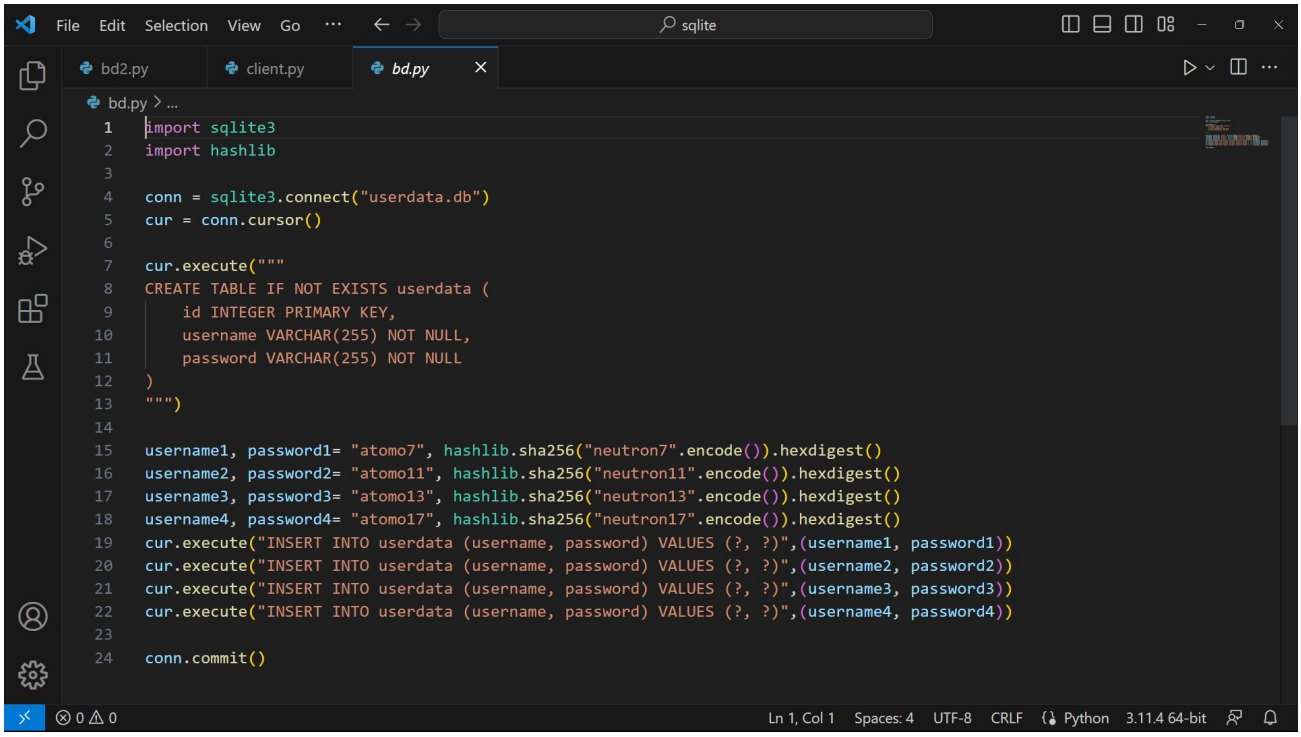
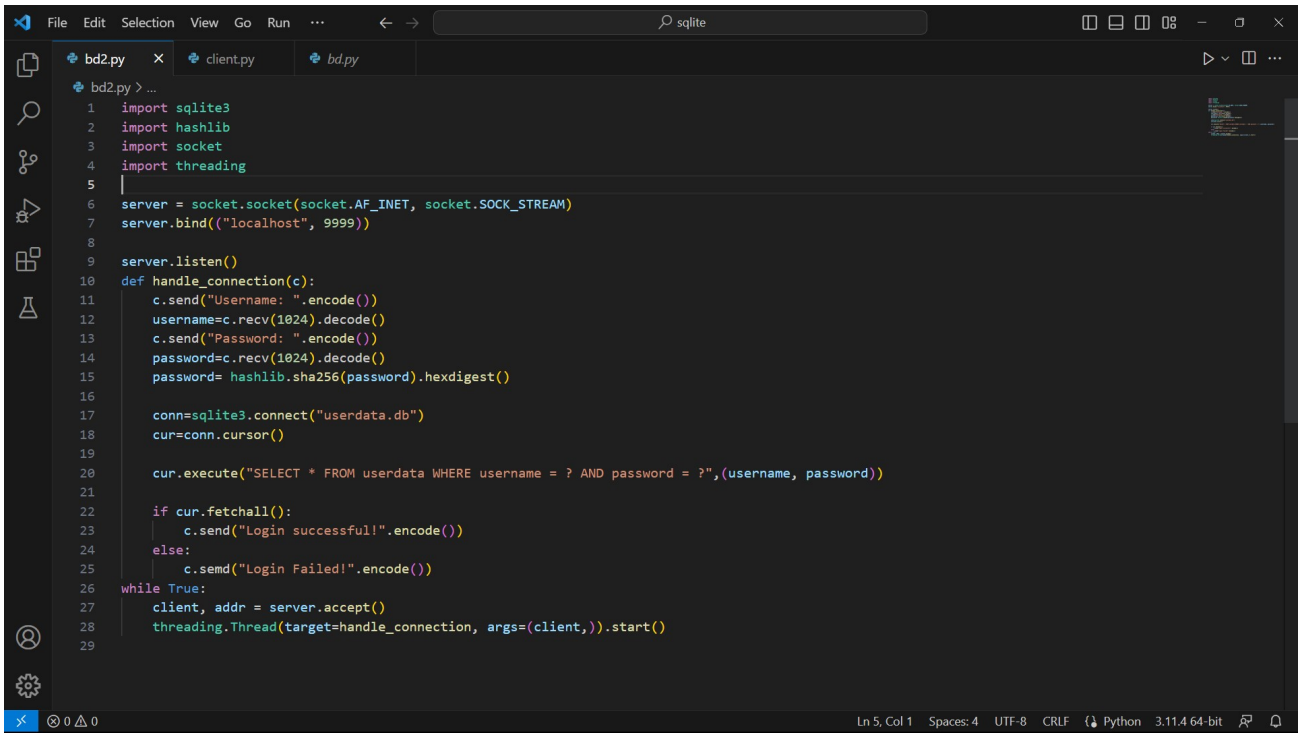
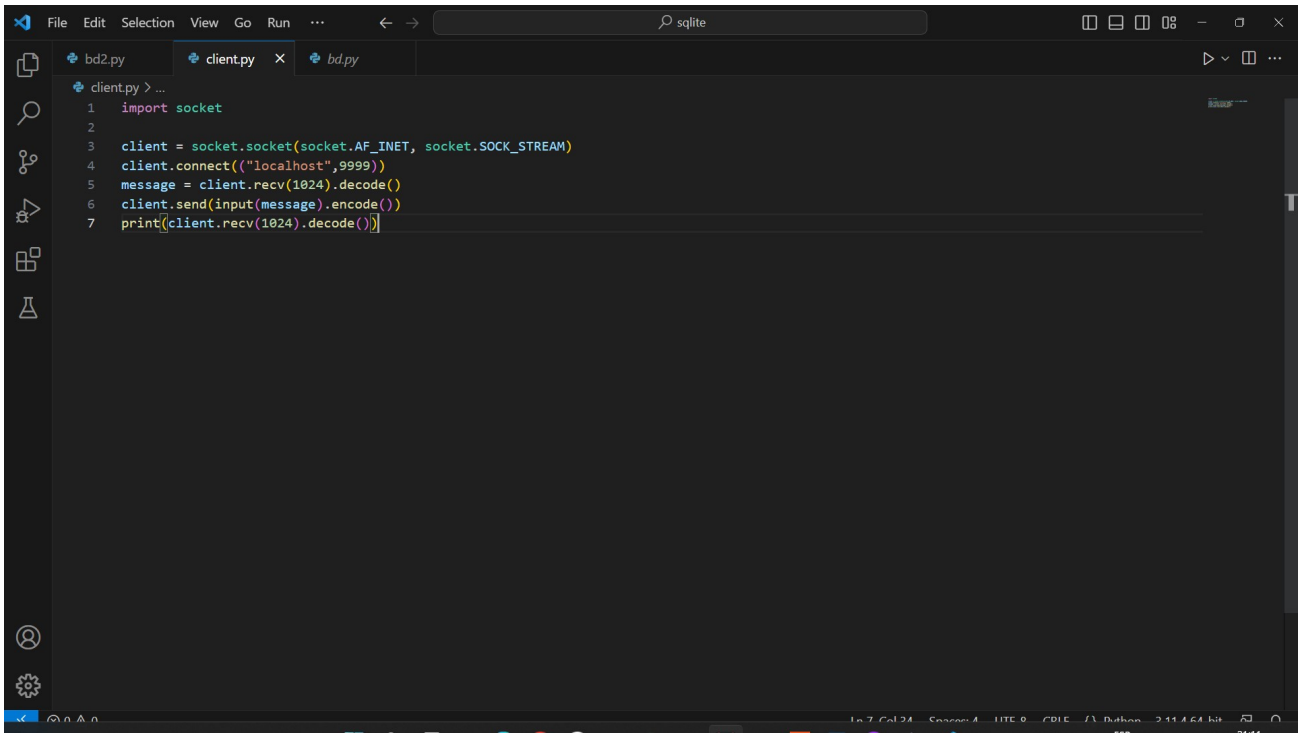
sqlite3 python



```python
import sqlite3
import hashlib

conn = sqlite3.connect("userdata.db")
cur = conn.cursor()

cur.execute("""
CREATE TABLE IF NOT EXISTS userdata (
    id INTEGER PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
)
""")

username1, password1= "atomo7", hashlib.sha256("neutron7".encode()).hexdigest()
username2, password2= "atomo11", hashlib.sha256("neutron11".encode()).hexdigest()
username3, password3= "atomo13", hashlib.sha256("neutron13".encode()).hexdigest()
username4, password4= "atomo17", hashlib.sha256("neutron17".encode()).hexdigest()
cur.execute("INSERT INTO userdata (username, password) VALUES (?, ?)",(username1, password1))
cur.execute("INSERT INTO userdata (username, password) VALUES (?, ?)",(username2, password2))
cur.execute("INSERT INTO userdata (username, password) VALUES (?, ?)",(username3, password3))
cur.execute("INSERT INTO userdata (username, password) VALUES (?, ?)",(username4, password4))

conn.commit()
```



```python
import sqlite3
import hashlib
import socket
import threading

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(("localhost", 9999))

server.listen()
def handle_connection(c):
    c.send("Username: ".encode())
    username=c.recv(1024).decode()
    c.send("Password: ".encode())
    password=c.recv(1024).decode()
    password= hashlib.sha256(password).hexdigest()

    conn=sqlite3.connect("userdata.db")
    cur=conn.cursor()

    cur.execute("SELECT * FROM userdata WHERE username = ? AND password = ?",(username, password))

    if cur.fetchall():
        c.send("Login successful!".encode())
    else:
        c.semd("Login Failed!".encode())
while True:
    client, addr = server.accept()
    threading.Thread(target=handle_connection, args=(client,)).start()
```


File Edit Selection View Go Run
sqlite
bd2.py
client.py
bd.py
client.py > ...
1 import socket
2
3 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 client.connect(("localhost",9999))
5 message = client.recv(1024).decode()
6 client.send(input(message).encode())
7 print(client.recv(1024).decode())