**AMITY INTERNATIONAL SCHOOL NOIDA**

**COMPUTER SCIENCE PROJECT**

Accounts-manager



By:

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class: \_\_\_\_\_\_\_XII\_\_\_\_\_\_

Roll Number: \_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

*I acknowledge the valuable contribution of Mrs \_\_\_\_\_\_\_\_\_\_\_in providing me the proper guidance to complete the computer science project. The project would not have been completed without her support and kind help.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_(name of student)*

CERTIFICATE

**AMITY INTERNATIONAL SCHOOL, NOIDA**

*This is to certify that \_\_\_\_\_\_\_\_\_\_ Roll number: \_\_\_\_\_\_\_\_ of Class XII has successfully completed the \_\_\_Computer Science Project: “Account Manager” under my supervision according to the guidelines laid down by CBSE.*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Ms. Kirti Tripathi*

CONTENTS

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **DESCRIPTION** | **PAGE NO.** |
| 1 | Acknowledgement |  |
| 2 | Certificate |  |
| 3 | Introduction |  |
| 4 | Source Code |  |
| 5 | Output & Screenshots |  |
| 6 | Conclusion & Future Expansion |  |
| 7 | User Manual |  |
| 8 | Bibliography |  |

INTRODUCTION

Our code is aimed at making the life of a modern social media enthusiast or practically anyone who uses the internet easier. Our project is to design a program which asks a user for their online account credentials and then when submitted, logs in to their Gmail account.

Our project uses a Tkinter interface to ask the user for their name, email and password for their online accounts (Gmail in this case) and use selenium module to open the gmail website

REQUIREMENTS

The game requires the following hardware –

* A Keyboard (To provide the necessary controls)
* A Mouse / Track-pad (To browse through menus)

The game requires the following software –

* OS – Microsoft Windows 98/XP/7
* RAM – 500 MB or greater
* Processor – 1 GHz Intel compatible CPU
* Hard disk space – 3 MB or greater

WORKING OF THE CODE

SOURCE CODE

from tkinter import \*

from tkinter import messagebox

import tkinter.messagebox

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import getpass

# \*\*\*\*\*\* GLOBAL VARIABLES \*\*\*\*\*\*

objects = []

window = Tk()

window.withdraw()

window.title('Email Storage')

emailAuto = StringVar()

passwordAuto = StringVar()

class popupWindow(object):

loop = False

attempts = 0

def \_\_init\_\_(self, master):

top = self.top = Toplevel(master)

top.title('Input Password')

top.geometry('{}x{}'.format(250, 100))

top.resizable(width=False, height=False)

self.l = Label(top, text=" Password: ", font=('Courier', 14), justify=CENTER)

self.l.pack()

self.e = Entry(top, show='\*', width=30)

self.e.pack(pady=7)

self.b = Button(top, text='Submit', command=self.cleanup, font=('Courier', 14))

self.b.pack()

def cleanup(self):

self.value = self.e.get()

access = ''

if self.value == access:

self.loop = True

self.top.destroy()

window.deiconify()

else:

self.attempts += 1

if self.attempts == 5:

window.quit()

self.e .delete(0, 'end')

messagebox.showerror('Incorrect Password', 'Incorrect password, attempts remaining: ' + str(5 - self.attempts))

class entity\_add:

def \_\_init\_\_(self, master, n, p, e):

self.password = p

self.name = n

self.email = e

self.window = master

def write(self):

f = open('data.txt', "a")

n = self.name

e = self.email

p = self.password

encryptedN = ""

encryptedE = ""

encryptedP = ""

for letter in n:

if letter == ' ':

encryptedN += ' '

else:

encryptedN += chr(ord(letter) + 5)

for letter in e:

if letter == ' ':

encryptedE += ' '

else:

encryptedE += chr(ord(letter) + 5)

for letter in p:

if letter == ' ':

encryptedP += ' '

else:

encryptedP += chr(ord(letter) + 5)

f.write(encryptedN + ',' + encryptedE + ',' + encryptedP + ', \n')

f.close()

class entity\_display:

def \_\_init\_\_(self, master, n, e, p, i):

self.password = p

self.name = n

self.email = e

self.window = master

self.i = i

dencryptedN = ""

dencryptedE = ""

dencryptedP = ""

for letter in self.name:

if letter == ' ':

dencryptedN += ' '

else:

dencryptedN += chr(ord(letter) - 5)

for letter in self.email:

if letter == ' ':

dencryptedE += ' '

else:

dencryptedE += chr(ord(letter) - 5)

for letter in self.password:

if letter == ' ':

dencryptedP += ' '

else:

dencryptedP += chr(ord(letter) - 5)

self.label\_name = Label(self.window, text=dencryptedN, font=('Courier', 14))

self.label\_email = Label(self.window, text=dencryptedE, font=('Courier', 14))

self.label\_pass = Label(self.window, text=dencryptedP, font=('Courier', 14))

self.deleteButton = Button(self.window, text='X', fg='red', command=self.delete)

def display(self):

self.label\_name.grid(row=6 + self.i, sticky=W)

self.label\_email.grid(row=6 + self.i, column=1)

self.label\_pass.grid(row=6 + self.i, column=2, sticky=E)

self.deleteButton.grid(row=6 + self.i, column=3, sticky=E)

def delete(self):

answer = tkinter.messagebox.askquestion('Delete', 'Are you sure you want to delete this entry?')

if answer == 'yes':

for i in objects:

i.destroy()

f = open('data.txt', 'r')

lines = f.readlines()

f.close()

f = open('data.txt', "w")

count = 0

for line in lines:

if count != self.i:

f.write(line)

count += 1

f.close()

readfile()

def destroy(self):

self.label\_name.destroy()

self.label\_email.destroy()

self.label\_pass.destroy()

self.deleteButton.destroy()

# \*\*\*\*\*\*\* FUNCTIONS \*\*\*\*\*\*\*\*\*

def onsubmit():

m1 = email.get()

p1 = password.get()

n1 = name.get()

e = entity\_add(window, n1, p1, m1)

e.write()

name.delete(0, 'end')

email.delete(0, 'end')

password.delete(0, 'end')

messagebox.showinfo('Added Entity', 'Successfully Added, \n' + 'Name: ' + n1 + '\nEmail: ' + m1 + '\nPassword: ' + p1)

readfile()

def automate():

val1 = emailAuto.get()

val2 = passwordAuto.get()

print(val1)

print(val2)

# val1 = emailAuto.get()

# val2 = passwordAuto.get()

# usernameStr = input("> ")

# passwordStr = getpass.getpass(prompt="> ")

# val1 = emailAuto.get()

# val2 = passwordAuto.get()

browser = webdriver.Chrome()

browser.get(('https://accounts.google.com/ServiceLogin?'

'service=mail&continue=https://mail.google'

'.com/mail/#identifier'))

# fill in username and hit the next button

username = browser.find\_element\_by\_id('identifierId')

username.send\_keys(val1)

nextButton = browser.find\_element\_by\_id('identifierNext')

nextButton.click()

# wait for transition then continue to fill items

password = WebDriverWait(browser, 10).until(

EC.presence\_of\_element\_located((By.NAME, "password")))

password.send\_keys(val2)

signInButton = browser.find\_element\_by\_id('passwordNext')

signInButton.click()

def clearfile():

f = open('data.txt', "w")

f.close()

def readfile():

f = open('data.txt', 'r')

count = 0

for line in f:

entityList = line.split(',')

e = entity\_display(window, entityList[0], entityList[1], entityList[2], count)

objects.append(e)

e.display()

count += 1

f.close()

# \*\*\*\*\*\*\* GRAPHICS \*\*\*\*\*\*\*\*\*

popupwindow = popupWindow(window)

entity\_label = Label(window, text='Storage Vault', font=('Courier', 18))

name\_label = Label(window, text='Name: ', font=('Courier', 14))

email\_label = Label(window, text='Email: ', font=('Courier', 14))

pass\_label = Label(window, text='Password: ', font=('Courier', 14))

name = Entry(window, font=('Courier', 14))

email = Entry(window,textvariable = emailAuto, font=('Courier', 14))

password = Entry(window,textvariable = passwordAuto, show='\*', font=('Courier', 14))

submit2 = Button(window, text="Open Browser", command = automate, font=('Courier', 14))

submit = Button(window, text='Add Email', command=onsubmit, font=('Courier', 14))

entity\_label.grid(columnspan=3, row=0)

name\_label.grid(row=1, sticky=E, padx=3)

email\_label.grid(row=2, sticky=E, padx=3)

pass\_label.grid(row=3, sticky=E, padx=3)

name.grid(columnspan=3, row=1, column=1, padx=2, pady=2, sticky=W)

email.grid(columnspan=3, row=2, column=1, padx=2, pady=2, sticky=W)

password.grid(columnspan=3, row=3, column=1, padx=2, pady=2, sticky=W)

submit.grid(columnspan=3, pady=4, sticky = W)

submit2.grid(columnspan=3, row =4, padx = 2, pady=2, sticky=E)

name\_label2 = Label(window, text='Name: ', font=('Courier', 14))

email\_label2 = Label(window, text='Email: ', font=('Courier', 14))

pass\_label2 = Label(window, text='Password: ', font=('Courier', 14))

name\_label2.grid(row=5)

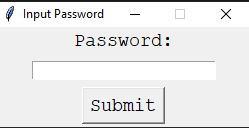
email\_label2.grid(row=5, column=1)

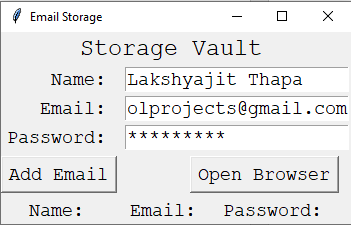
pass\_label2.grid(row=5, column=2)

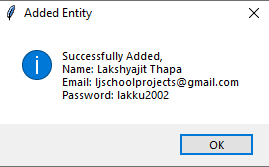
readfile()

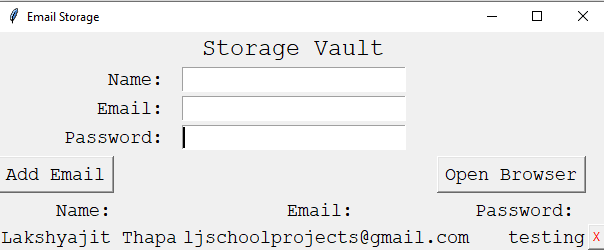
window.mainloop()

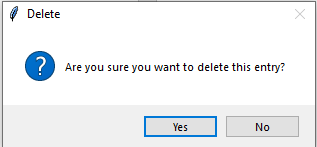
OUTPUT AND SCREENSHOTS

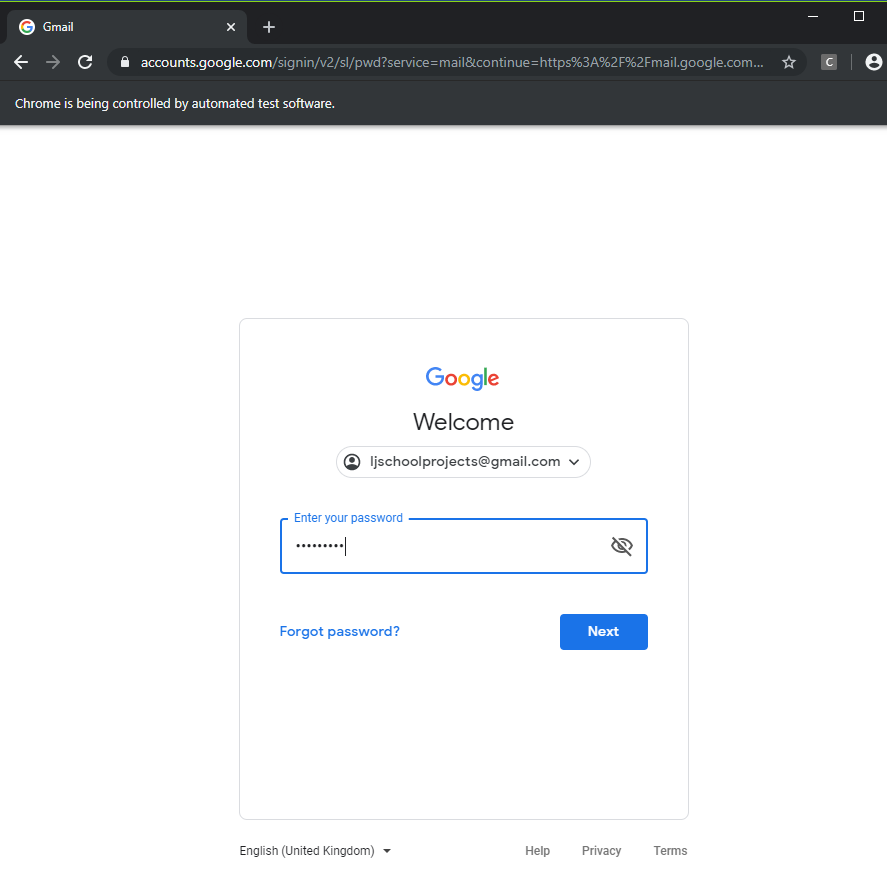












USER MANUAL

* Run the main.exe file to run the program.
* Enter the “password” as the password to access the account vault.
* Enter the credentials.
* To save credentials, press add email.
* To open Gmail account, click on web browser.

CONCLUSION & FUTURE EXPANSION

This program was designed using the python programming language using modules tkinter, selenium, getpass and our own user defined module, automate. Our program sometimes gives an error on termination.

EXPANSION

* We can add a feature where our program logs in to multiple accounts at once, for e.g.- Instagram, Facebook etc
* We could make our interface more graphical.

BIBLIOGRAPHY

* Google.co.in
* w3resources
* stackoverflow.com
* anaconda network
* Sumita Arora Class 12