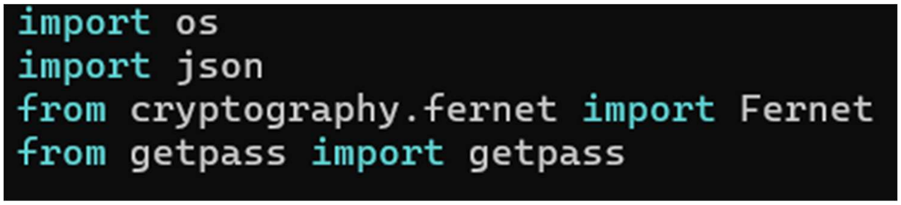
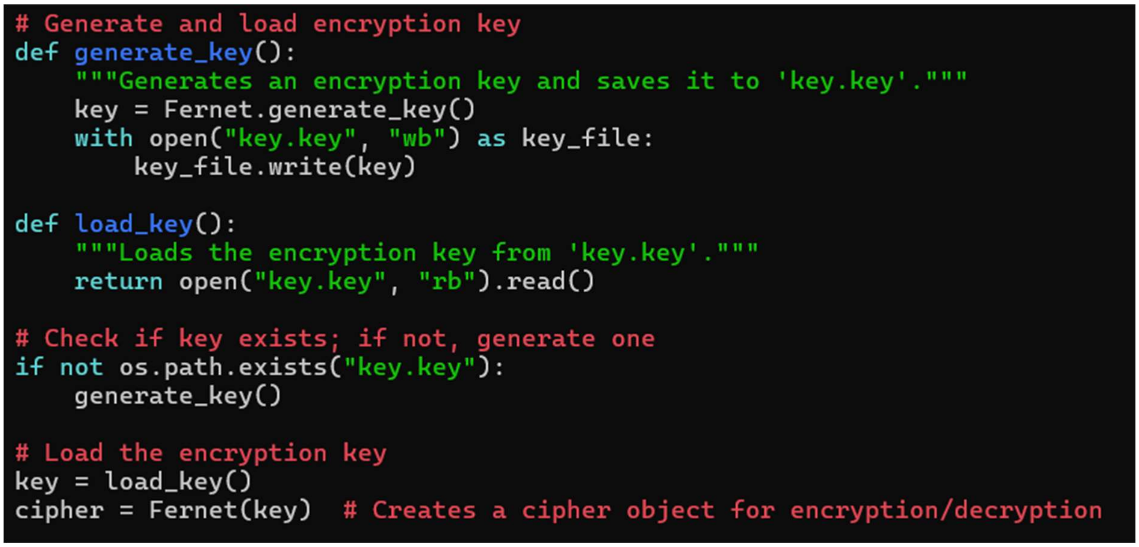
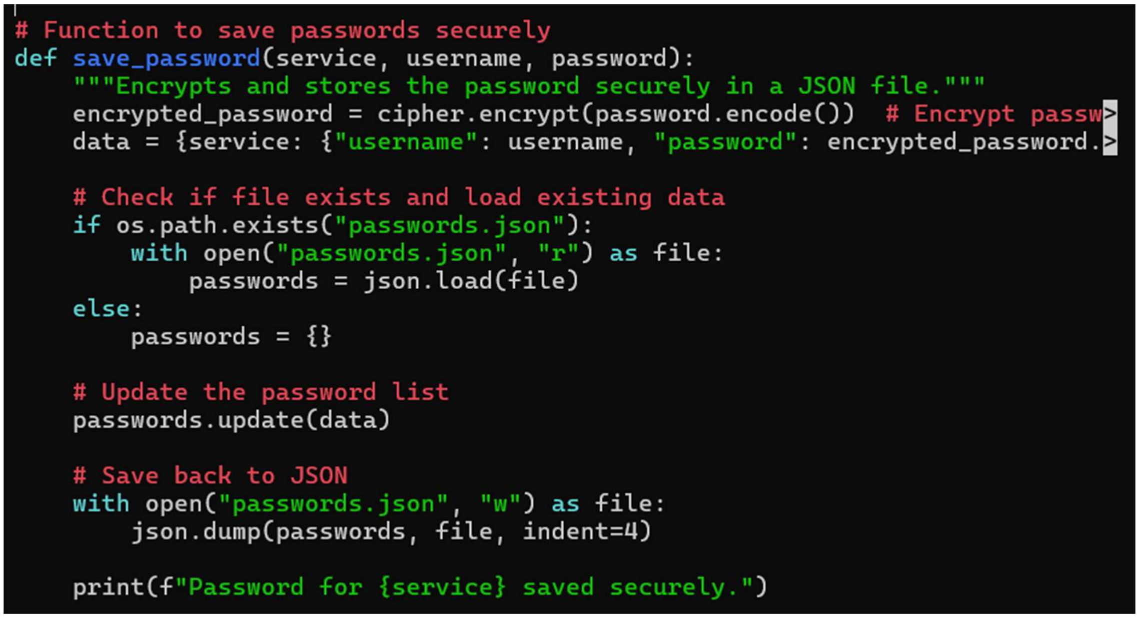
Images of code



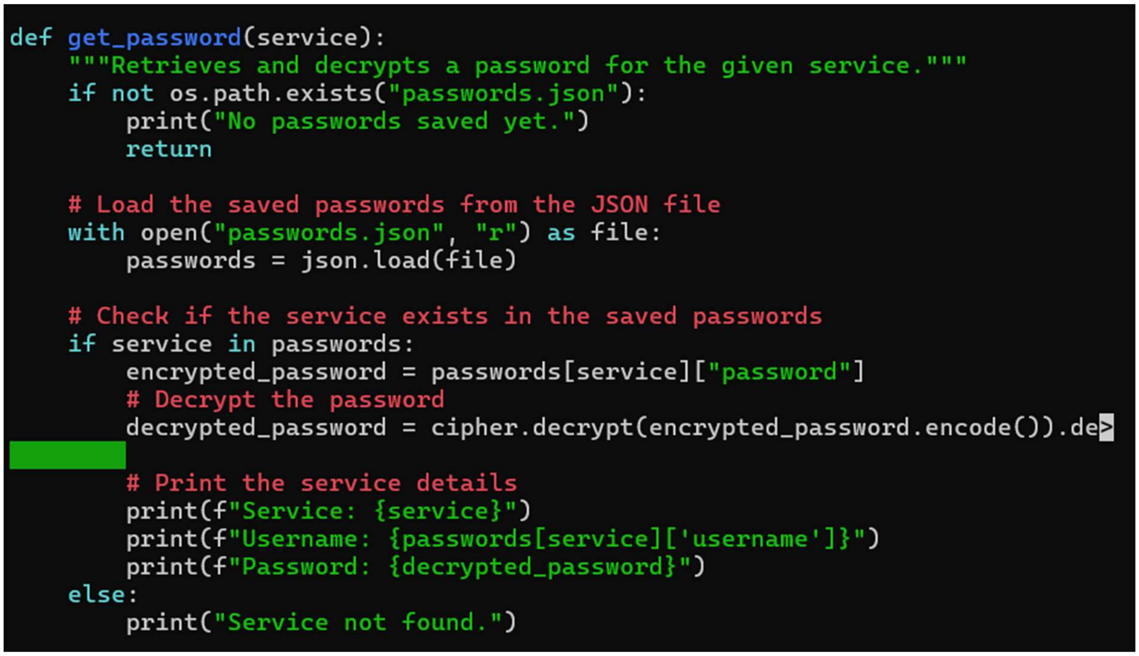
```python
import os
import json
from cryptography.fernet import Fernet
from getpass import getpass
```



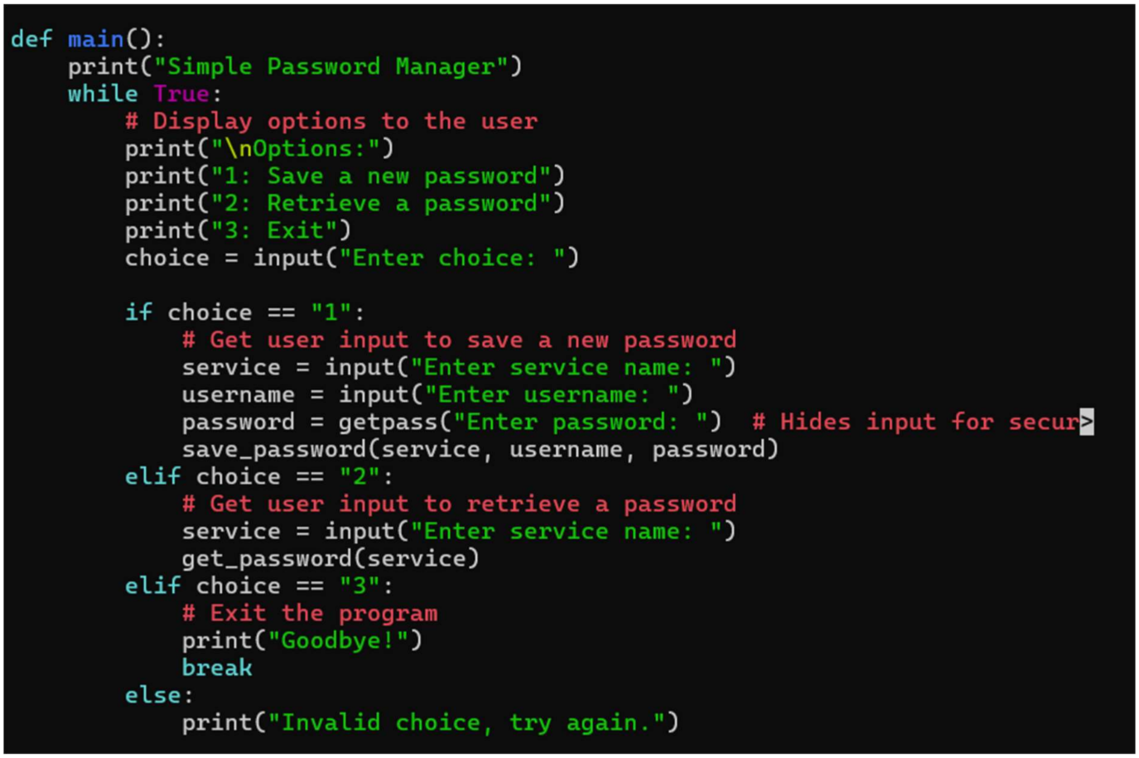
```python
# Generate and load encryption key
def generate_key():
    """Generates an encryption key and saves it to 'key.key'."""
    key = Fernet.generate_key()
    with open("key.key", "wb") as key_file:
        key_file.write(key)

def load_key():
    """Loads the encryption key from 'key.key'."""
    return open("key.key", "rb").read()

# Check if key exists; if not, generate one
if not os.path.exists("key.key"):
    generate_key()

# Load the encryption key
key = load_key()
cipher = Fernet(key)  # Creates a cipher object for encryption/decryption
```



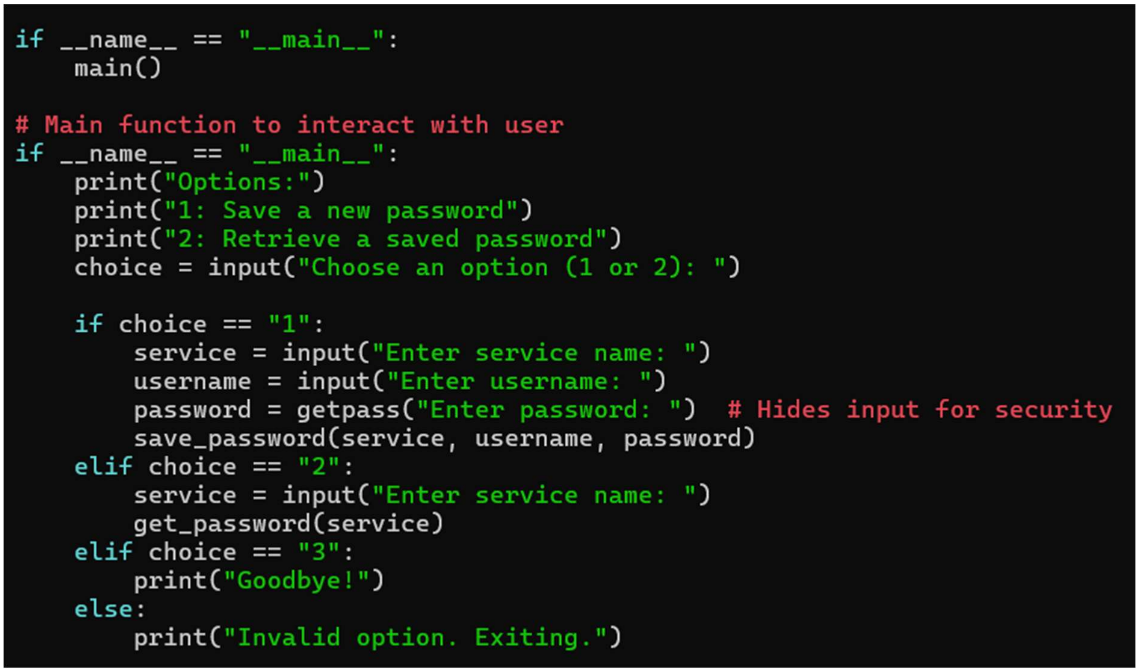
```python
# Function to save passwords securely
def save_password(service, username, password):
    """Encrypts and stores the password securely in a JSON file."""
    encrypted_password = cipher.encrypt(password.encode())  # Encrypt passw>
    data = {service: {"username": username, "password": encrypted_password.>

    # Check if file exists and load existing data
    if os.path.exists("passwords.json"):
        with open("passwords.json", "r") as file:
            passwords = json.load(file)
    else:
        passwords = {}

    # Update the password list
    passwords.update(data)

    # Save back to JSON
    with open("passwords.json", "w") as file:
        json.dump(passwords, file, indent=4)

    print(f"Password for {service} saved securely.")
```



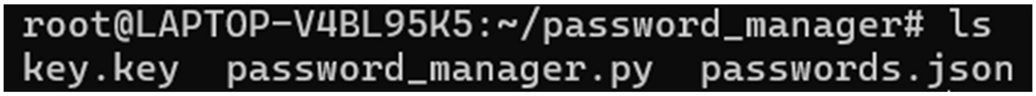
```
def get_password(service):
    """Retrieves and decrypts a password for the given service."""
    if not os.path.exists("passwords.json"):
        print("No passwords saved yet.")
        return

    # Load the saved passwords from the JSON file
    with open("passwords.json", "r") as file:
        passwords = json.load(file)

    # Check if the service exists in the saved passwords
    if service in passwords:
        encrypted_password = passwords[service]["password"]
        # Decrypt the password
        decrypted_password = cipher.decrypt(encrypted_password.encode()).de>

        # Print the service details
        print(f"Service: {service}")
        print(f"Username: {passwords[service]['username']}")
        print(f"Password: {decrypted_password}")
    else:
        print("Service not found.")
```



```
def main():
    print("Simple Password Manager")
    while True:
        # Display options to the user
        print("\nOptions:")
        print("1: Save a new password")
        print("2: Retrieve a password")
        print("3: Exit")
        choice = input("Enter choice: ")

        if choice == "1":
            # Get user input to save a new password
            service = input("Enter service name: ")
            username = input("Enter username: ")
            password = getpass("Enter password: ")  # Hides input for secur>
            save_password(service, username, password)
        elif choice == "2":
            # Get user input to retrieve a password
            service = input("Enter service name: ")
            get_password(service)
        elif choice == "3":
            # Exit the program
            print("Goodbye!")
            break
        else:
            print("Invalid choice, try again.")
```

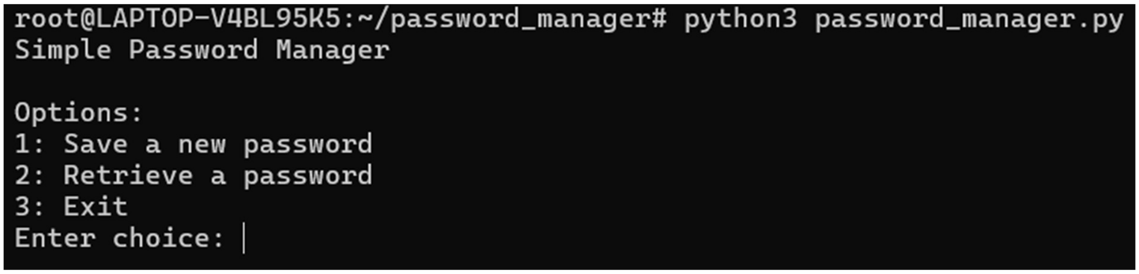


```
if __name__ == "__main__":
    main()

# Main function to interact with user
if __name__ == "__main__":
    print("Options:")
    print("1: Save a new password")
    print("2: Retrieve a saved password")
    choice = input("Choose an option (1 or 2): ")

    if choice == "1":
        service = input("Enter service name: ")
        username = input("Enter username: ")
        password = getpass("Enter password: ")  # Hides input for security
        save_password(service, username, password)
    elif choice == "2":
        service = input("Enter service name: ")
        get_password(service)
    elif choice == "3":
        print("Goodbye!")
    else:
        print("Invalid option. Exiting.")
```

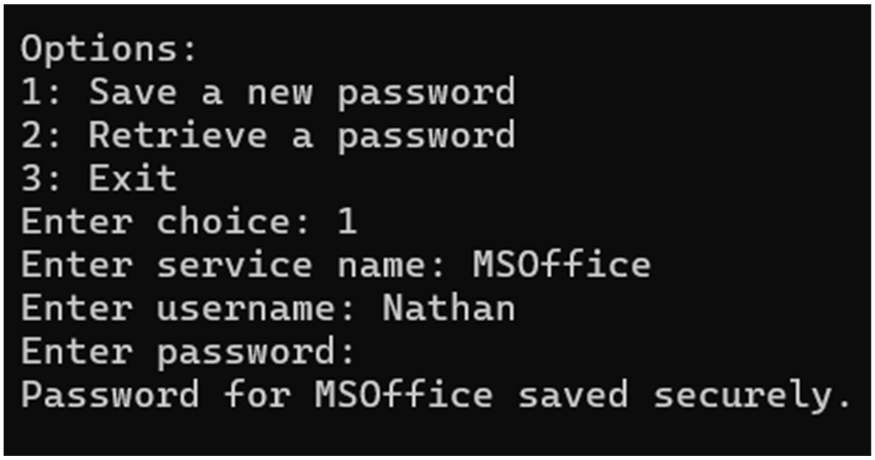


```
root@LAPTOP-V4BL95K5:~/password_manager# ls
key.key  password_manager.py  passwords.json
```

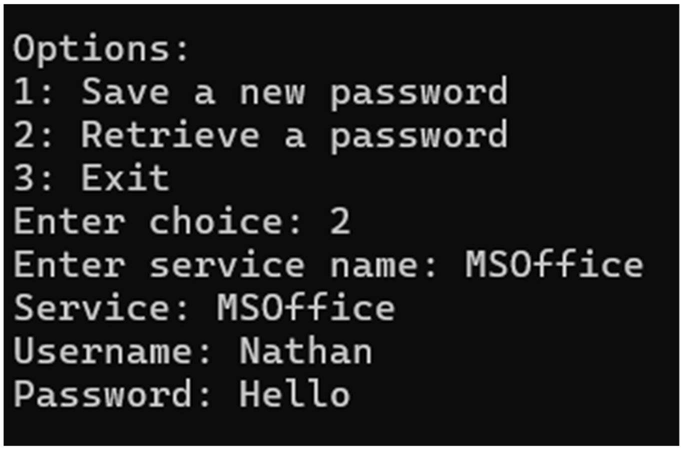
Images of Output



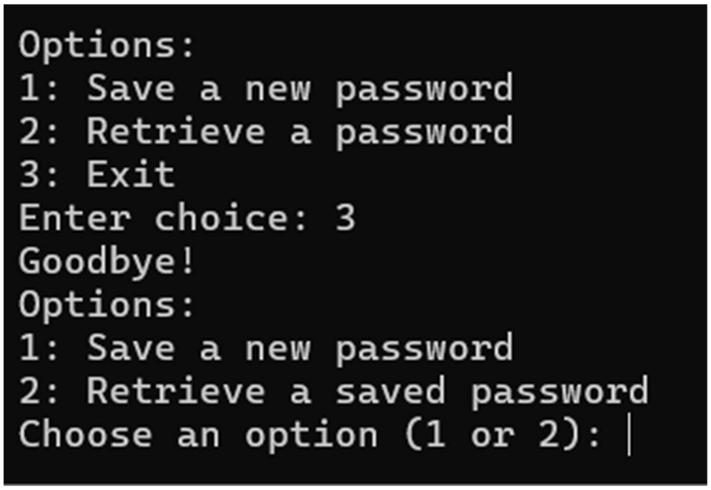
```
root@LAPTOP-V4BL95K5:~/password_manager# python3 password_manager.py
Simple Password Manager

Options:
1: Save a new password
2: Retrieve a password
3: Exit
Enter choice: 
```



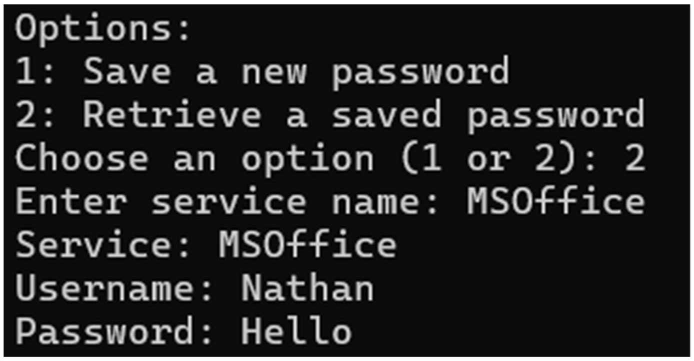
```
Options:
1: Save a new password
2: Retrieve a password
3: Exit
Enter choice: 1
Enter service name: MSOffice
Enter username: Nathan
Enter password:
Password for MSOffice saved securely.
```



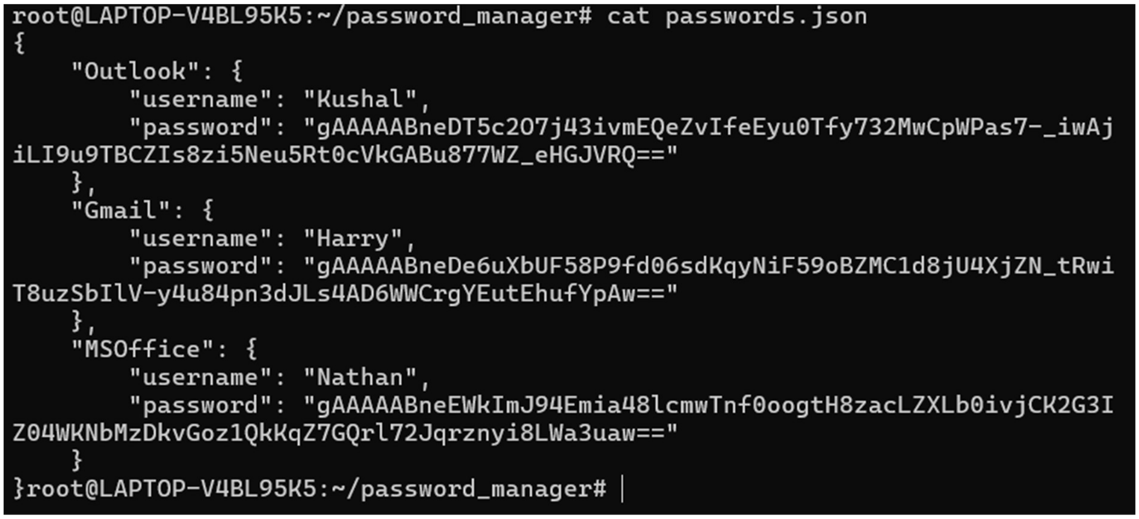
```
Options:
1: Save a new password
2: Retrieve a password
3: Exit
Enter choice: 2
Enter service name: MSOffice
Service: MSOffice
Username: Nathan
Password: Hello
```



```
Options:
1: Save a new password
2: Retrieve a password
3: Exit
Enter choice: 3
Goodbye!
Options:
1: Save a new password
2: Retrieve a saved password
Choose an option (1 or 2): 
```



```
Options:
1: Save a new password
2: Retrieve a saved password
Choose an option (1 or 2): 2
Enter service name: MSOffice
Service: MSOffice
Username: Nathan
Password: Hello
```



```
root@LAPTOP-V4BL95K5:~/password_manager# cat passwords.json
{
    "Outlook": {
        "username": "Kushal",
        "password": "gAAAAABneDT5c2O7j43ivmEQeZvIfeEyu0Tfy732MwCpWPas7-_iwAj
iLI9u9TBCZIs8zi5Neu5Rt0cVkGABu877WZ_eHGJVRQ=="
    },
    "Gmail": {
        "username": "Harry",
        "password": "gAAAAABneDe6uXbUF58P9fd06sdKqyNiF59oBZMC1d8jU4XjZN_tRwi
T8uzSbIlV-y4u84pn3dJLs4AD6WWCrgYEutEhufYpAw=="
    },
    "MSOffice": {
        "username": "Nathan",
        "password": "gAAAAABneEWkImJ94Emia48lcmwTnf0oogtH8zacLZXLb0ivjCK2G3I
Z04WKNbMzDkvGoz1QkKqZ7GQrl72Jqrznyi8LWa3uaw=="
    }
}root@LAPTOP-V4BL95K5:~/password_manager# 
```