Ramos, Wendell

Task Manager (Code Program)

# Import necessary modules

import random

# Global dictionary to store users and tasks

users = {}

tasks = {}

# Class for User Management

class User:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

def authenticate(self, username, password):

return self.username == username and self.password == password

# Function to create a new user

def create\_user():

global users

username = input("\nEnter new username: ")

if username in users:

print("❌ Username already exists!")

return False

password = input("Enter new password: ")

users[username] = User(username, password)

tasks[username] = [] # Initialize empty task list for new user

print("✅ User created successfully!")

return True

# Function to login

def login():

username = input("\nEnter username: ")

password = input("Enter password: ")

if username in users and users[username].authenticate(username, password):

print(f"✅ Welcome back, {username}!")

return username

else:

print("❌ Invalid credentials!")

return None

# Function to add a task

def add\_task(username):

task\_name = input("\nEnter task: ")

if username not in tasks:

tasks[username] = [] # Initialize if not exists

tasks[username].append(task\_name)

print(f"✅ Task '{task\_name}' added!")

# Function to remove a task

def remove\_task(username):

if username not in tasks or not tasks[username]:

print("❌ No tasks to remove!")

return

print("\nYour Tasks:")

for i, task in enumerate(tasks[username], 1):

print(f"{i}. {task}")

try:

task\_index = int(input("\nEnter task number to remove: ")) - 1

assert 0 <= task\_index < len(tasks[username]), "Invalid task number!"

removed\_task = tasks[username].pop(task\_index)

print(f"✅ Task '{removed\_task}' removed!")

except (ValueError, AssertionError) as e:

print("❌ Error:", e)

# Function to list tasks

def list\_tasks(username):

if username not in tasks or not tasks[username]:

print("\n📌 No tasks available.")

return

print("\n📋 Your Tasks:")

for i, task in enumerate(tasks[username], 1):

print(f"{i}. {task}")

# Function using with statement (file handling)

def save\_tasks():

with open("tasks.txt", "w") as file:

file.write(str(tasks))

print("\n💾 Tasks saved successfully!")

# Main Execution

if \_\_name\_\_ == "\_\_main\_\_":

print("📌 Welcome to Task Manager!")

while True:

print("\n1. Register\n2. Login\n3. Exit")

choice = input("Choose an option: ")

if choice == "1":

create\_user()

elif choice == "2":

user = login()

if user:

while True:

print("\n--- Task Manager Menu ---")

print("1. Add Task\n2. Remove Task\n3. List Tasks\n4. Save Tasks\n5. Logout")

option = input("Choose an option: ")

if option == "1":

add\_task(user)

elif option == "2":

remove\_task(user)

elif option == "3":

list\_tasks(user)

elif option == "4":

save\_tasks()

elif option == "5":

print("👋 Logging out...\n")

break

else:

print("❌ Invalid option!")

elif choice == "3":

print("👋 Exiting Task Manager. Goodbye!")

break

else:

print("❌ Invalid choice! Try again.")# Import necessary modules

import random

# Global dictionary to store users and tasks

users = {}

tasks = {}

# Class for User Management

class User:

def \_\_init\_\_(self, username, password):

self.username = username

self.password = password

def authenticate(self, username, password):

return self.username == username and self.password == password

# Function to create a new user

def create\_user():

global users

username = input("\nEnter new username: ")

if username in users:

print("❌ Username already exists!")

return False

password = input("Enter new password: ")

users[username] = User(username, password)

tasks[username] = [] # Initialize empty task list for new user

print("✅ User created successfully!")

return True

# Function to login

def login():

username = input("\nEnter username: ")

password = input("Enter password: ")

if username in users and users[username].authenticate(username, password):

print(f"✅ Welcome back, {username}!")

return username

else:

print("❌ Invalid credentials!")

return None

# Function to add a task

def add\_task(username):

task\_name = input("\nEnter task: ")

if username not in tasks:

tasks[user