**Name:**

**Student ID:**

**Program:**

**11. API and SQLite**

**What is an API?**

An API (Application Programming Interface) allows different software applications to communicate with each other. It provides a set of rules and endpoints for accessing and manipulating data.

**GET Request Using requests**

python

import requests

def make\_get\_request():

url = "https://jsonplaceholder.typicode.com/posts/1" # Example API

try:

response = requests.get(url)

if response.status\_code == 200:

print("Response Data:", response.json())

else:

print("Failed to fetch data. Status code:", response.status\_code)

except requests.RequestException as e:

print("An error occurred while making the GET request:", e)

make\_get\_request()

**SQLite Connection**

Steps to connect to SQLite:

1. **Import sqlite3:** To use SQLite in Python.
2. **Connect to Database:** Establish a connection to an SQLite database file (or create a new one if it doesn’t exist).
3. **Create a Cursor:** To execute SQL commands.
4. **Execute Queries:** Use SQL commands to interact with the database.
5. **Commit Changes:** Save changes to the database.
6. **Close Connection:** Close the database connection to avoid memory leaks.

Example Code:

python

import sqlite3

def connect\_to\_database():

try:

# Connect to SQLite database (creates file if it doesn't exist)

conn = sqlite3.connect("example.db")

cursor = conn.cursor()

# Create a table

cursor.execute("""

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

age INTEGER NOT NULL

)

""")

# Insert data

cursor.execute("INSERT INTO users (name, age) VALUES (?, ?)", ("Alice", 25))

# Commit changes and fetch data

conn.commit()

cursor.execute("SELECT \* FROM users")

print("Users:", cursor.fetchall())

except sqlite3.Error as e:

print("Database error:", e)

finally:

# Close connection

conn.close()

connect\_to\_database()