```
pip install cryptography
```

```
Requirement already satisfied: cryptography in /usr/local/lib/python3.12/dist-packages (43.0.3)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.12/dist-packages (from cryptograph
Requirement already satisfied: pycparser in /usr/local/lib/python3.12/dist-packages (from cffi>=1.12->
```

```
import json
import os
import base64
import hashlib
from cryptography.fernet import Fernet
```

```
DATA_FILE = "passwords.json"
KEY_FILE = "key.key"


# ---------- Key Generation ----------
def generate_key(master_password):
    hashed = hashlib.sha256(master_password.encode()).digest()
    key = base64.urlsafe_b64encode(hashed)
    return key


def load_key(master_password):
    key = generate_key(master_password)
    return Fernet(key)
```

```
# ---------- File Handling ----------
def load_data():
    if not os.path.exists(DATA_FILE):
        return {}
    with open(DATA_FILE, "r") as file:
        return json.load(file)


def save_data(data):
    with open(DATA_FILE, "w") as file:
        json.dump(data, file, indent=4)
```

```
# ---------- Password Operations ----------
def add_password(fernet):
    site = input("Enter website name: ")
    username = input("Enter username: ")
    password = input("Enter password: ")

    encrypted_pwd = fernet.encrypt(password.encode()).decode()

    data = load_data()
    data[site] = {
        "username": username,
        "password": encrypted_pwd
    }
    save_data(data)
    print("✅ Password added successfully")


def view_password(fernet):
    site = input("Enter website name: ")
```

```
    data = load_data()

    if site in data:
        decrypted_pwd = fernet.decrypt(data[site]["password"].encode()).decode()
        print(f"Username: {data[site]['username']}")
        print(f"Password: {decrypted_pwd}")
    else:
        print("❌ No entry found")


def delete_password():
    site = input("Enter website name: ")
    data = load_data()

    if site in data:
        del data[site]
        save_data(data)
        print("🗑️ Password deleted")
    else:
        print("❌ No entry found")


def search_password():
    keyword = input("Enter keyword: ")
    data = load_data()

    for site in data:
        if keyword.lower() in site.lower():
            print("🔍 Found:", site)
```

```
# ---------- Main Program ----------
def main():
    print("🔐 PASSWORD MANAGER")
    master_password = input("Enter Master Password: ")
    fernet = load_key(master_password)

    while True:
        print("\n1. Add Password")
        print("2. View Password")
        print("3. Search Password")
        print("4. Delete Password")
        print("5. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            add_password(fernet)
        elif choice == "2":
            view_password(fernet)
        elif choice == "3":
            search_password()
        elif choice == "4":
            delete_password()
        elif choice == "5":
            print("👋 Exiting Password Manager")
            break
        else:
            print("❌ Invalid choice")


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

```
🔐 PASSWORD MANAGER
Enter Master Password: manisha123

1. Add Password
2. View Password
3. Search Password
```

```
4. Delete Password
5. Exit
Choose an option: 1
Enter website name: gmail
Enter username: manisha
Enter password: manisha123
✅ Password added successfully

1. Add Password
2. View Password
3. Search Password
4. Delete Password
5. Exit
Choose an option: 2
Enter website name: gmail
Username: manisha
Password: manisha123

1. Add Password
2. View Password
3. Search Password
4. Delete Password
5. Exit
Choose an option: 3
Enter keyword: m
🔍 Found: gmail

1. Add Password
2. View Password
3. Search Password
4. Delete Password
5. Exit
Choose an option: 4
Enter website name: gmail
🗑️ Password deleted

1. Add Password
2. View Password
3. Search Password
4. Delete Password
5. Exit
Choose an option: 3
Enter keyword: mani

1. Add Password
2. View Password
3. Search Password
4. Delete Password
5. Exit
Choose an option: 5
👋 Exiting Password Manager
```