**Microservice of biling**

import json

import random

import sqlite3

def create\_database():

conn = sqlite3.connect('billing.db')

cursor = conn.cursor()

# Create a table to store billing information

cursor.execute('''CREATE TABLE IF NOT EXISTS bills (

id INTEGER PRIMARY KEY AUTOINCREMENT,

customer\_name TEXT NOT NULL,

amount REAL NOT NULL,

paid BOOLEAN NOT NULL DEFAULT 0

)''')

# Commit the changes and close the database connection

conn.commit()

conn.close()

def add\_bill(customer\_name, amount):

conn = sqlite3.connect('billing.db')

cursor = conn.cursor()

cursor.execute('INSERT INTO bills (customer\_name, amount) VALUES (?, ?)', (customer\_name, amount))

conn.commit()

conn.close()

def pay\_bill(bill\_id):

conn = sqlite3.connect('billing.db')

cursor = conn.cursor()

cursor.execute('UPDATE bills SET paid=1 WHERE id=?', (bill\_id,))

conn.commit()

conn.close()

def get\_unpaid\_bills():

conn = sqlite3.connect('billing.db')

cursor = conn.cursor()

cursor.execute('SELECT \* FROM bills WHERE paid=0')

unpaid\_bills = cursor.fetchall()

conn.close()

return unpaid\_bills

if \_name\_ == '\_main\_':

# Create the database (if not already created)

create\_database()

# Add some sample bills

add\_bill('Uzwa Safdar', 100.0)

add\_bill('Esha Afzal ', 50.0)

# Get unpaid bills

unpaid\_bills = get\_unpaid\_bills()

print("Unpaid Bills:")

for bill in unpaid\_bills:

print(f"Bill ID: {bill[0]}, Customer: {bill[1]}, Amount: {bill[2]}, Paid: {bill[3]}")

# Mark a bill as paid

bill\_id\_to\_pay = 1

pay\_bill(bill\_id\_to\_pay)

# Get updated unpaid bills

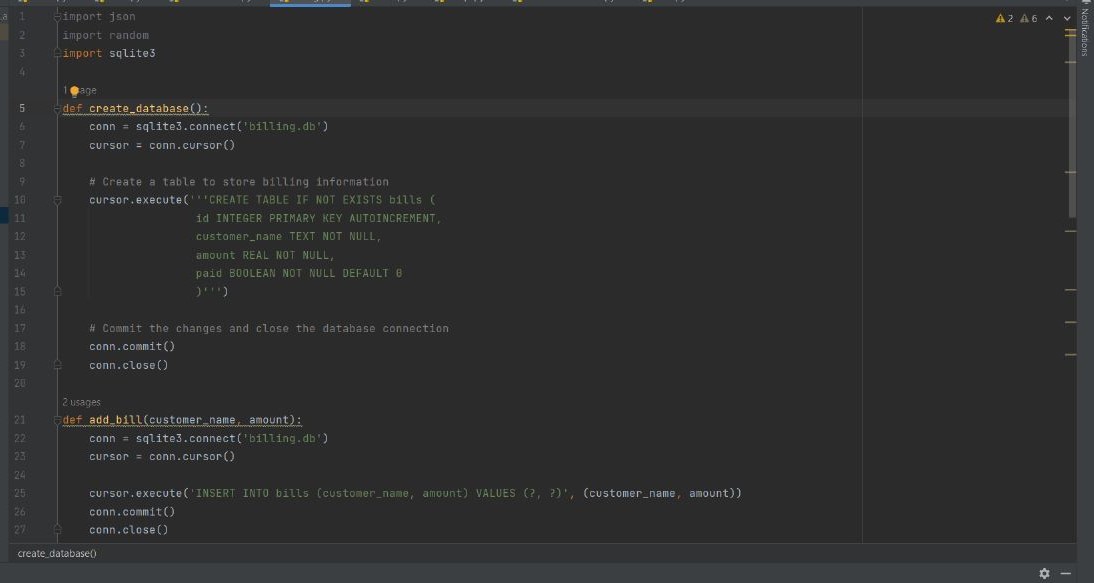
updated\_unpaid\_bills = get\_unpaid\_bills()

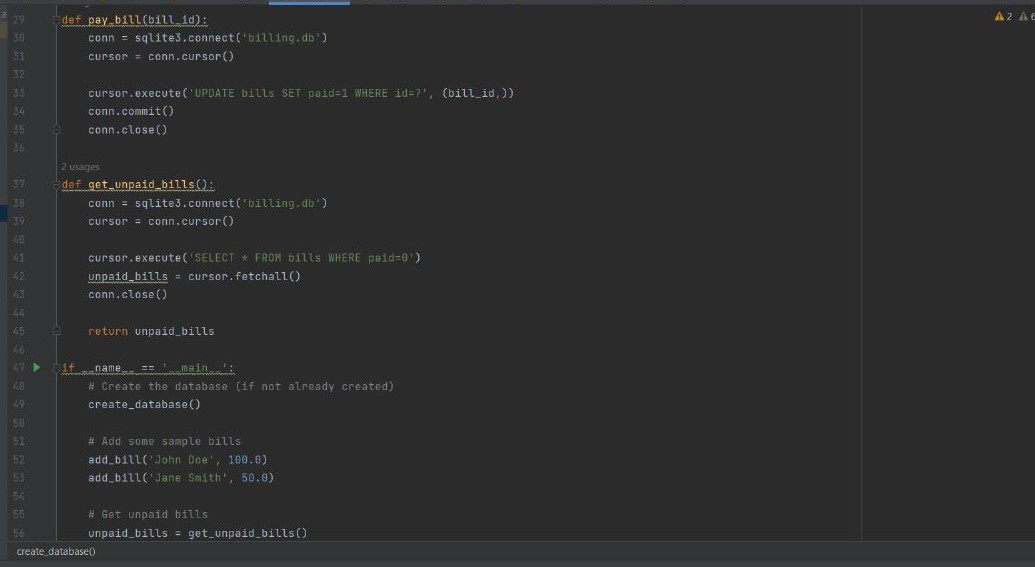
print("\nUpdated Unpaid Bills:")

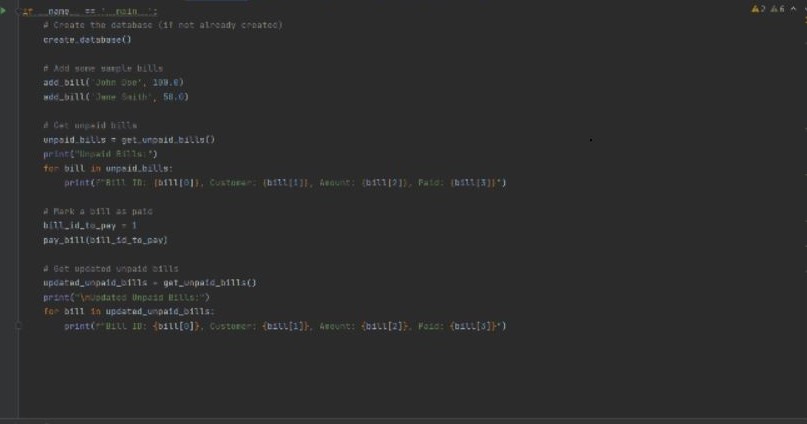
for bill in updated\_unpaid\_bills:

print(f"Bill ID: {bill[0]}, Customer: {bill[1]}, Amount: {bill[2]}, Paid: {bill[3]}"

Code screenshot







**output**

