*Week 5 Progress Report on password Manager*

UpSkill Campus, UniConverge Technologies Pvt.Ltd.

Submitted by: Singam Sree Charitha

Correspondence Address: Singam Sree Charitha

Madanapalle Institute of Technology and Science, Department of Computer Science and Engineering, Angallu, Andhra Pradesh, 517326,

India.

Mail: 191051350sms@gmail.com

Phone: +91 9160086756

**Overview**

During the fifth week,I took my step towards the source code of my project work,I am almost there to complete my project this is the last step to my project

Tasks Completed

As the implementation have three stages

* Configure
* Add new entries
* Get entries

As I completed writing source code for above three section in this code I wrote drivers code which combines all three sections

Challenges Faced

* This week task is really important because this code combines all three sections together.Finally,I made it.
* As always, this week motivates me by some errors.

Lessons Learned

Overall, the lessons learned from the "Password Manger" project highlighted the importance of data security and it is a real-world application. These insights will guide us in future projects and contribute to our professional growth.

Code progress

import argparse

from getpass import getpass

import hashlib

import pyperclip

from rich import print as printc

import utils.add

import utils.retrieve

import utils.generate

from utils.dbconfig import dbconfig

parser = argparse.ArgumentParser(description='Description')

parser.add\_argument('option', help='(a)dd / (e)xtract / (g)enerate')

parser.add\_argument("-s", "--name", help="Site name")

parser.add\_argument("-u", "--url", help="Site URL")

parser.add\_argument("-e", "--email", help="Email")

parser.add\_argument("-l", "--login", help="Username")

parser.add\_argument("--length", help="Length of the password to generate",type=int)

parser.add\_argument("-c", "--copy", action='store\_true', help='Copy password to clipboard')

args = parser.parse\_args()

def inputAndValidateMasterPassword():

mp = getpass("MASTER PASSWORD: ")

hashed\_mp = hashlib.sha256(mp.encode()).hexdigest()

db = dbconfig()

cursor = db.cursor()

query = "SELECT \* FROM pm.secrets"

cursor.execute(query)

result = cursor.fetchall()[0]

if hashed\_mp != result[0]:

printc("[red][!] WRONG! [/red]")

return None

return [mp,result[1]]

def main():

if args.option in ["add","a"]:

if args.name == None or args.url == None or args.login == None:

if args.name == None:

printc("[red][!][/red] Site Name (-s) required ")

if args.url == None:

printc("[red][!][/red] Site URL (-u) required ")

if args.login == None:

printc("[red][!][/red] Site Login (-l) required ")

return

if args.email == None:

args.email = ""

res = inputAndValidateMasterPassword()

if res is not None:

utils.add.addEntry(res[0],res[1],args.name,args.url,args.email,args.login)

if args.option in ["extract","e"]:

# if args.name == None and args.url == None and args.email == None and args.login == None:

# # retrieve all

# printc("[red][!][/red] Please enter at least one search field (sitename/url/email/username)")

# return

res = inputAndValidateMasterPassword()

search = {}

if args.name is not None:

search["sitename"] = args.name

if args.url is not None:

search["siteurl"] = args.url

if args.email is not None:

search["email"] = args.email

if args.login is not None:

search["username"] = args.login

if res is not None:

utils.retrieve.retrieveEntries(res[0],res[1],search,decryptPassword = args.copy)

if args.option in ["generate","g"]:

if args.length == None:

printc("[red][+][/red] Specify length of the password to generate (--length)")

return

password = utils.generate.generatePassword(args.length)

pyperclip.copy(password)

printc("[green][+][/green] Password generated and copied to clipboard")

main()