

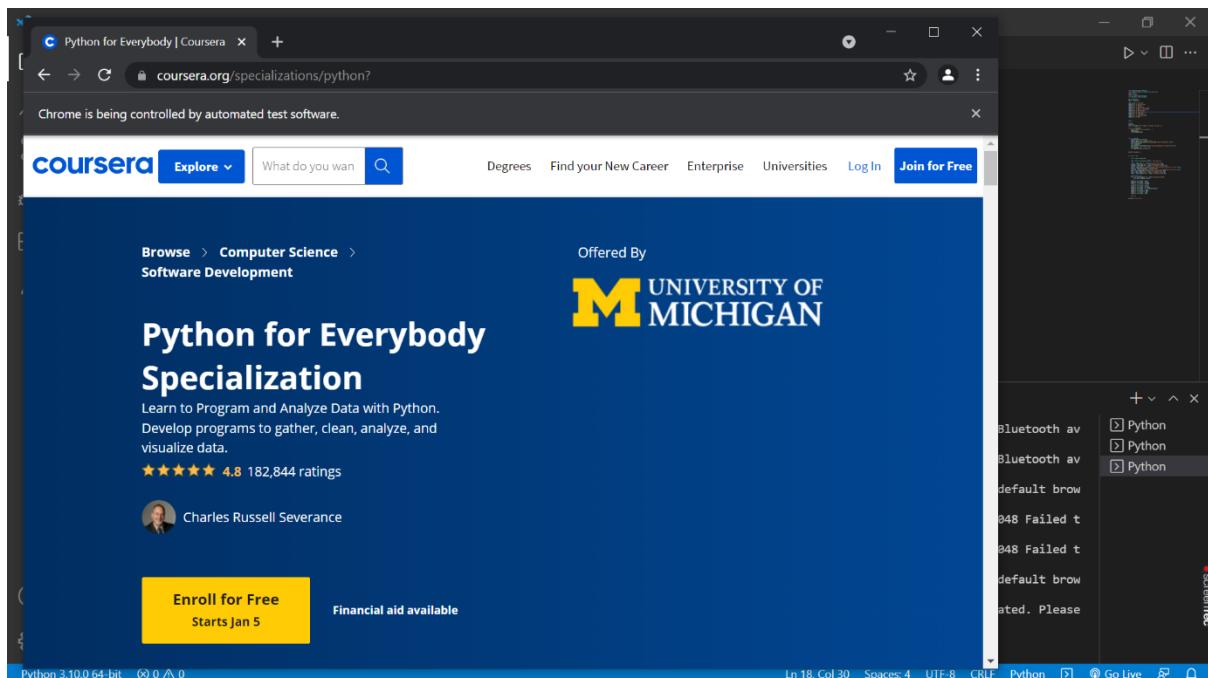
★★★★★ 4.8 182,844 ratings

Charles Russell Severance

Enroll for Free Starts Jan 5 Financial aid available

Python 3.10.0 64-bit 0 0 0

Ln 18, Col 30 Spaces: 4 UTRF Python Go Live



Coursera in India | Online Courses

coursera.org/in

Chrome is being controlled by automated test software.

coursera Explore java Degrees Find your New Career Enterprise Universities Log In Join for Free

Advance your career with Coursera

Lear

Limit

Open your door to Professional Careers

Join for Free

Java Programming and Software Engineering Fundamentals Duke University

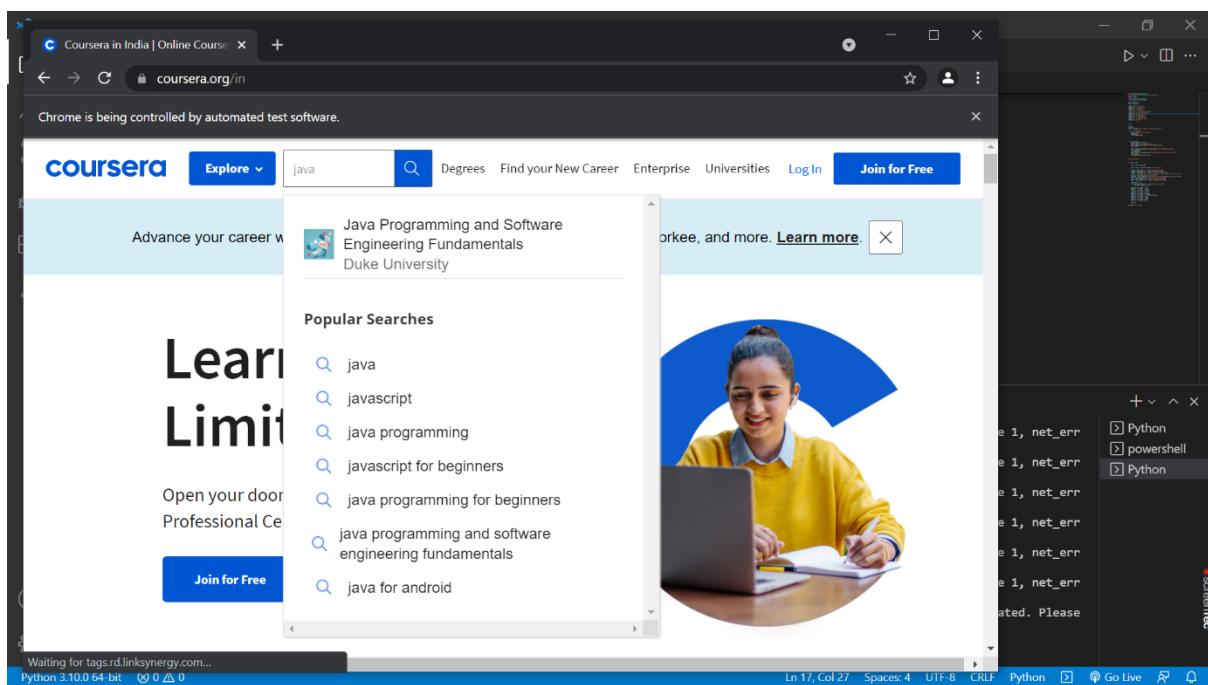
Popular Searches

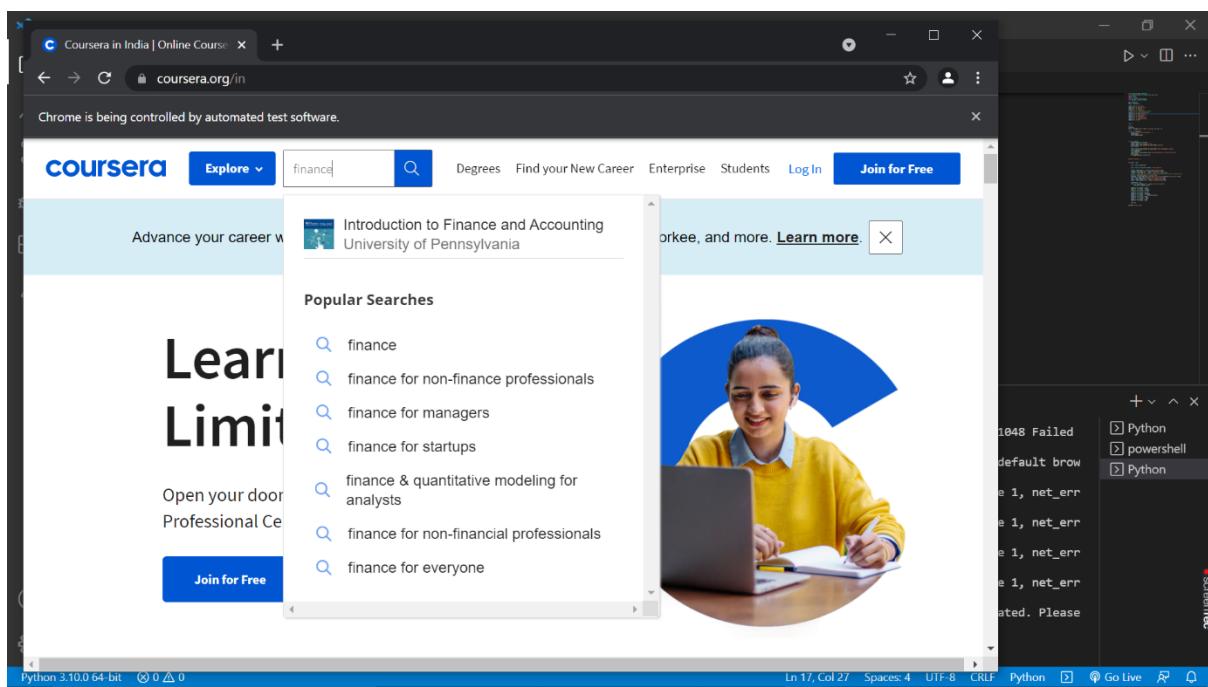
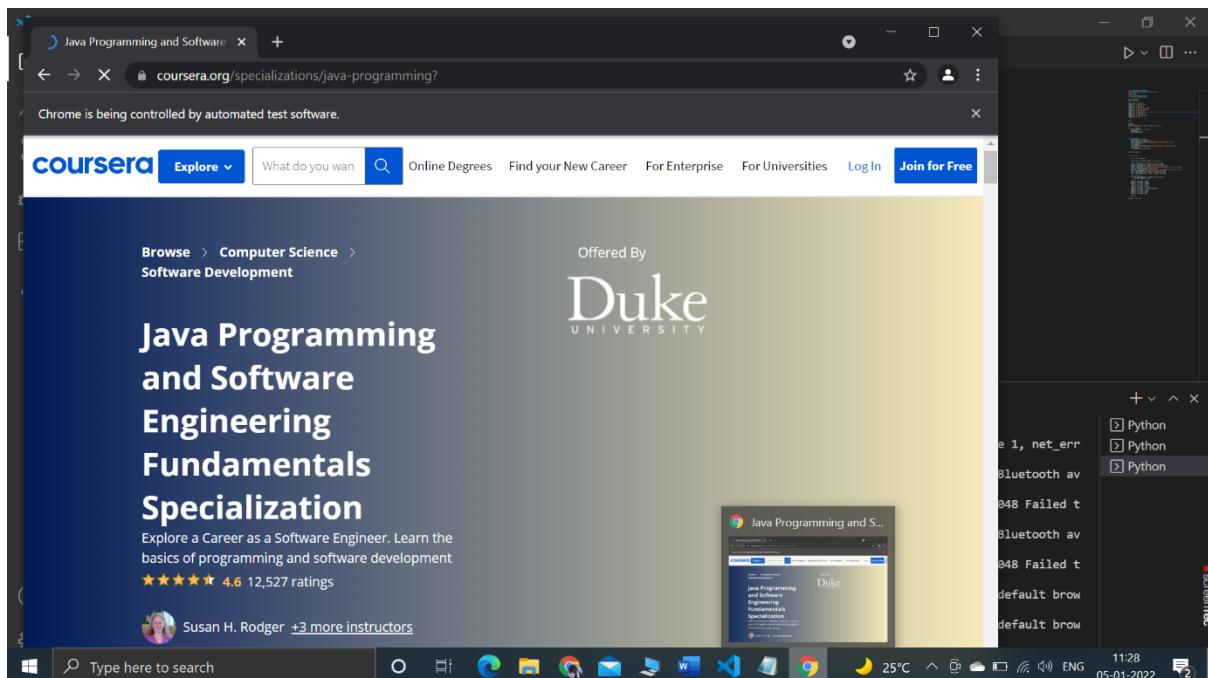
- java
- javascript
- java programming
- javascript for beginners
- java programming for beginners
- java programming and software engineering fundamentals
- java for android

Waiting for tags.rd.linksynergy.com...

Python 3.10.0 64-bit 0 0 0

Ln 17, Col 27 Spaces: 4 UTRF Python Go Live





C Introduction to Finance and Acc... +

coursera.org/specializations/finance-accounting?

Chrome is being controlled by automated test software.

coursera Explore What do you want? Degrees Find your New Career Enterprise Students Log In Join for Free

Browse > Business > Finance Offered By Wharton ONLINE

Introduction to Finance and Accounting Specialization

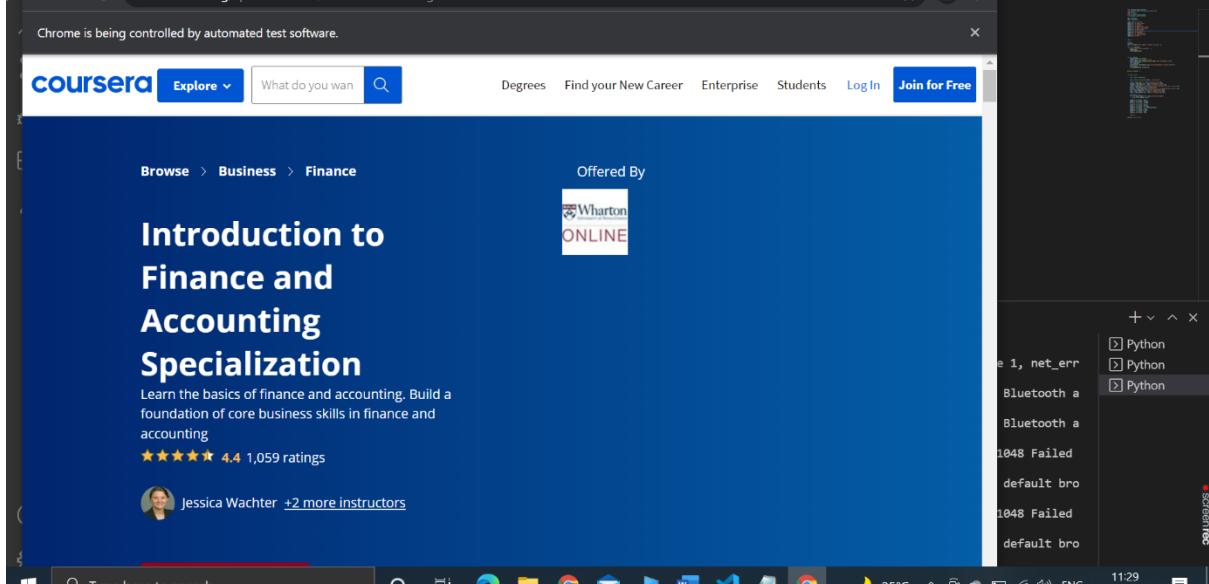
Learn the basics of finance and accounting. Build a foundation of core business skills in finance and accounting

★★★★★ 4.4 1,059 ratings

Jessica Wachter +2 more instructors

Type here to search

Windows 11: 25°C ENG 05-01-2022 11:29



C Coursera in India | Online Course +

coursera.org/in

Chrome is being controlled by automated test software.

coursera Explore marketing Degrees Find your New Career Enterprise Students Log In Join for Free

Advance your career with Coursera, and more. Learn more.

Learn Without Limits

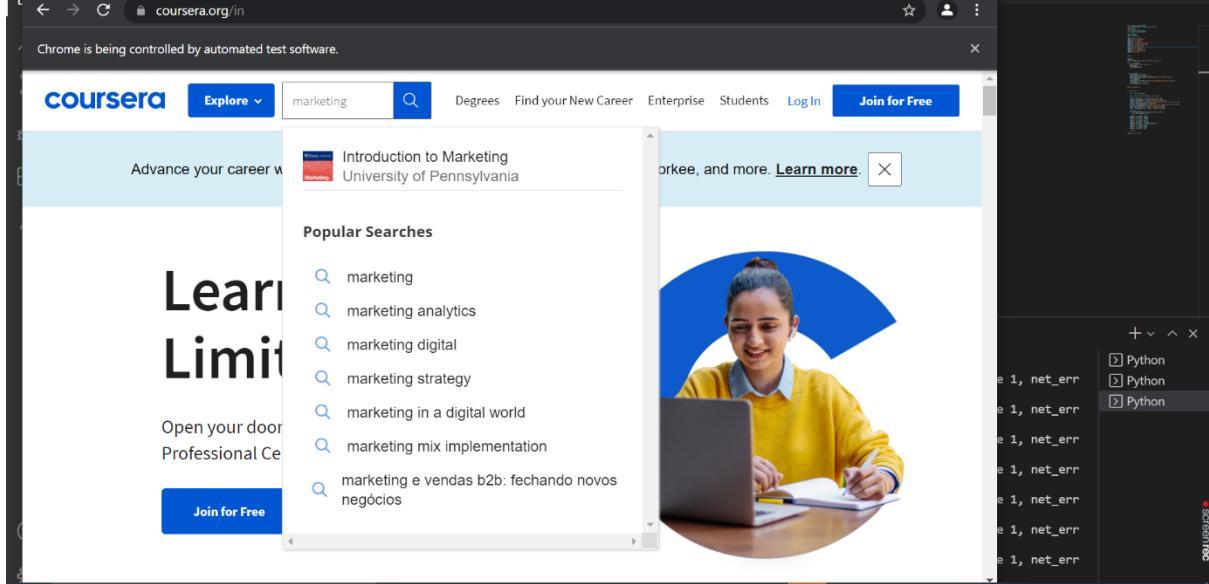
Open your doors to professional opportunities

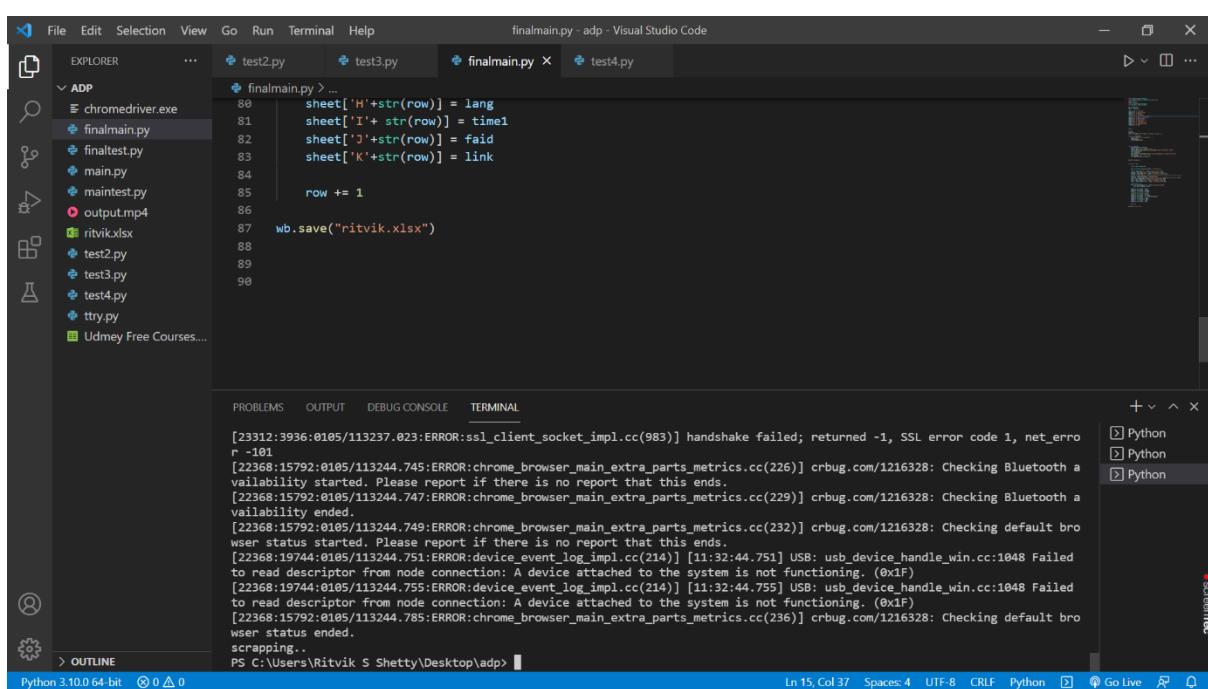
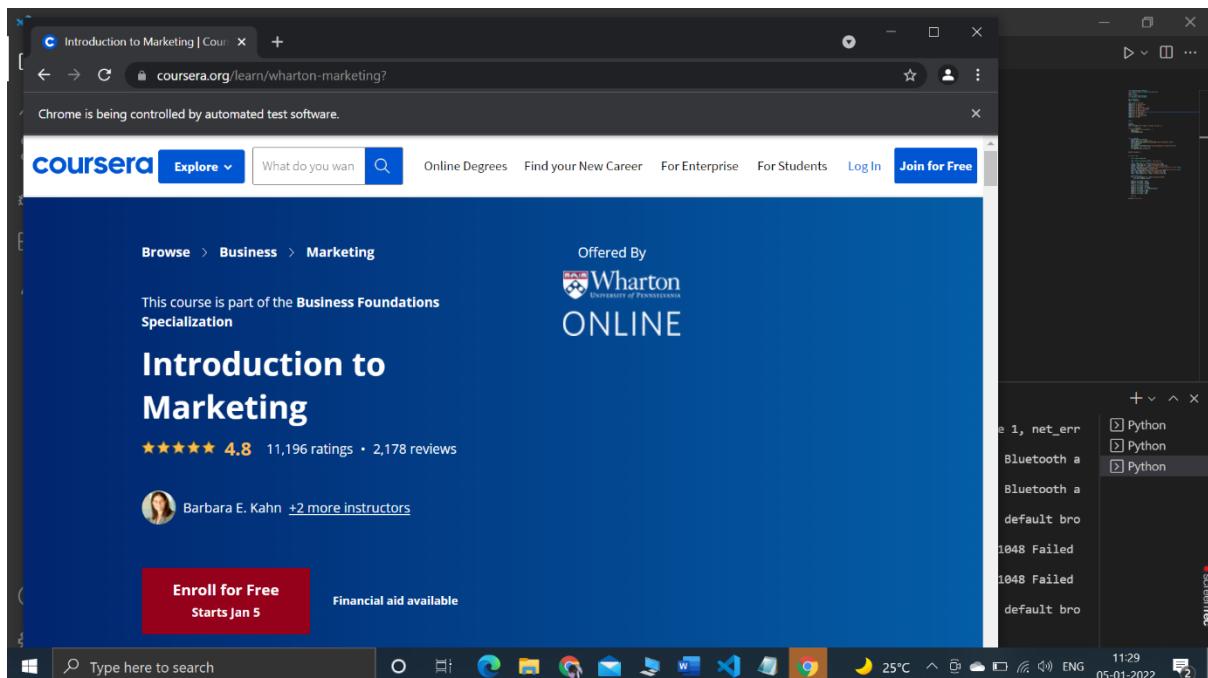
Join for Free

marketing marketing analytics marketing digital marketing strategy marketing in a digital world marketing mix implementation marketing e vendas b2b: fechando novos negócios

Type here to search

Windows 11: 25°C ENG 05-01-2022 11:29





Problem Statement-2:

The Python program below demonstrates a graphical user interface which can perform the below four functions:

- 1)Merging PDF** : Merging PDF helps users to make things less complicated. An user can merge PDFs online but uploading crucial documents to websites is not safe. This GUI will take any number of PDFs from user, merge it and store it in the location specified by user.
- 2)Word Document** : If a user wants to convert any text to word he can do so by entering the text in the field. This interface will convert the text from user and convert it to word document and saves it in the location specified by the user.
- 3)Selenium** : When supplied with user details through a CSV file this feature will automate the working of the popular social media instagram for the user.
- 4)CSV to JSON converter** : This feature will convert a CSV(Comma Separated Values) file to JSON file.

Source Code:

```
import tkinter as tk

from tkinter import *

from tkinter import filedialog

import pyautogui as pgui

import PyPDF2

import docx

import csv

import json

from selenium import webdriver

from selenium.webdriver.common.keys import Keys

from selenium.common.exceptions import NoSuchElementException

import time

pdfmrg=[] #list to store the path of all the selected files

pdfWriter=PyPDF2.PdfFileWriter()

n=0 #keeps the count of pdf

count=1 #used as a counter

while True:

    user=pgui.prompt("Enter your choice\n 1.Merge PDF\n2.Word Document\n3.Selenium\n4.CSV to JSON") #asks the user for choice

    if user=='1' or user=='2' or user=='3'or user=='4' or user=='5': #exception handling if user enters invalid choice

        break

    else:

        pgui.alert("Enter the correct choice")
```

```
if user=='1': # for merging pdf

    def pdffunc():

        frame=Tk()

        frame.geometry('550x200') #set the dimensions of the window

        e=Entry(frame,width=25) #create entry box

        e.grid(row=0,column=2) #position of the entry box in the window

        nlab=Label(frame,text="Enter the number of pdf to be merged") #asks
user input for number of pdfs

        nlab.grid(row=0,column=0)

        def click():

            global n

            n=e.get() #extract the data entered in the field box

            frame.destroy() #destroys the window displayed

            butn=Button(frame,text='SUBMIT',command=click)

            butn.grid(row=5,column=3)

            frame.mainloop() #keeps running the window

        pdffunc()

while(count==1):

    try: #checks if user has entered integer value or not

        n=int(n)

        count=0

    except:

        pgui.alert("Invalid input:Please Enter a Number")

        pdffunc()
```

```
#asks users the location of the pdf
pgui.alert("Please select the PDF Files in sequence")

for i in range(n):
    mrg=filedialog.askopenfilename() #selects the file paths
    pdfmrg.append(mrg) #stores the file paths in the list

#asks users the name of the file to save
pgui.alert("Please select the Name and Location for the merged PDF files")
ufname=filedialog.asksaveasfilename()

#loops through all the pdf
for filename in pdfmrg:
    pdfobj=open(filename, 'rb')
    pdfReader=PyPDF2.PdfFileReader(pdfobj)
    #opens each page of the specified pdf
    for pgnum in range(pdfReader.numPages):
        pgObj=pdfReader.getPage(pgnum)
        pdfWriter.addPage(pgObj)

#saves pdf to file
pdfOutput=open(ufname+'.pdf', 'wb')
#outputting pdf
pdfWriter.write(pdfOutput)
#closes the pdf writer
pdfOutput.close()

pgui.alert("PDF Merging is Complete!!!!")
```

```
#for word document

if user=='2':

    frame=TK()

    doc=docx.Document()

    pgui.alert("Please select the Name and Location of the word file,you want
to create")

    fn=filedialog.asksaveasfilename() #selects the path of the file to save
as

    frame.destroy() #destroys the window displayed


def word():

    frame=TK()

    frame.geometry('550x300') #sets the size of the window

    e=Entry(frame,width=50) #creates entry box

    e.grid(row=0,column=2)

    nlab=Label(frame,text="Enter the text!:") #takes the input from user

    nlab.grid(row=0,column=0)


    def click():

        global n

        n=e.get()

        doc.add_paragraph(n) #adds the paragraph to the word document

        doc.save(fn+'.docx') #saves the file,whenever a new paragraph is
created and when the data is entered

        frame.withdraw()

        frame.update()

        frame.destroy() #destroys the window displayed


    btn=Button(frame,text='submit',command=click)
```

```
btn.grid(row=4,column=2)

frame.mainloop()

word()

while(True):

    dx=pgui.confirm("Do you wish to add another Paragraph??") #asks the
user if he wishes to add a new paragraph to the document

    if dx=='OK':

        word()

    else:

        pgui.alert("Process is completed!!!")

        break

#for selenium

if user=='3':


    #function where instagram website opens and logins automatically

    def insta():

        path="C:\chromedriver.exe" #chromedriver path to automate the tasks

        url="https://www.instagram.com/" #instagram url to be visited and to
perform action

        browser=webdriver.Chrome(path)

        browser.get(url) #response object is returned

        print(browser.title)

        with open('C:\\Users\\Divya
Pai\\OneDrive\\Documents\\instagram.csv','r') as csv_file: #opens the csv file
which contains username and password to login to instagram

            csvReader=csv.reader(csv_file) #reads the contents of csv file

            next(csvReader) #starts reading from the second row

            for x in csvReader:
```

```
try: #try exception statements which avoids exceptions caused
during the execution of the program

    m=webdriver.find_element_by_css_selector('#loginForm > div >
div:nth-child(1) > div > label > input') #finds the element in the html page
of the url

    m.send_keys(Keys.CONTROL+"a")

    m.send_keys(Keys.DELETE) #to clear the input in case of
predefined filler text

    m.send_keys(x[0]) #grabs the contents from csv file

    pswd=webdriver.find_element_by_name('password') #finds the
element in the html page of the url

    pswd.send_keys(Keys.CONTROL+"a")

    pswd.send_keys(Keys.DELETE) #to clear the input in case
of predefined filler text

    pswd.send_keys(x[1]) #grabs the contents from csv file

    log=webdriver.find_element_by_css_selector('#loginForm >
div.qF0y9.Igw0E.IwRSH.eGOV_.4EzTm.kEKum > div:nth-child(3)')

    log.submit() #clicks the login button

except NoSuchElementException: #to avoid the exception of not
finding the specific element

    pass

    time.sleep(2) #sleeps or pauses the screen for a second

    if url.endswith('browse'): #if the url ends with browser
address path the for loop terminates and ends the task

        break

    time.sleep(20)

insta()

if user=='4':
```

```
# Function to convert a CSV to JSON

# Takes the file paths as arguments

def make_json(csvFilePath, jsonFilePath):

    # create a dictionary
    data = {}

    # Open a csv reader called DictReader
    with open(csvFilePath) as csvf:

        csvReader = csv.DictReader(csvf)

        # Convert each row into a dictionary
        # and add it to data
        for rows in csvReader:

            # Assuming a column named 'No' to
            # be the primary key
            key = rows['No']
            data[key] = rows

    # Open a json writer, and use the json.dumps()
    # function to dump data
    with open(jsonFilePath, 'w') as jsonf:

        jsonf.write(json.dumps(data, indent=4))

# Decide the two file paths according to your
# computer system

csvFilePath = r'Names.csv'
```

```

jsonFilePath = r'Names.json'

# Call the make_json function

make_json(csvFilePath, jsonFilePath)

pgui.alert("Conversion completed")

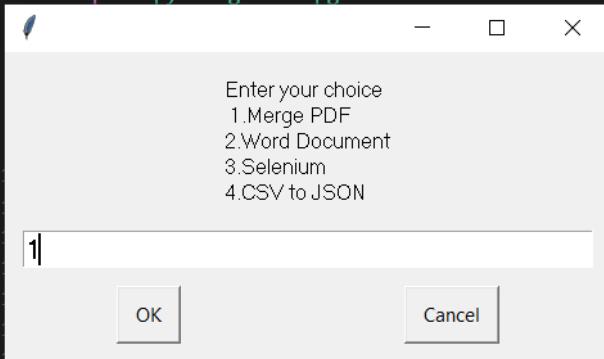
```

Output:

D: > photos > New folder > python > adpproject.py > insta

```

1  import tkinter as tk
2  from tkinter import *
3  from tkinter import filedialog
4  import pyautogui as pgui


5  Enter your choice
6  1.Merge PDF
7  2.Word Document
8  3.Selenium
9  4.CSV to JSON
10
11
12
13
14
15
16
17  n=0 #keeps the count of pdf
18  count=1 #used as a counter
19
20  while True:
21      user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON")
22      if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling
23          break
24      else:
25          pgui.alert("Enter the correct choice")
26
27  if user=='1': # for merging pdf
28      def pdffunc():
29          frame=Tk()

```

```

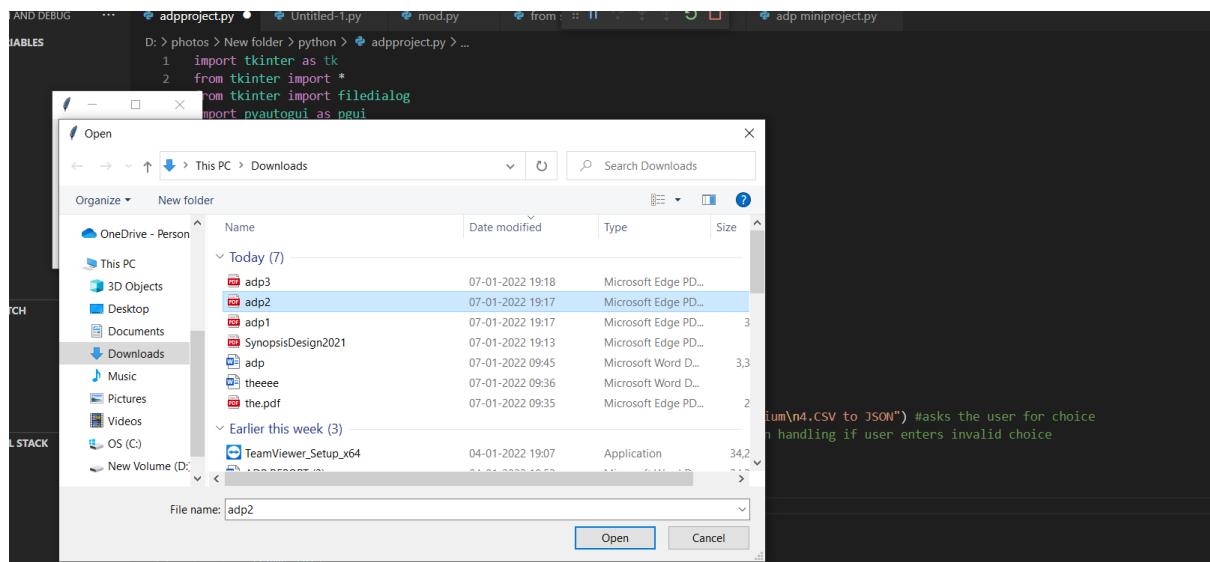
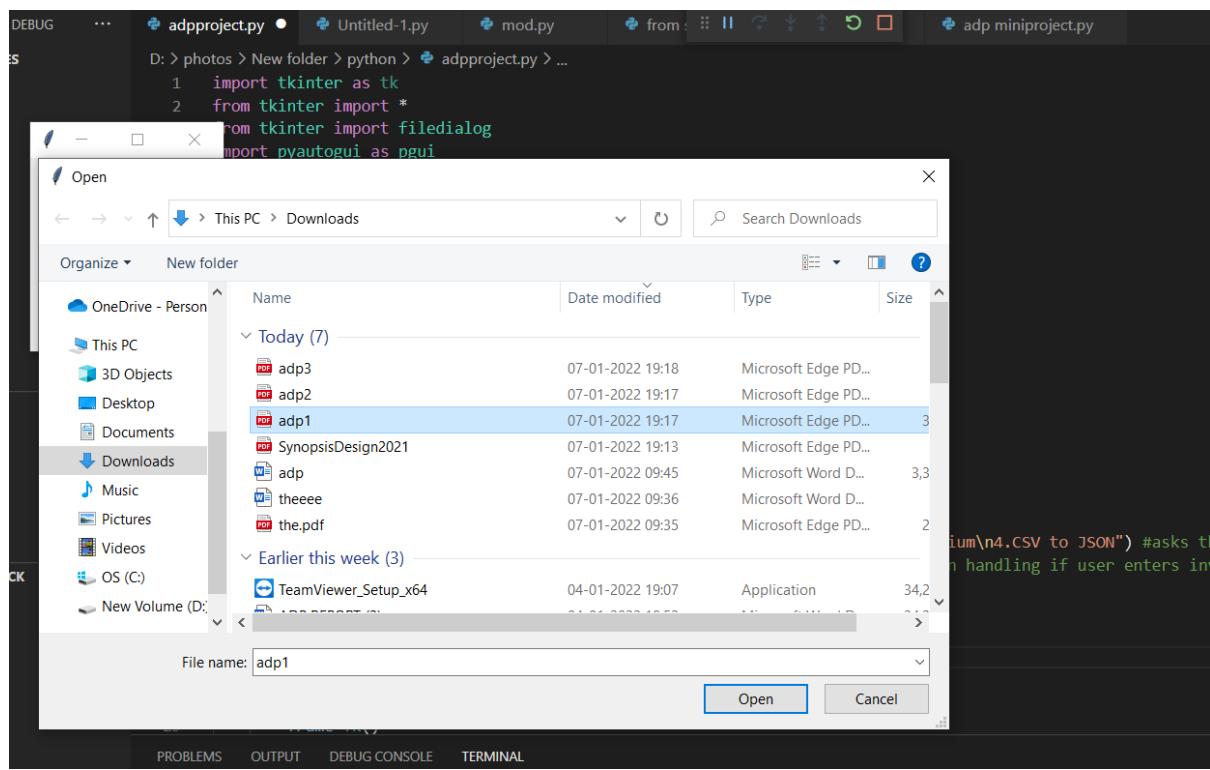
D: > photos > New folder > python > adpproject.py > insta
  1 import tkinter as tk
  2 from tkinter import *
  3 from tkinter import filedialog
  4 import pyautogui as pgui
  5 import PyPDF2
  6
  7 tk
  8 Enter the number of pdf to be merged[2]
  9
 10
 11
 12
 13
 14
 15
 16 pdfWriter=PyPDF2.PdfFileWriter()
 17 n=0 #keeps the count of pdf
 18 count=1 #used as a counter
 19
 20 while True:
 21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON")
 22     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling if user
 23         break
 24     else:
 25         pgui.alert("Enter the correct choice")
 26
 27 if user=='1': # for merging pdf
 28     def pdffunc():
 29         frame=Tk()

```

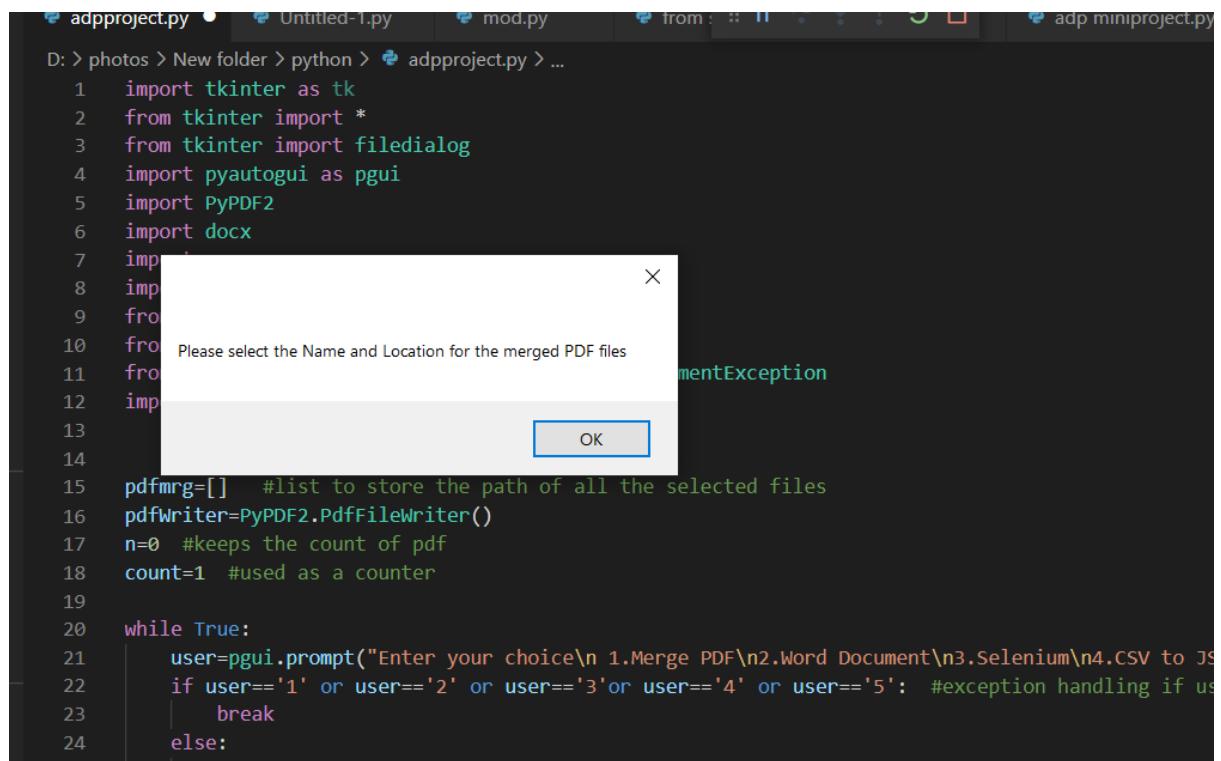
```

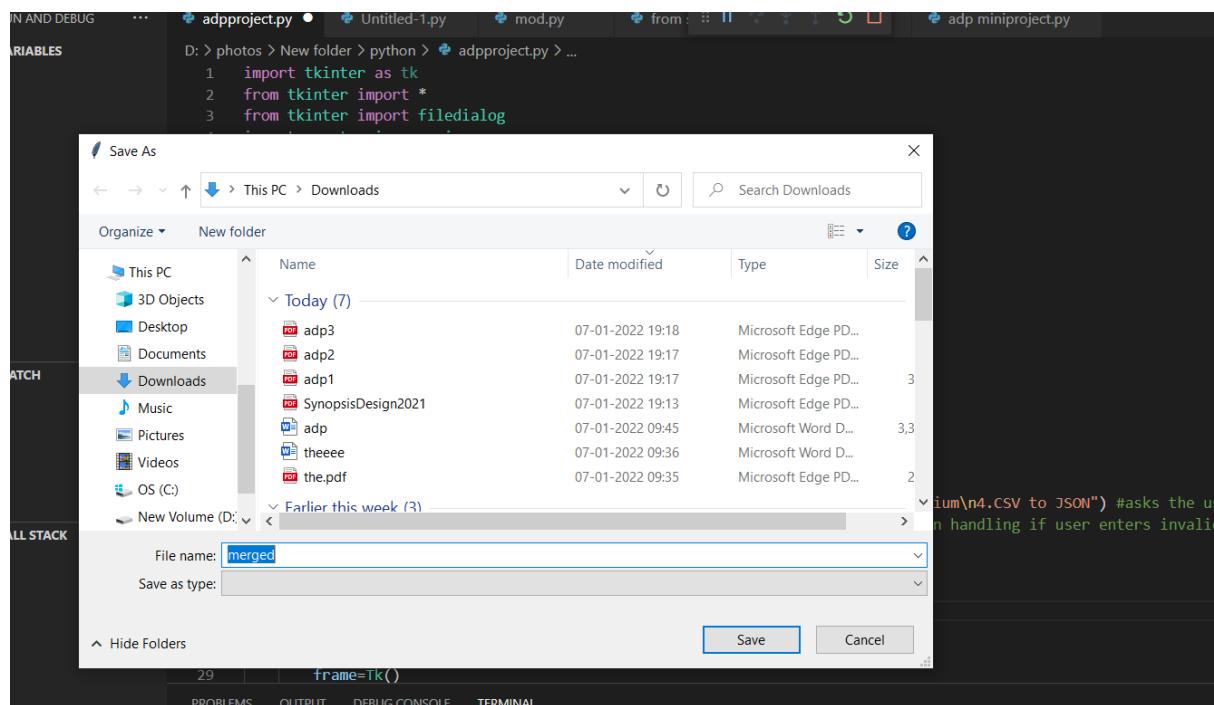
 2 from tkinter import *
 3 from tkinter import filedialog
 4 import pyautogui as pgui
 5 import PyPDF2
 6 import docx
 7 import csv
 8 import json
 9 from selenium import webdriver
10 from selenium.webdriver.common.keys import Keys
11 from selenium.common.exceptions import NoSuchElementException
12 import time,os
13
14 pdfmrg=[]
15 pdfWriter=PyPDF2.PdfFileWriter()
16 n=0 #keeps the count of pdf
17 count=1 #used as a counter
18
19 while True:
20     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON")
21     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling
22         break
23     else:
24         pgui.alert("Enter the correct choice")
25
26 if user=='1': # for merging pdf
27     def pdffunc():
28         frame=Tk()

```



```
adpproject.py • Untitled-1.py mod.py from: :: II . . . . adp miniproject.py
D: > photos > New folder > python > adpproject.py > ...
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import filedialog
4 import pyautogui as pgu
5 import PyPDF2
6 import docx
7 imp[1]
8 imp[2]
9 fro[3]
10 fro[4] Please select the Name and Location for the merged PDF files
11 fro[5]
12 imp[6]
13 [7] OK
14
15 pdfmrg=[] #list to store the path of all the selected files
16 pdfWriter=PyPDF2.PdfFileWriter()
17 n=0 #keeps the count of pdf
18 count=1 #used as a counter
19
20 while True:
21     user=pgu.prompt("Enter your choice\n 1.Merge PDF\n2.Word Document\n3.Selenium\n4.CSV to JS")
22     if user=='1' or user=='2' or user=='3'or user=='4' or user=='5': #exception handling if user enters wrong choice
23         break
24     else:
```





```
D: > photos > New folder > python > adpproject.py > ...
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import filedialog
4 import pyautogui as pgui
5 import PyPDF2
6 import docx
7 import csv
8 import json
9 from selenium
10 from selenium
11 from selenium
12 import time
13
14
15 pdfmrg=[] # for merging pdf
16 pdfWriter=PyPDF2.PdfWriter()
17 n=0 #keeps the count of pdf
18 count=1 #used as a counter
19
20 while True:
21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.Chrome\n 5.Exit")
22     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling
23         break
24     else:
25         pgui.alert("Enter the correct choice")
26
27 if user=='1': # for merging pdf
```

A modal dialog box is displayed in the center of the screen, reporting 'PDF Merging is Complete!!!!'. It has an 'OK' button at the bottom.

merged.pdf

File | C:/Users/Divya%20Pai/Downloads/merged.pdf

1 of 5

Page view | A Read aloud | Add text | Draw | Highlight | Erase | ...

APPLICATION DEVELOPMENT USING PYTHON (Effective from the academic year 2018-2019) SEMESTER – V			
Course Code	18CS55	IA Marks	40
Number of Lecture Hours/Week	03	Exam Marks	60
Total Number of Lecture Hours	40	Exam Hours	03
CREDITS – 03			
Course Objectives: This course (18CS55) will enable students to <ul style="list-style-type: none"> <input type="checkbox"/> Learn the syntax and semantics of Python programming language. <input type="checkbox"/> Illustrate the process of structuring the data using lists, tuples and dictionaries. <input type="checkbox"/> Demonstrate the use of built-in functions to navigate the file system. <input type="checkbox"/> Implement the Object Oriented Programming concepts in Python. <input type="checkbox"/> Appraise the need for working with various documents like Excel, PDF, Word and Others. 			
Module – 1			
Python Basics , Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control, Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys.exit(), Functions, def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number	Teaching Hours	08	
Textbook 1: Chapters 1 – 3			
RBT: L1, L2			
Module – 2			
Lists , The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and	08		

merged.pdf

File | C:/Users/Divya%20Pai/Downloads/merged.pdf

4 of 5

Page view | A Read aloud | Add text | Draw | Highlight | Erase | ...

➤ Demonstrate with example print(), input() and string replication. (6 marks)

➤ Explain elif, for, while, break and continue statements in Python with examples of each (10 marks)

➤ Write a Python program to check whether a given number is even or odd (4 marks)

➤ How can we pass parameters in user defined functions? Explain with suitable examples. (6 marks)

➤ Explain local and global scope with local and global variables (8 marks)

➤ Demonstrate the concept of exception. Implement a code which prompts user for Celsius temperature, convert the temperature to Fahrenheit, and print out the converted temperature by handling the exception. (6 marks)

➤ List the salient features of python programming language. (6 marks)

➤ What are the different flow control statements supports in python. Explain any 3 with an suitable example program and flow chart. (8 marks)

➤ Write a python program to calculate the area of circle, rectangular and triangle. print the results. (6 marks)

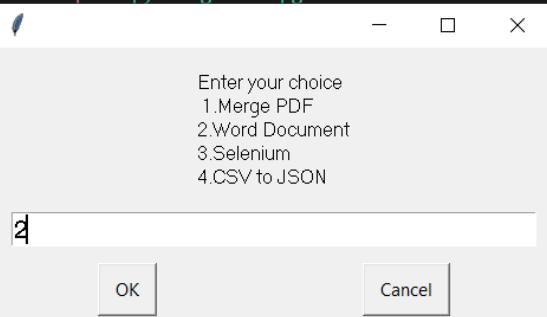
➤ What is a function? How to define a function in python? Write a program using function to find out the given string is palindrome or not. (6 marks)

➤ What is local and global scope of variable in python. Explain the different scenarios with an example snippet. (8 marks)

➤ Write a python program to create a function called collatz() which reads as parameter named number. If the number is even it should print and return number//2 and if the number is odd then it should print and return 3*number+1. The function should keep calling on that number until the function returns a value 1. (6 marks)

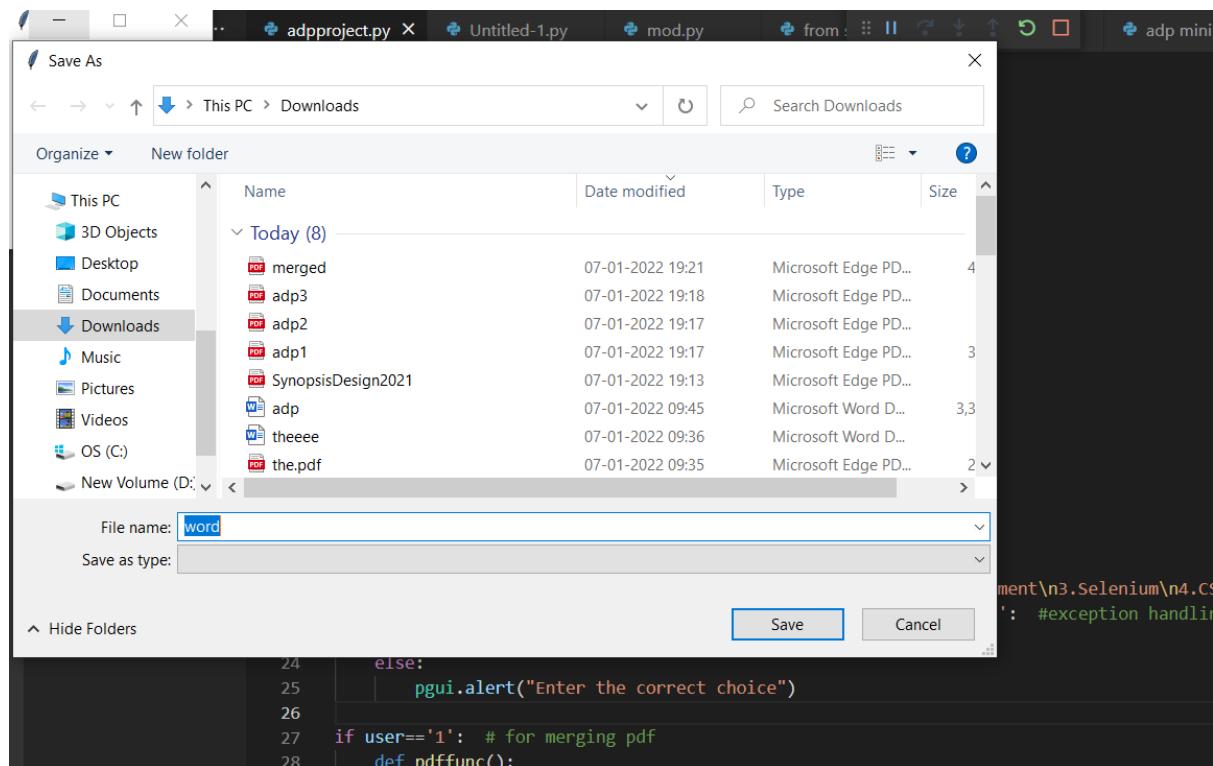
```
JG ... adpproject.py X Untitled-1.py mod.py from: || ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌍ adp min

D: > photos > New folder > python > adpproject.py > ...
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import filedialog
4 import pyautogui as pgu

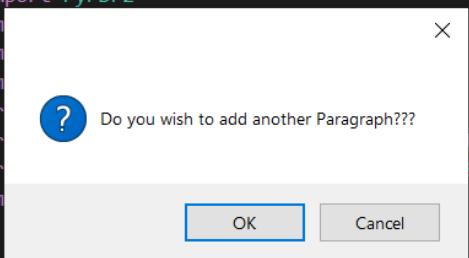

5
6
7
8
9
10
11
12
13
14
15
16
17 n=0 #keeps the count of pdf
18 count=1 #used as a counter
19
20 while True:
21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n2.Word Document\n3.Selenium\n4.CSV to JSON")
22     if user=='1' or user=='2' or user=='3'or user=='4' or user=='5': #exception handling
23         break
24     else:
25         pgu.alert("Enter the correct choice")
26
27 if user=='1': # for merging pdf
28     def pdffunc():

keys
hElementException
he selected files
```

```
D: > photos > New folder > python > adpproject.py > ...
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import filedialog
4 import pyautogui as pgui
5 import PyPDF2
6 import docx
7 import csv
8 import json
9 from selenium import webdriver
10 from selenium.webdriver.common.keys import Keys
11 from selenium.webdriver.common.by import By
12 import time
13
14 pdfmrg=[]
15 pdfWriter=PyPDF2.PdfFileWriter()
16 n=0 #keeps the count of pdf
17 count=1 #used as a counter
18
19 while True:
20     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV")
21     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling
22         break
23     else:
24         pgui.alert("Enter the correct choice")
25
26 if user=='1': # for merging pdf
27     def pdffunc():
28         frame=Tk()
```

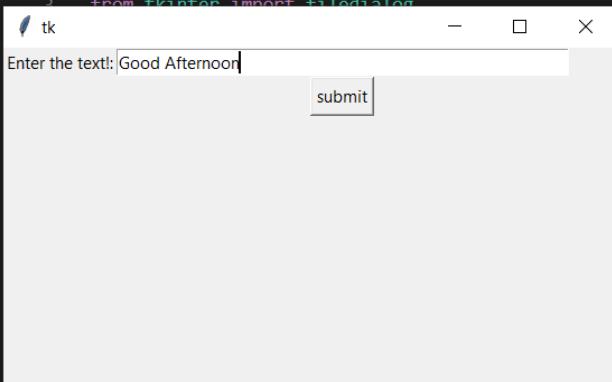



```
D: > photos > New folder > python > adpproject.py > ...
  1 import tkinter as tk
  2 from tkinter import *
  3 from tkinter import filedialog
  4 import pyautogui as pgui
  5 import PyPDF2
  6 im
  7 im
  8 im
  9 fr
  10 fr
  11 fr
  12 im
  13
  14
  15 pdfmrg=[] #list to store the path of all the selected files
  16 pdfWriter=PyPDF2.PdfFileWriter()
  17 n=0 #keeps the count of pdf
  18 count=1 #used as a counter
  19
  20 while True:
  21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON")
  22     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling if user enters wrong choice
  23         break
  24     else:
  25         pgui.alert("Enter the correct choice")
  26
  27 if user=='1': # for merging pdf
  28     def pdffunc():
  29         frame=Tk()
```



```
... adpproject.py X Untitled-1.py mod.py from: PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

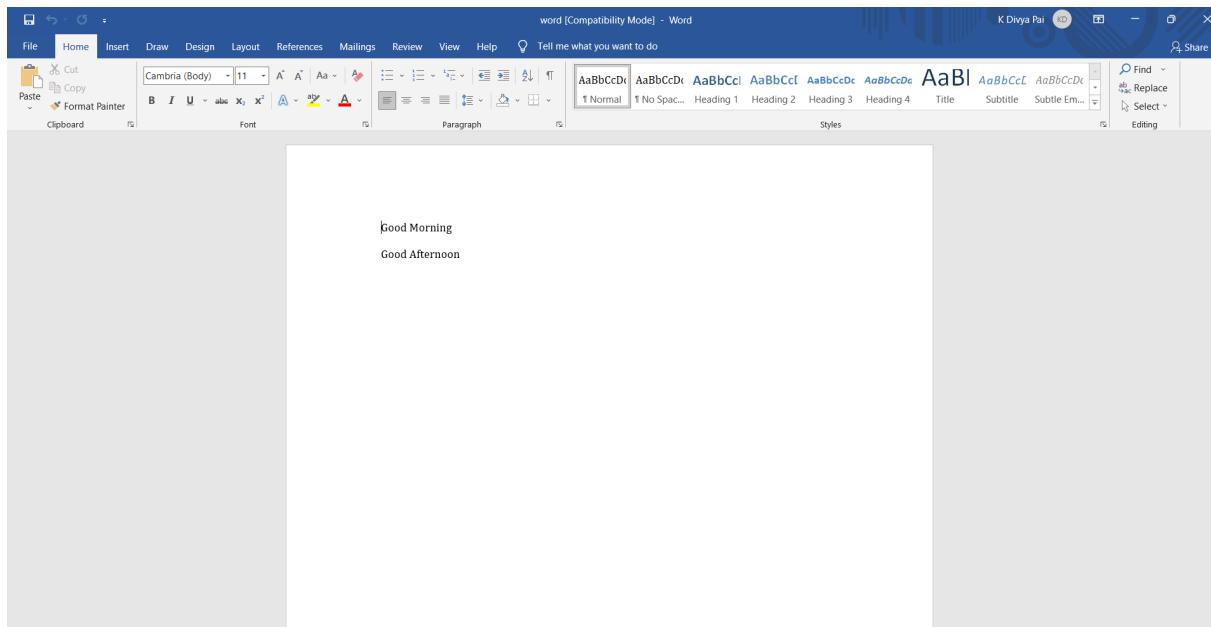
```
D: > photos > New folder > python > adpproject.py > ...
  1 import tkinter as tk
  2 from tkinter import *
  3 from tkinter import filedialog
  4
  5 tk
  6 Enter the text!:Good Afternoon
  7 submit
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18 count=1 #used as a counter
  19
  20 while True:
  21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON")
  22     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling if user enters wrong choice
  23         break
  24     else:
  25         pgui.alert("Enter the correct choice")
  26
  27 if user=='1': # for merging pdf
  28     def pdffunc():
  29         frame=Tk()
```



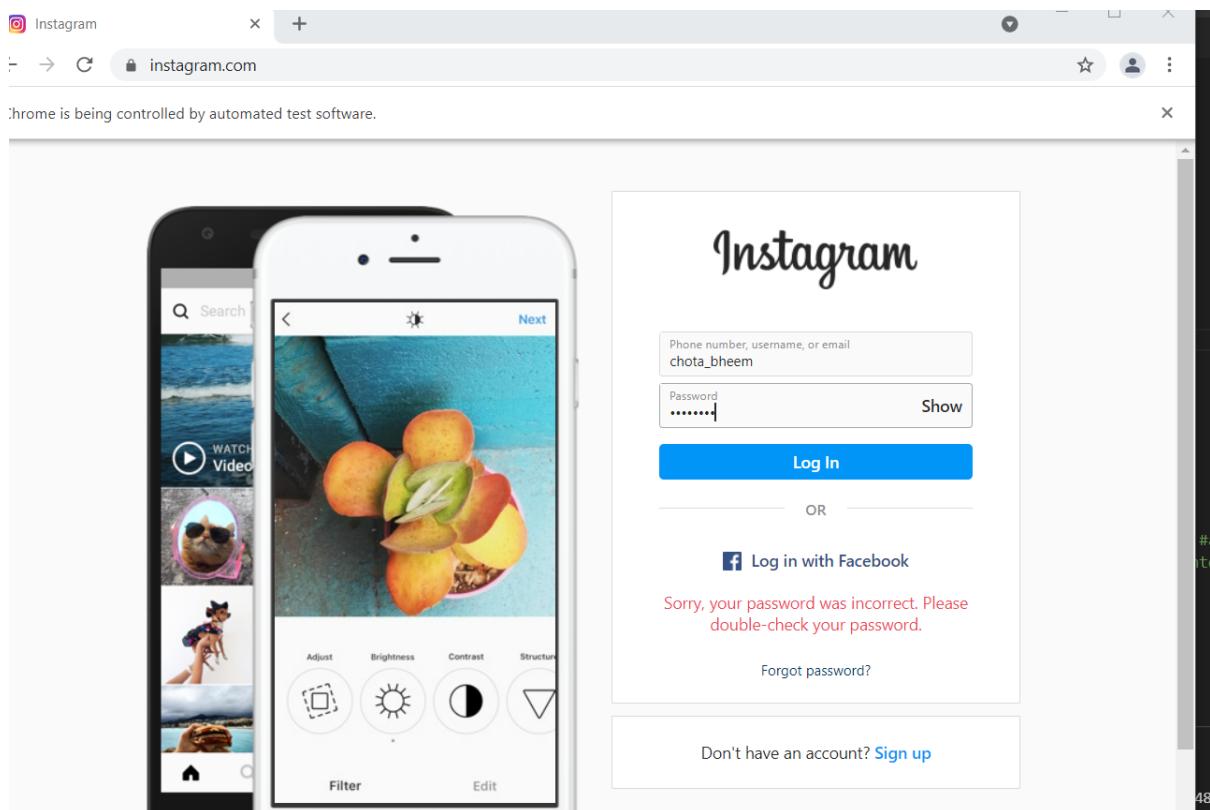
```

D: > photos > New folder > python > adpproject.py > ...
1 import tkinter as tk
2 from tkinter import *
3 from tkinter import filedialog
4 import pyautogui as pgui
5 import PyPDF2
6 import docx
7 import csv
8 import time
9 from selenium import webdriver
10 from selenium.webdriver.common.keys import Keys
11 from selenium.common.exceptions import NoSuchElementException
12 import os
13
14
15 pdfList = [] #list to store the path of all the selected files
16 pdfWriter=PyPDF2.PdfFileWriter()
17 n=0 #keeps the count of pdf
18 count=1 #used as a counter
19
20 while True:
21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n2.Word Document\n3.Selenium\n4.CSV\n5.Exit")
22     if user=='1' or user=='2' or user=='3' or user=='4' or user=='5': #exception handling
23         break
24     else:
25         pgui.alert("Enter the correct choice")
26
27 if user=='1': # for merging pdf
28     def pdfffunc():
29         frame=Tk()

```



```
D:\ > photos > New folder > python > adpproject.py ...  
1 import tkinter as tk  
2 from tkinter import *  
3 from tkinter import filedialog  
4 import pyautogui as pgui  
  
Enter your choice  
1.Merge PDF  
2.Word Document  
3.Selenium  
4.CSV to JSON  
3  
OK Cancel  
the selected files  
16 parameter(sys.argv[1])  
17 n=0 #keeps the count of pdf  
18 count=1 #used as a counter  
19  
20 while True:  
21     user=pgui.prompt("Enter your choice\n 1.Merge PDF\n 2.Word Document\n 3.Selenium\n 4.CSV to JSON") #a  
22     if user=='1' or user=='2' or user=='3'or user=='4' or user=='5': #exception handling if user ente  
23         break  
24     else:  
25         pgui.alert("Enter the correct choice")  
26  
27 if user=='1': # for merging pdf  
28     def pdffunc():  
29         frame=tk()  
  
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
```



Instagram x

← → × instagram.com ⋮

Chrome is being controlled by automated test software.

Instagram Search Home Logout

Now you can create and share posts directly from your computer.

Getting Started

Find Facebook Friends
You choose which friends to follow.
We'll never post to Facebook without your permission.

Add Profile Photo
Add a profile photo so your friends know it's you.

[Connect to Facebook](#) [Add Profile Photo](#)

 **_charliechap** ...

charliechap charlie Chaplin [Switch](#)

Suggestions For You [See All](#)

Profile Picture	Username	Status	Action
	beyonce	Popular	Follow
	kyliejenner	Popular	Follow
	therock	Popular	Follow
	sudharanigovardhan	Popular	Follow
	selenagomez	Popular	Follow

About · Help · Press · API · Jobs · Privacy · Terms · Locations · Top Accounts · Hashtags · Language

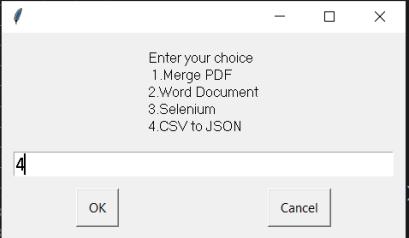
© 2022 INSTAGRAM FROM META

Type here to search Windows Start File Explorer Edge PDF Dell Google Chrome Microsoft Edge Windows Update Google Chrome

28°C Haze ⌂ ⌂ ⌂ ENG IN 19:37 07-01-2022

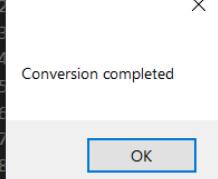
File Go Run Terminal Help adpproject.py - Untitled-1.py mod.py adp n

```
D: > photos > New folder > python > adpproject.py > ...
66     pdfobj=open(filename, 'rb')
67     pdfReader=PyPDF2.PdfFileReader(pdfobj)
68     #opens each page of the specified pdf
69     for pgnum in range(pdfReader.numPages):
70
71         pgObj=pdfReader.getPage(pgnum)
72         pdfWriter.addPage(pgObj)
73
74         pdfWriter.write(ufname+'.pdf', 'wb')
75
76         pdfWriter.close()
77
78     pgui.alert("PDF Merging is Complete!!!!")
79
80
81 #for word document
82 if user=='2':
83     frame=Tk()
84     doc=docx.Document()
85     pgui.alert("Please select the Name and Location of the word file,you want to create")
86     fn=filedialog.asksaveasfilename() #selects the path of the file to save as
87     frame.destroy() #destroys the window displayed
88
89 def word():
90     frame=Tk()
91     frame.geometry('550x300') #sets the size of the window
92     e=Entry(frame,width=50) #creates entry box
93     e.grid(row=0,column=2)
94     lab=Label(frame,text="Enter the text!").grid(row=0,column=0)
```



File Go Run Terminal Help adpproject.py - Untitled-1.py mod.py adp n

```
D: > photos > New folder > python > adpproject.py > ...
66     pdfobj=open(filename, 'rb')
67     pdfReader=PyPDF2.PdfFileReader(pdfobj)
68     #opens each page of the specified pdf
69     for pgnum in range(pdfReader.numPages):
70         pgObj=pdfReader.getPage(pgnum)
71         pdfWriter.addPage(pgObj)
72
73         pdfWriter.write(ufname+'.pdf', 'wb')
74
75         pdfWriter.close()
76
77     pgui.alert("PDF Merging is Complete!!!!")
78
79
80 #for word document
81 if user=='2':
82     frame=Tk()
83     doc=docx.Document()
84     pgui.alert("Please select the Name and Location of the word file,you want to create")
85     fn=filedialog.asksaveasfilename() #selects the path of the file to save as
86     frame.destroy() #destroys the window displayed
87
88 def word():
89     frame=Tk()
90     frame.geometry('550x300') #sets the size of the window
91     e=Entry(frame,width=50) #creates entry box
92     e.grid(row=0,column=2)
```



The screenshot shows a code editor interface with a dark theme. At the top, there are tabs for 'adpproject.py', 'Names.json', 'Untitled-1.py', 'mod.py', and 'adp miniproject.py'. The current file is 'Names.json', which contains the following JSON data:

```
1  [
2   "1": {
3     "No": "1",
4     "username": "python_app",
5     "password": "q1w2e3r4@"
6   },
7   "2": {
8     "No": "2",
9     "username": "thug_life",
10    "password": "rfed2321?"
11  },
12  "3": {
13    "No": "3",
14    "username": "mr_beans",
15    "password": "teddy@123"
16  },
17  "4": {
18    "No": "4",
19    "username": "chota_bheem",
20    "password": "raju^789"
21  },
22  "5": {
23    "No": "5",
24    "username": "_charliechap",
25    "password": "Popstar89"
26  },
27  "6": {
28    "No": "6",
29    "username": "arya_prasad",
```

Below the code editor, there are four tabs: 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'.