

CS5600 Programming Assignment2 (CRN 13892)

Name: Venkata Lakshmi Sasank Tipparaju

Student ID: 700738838

CS5600 – CRN:13892

Programming Assignment 2 – Neo4J CRUD Operations API Calls

Nodes:

Movie [show_id, type, title, release_year, rating, duration, description]

Person [name]

Country [name]

Relationships: ACTED_IN, DIRECTED, DISTRIBUTED_IN

The Neo4j Database Configuration:

url = "bolt://localhost:7687" for local database

usr = "neo4j" //default username

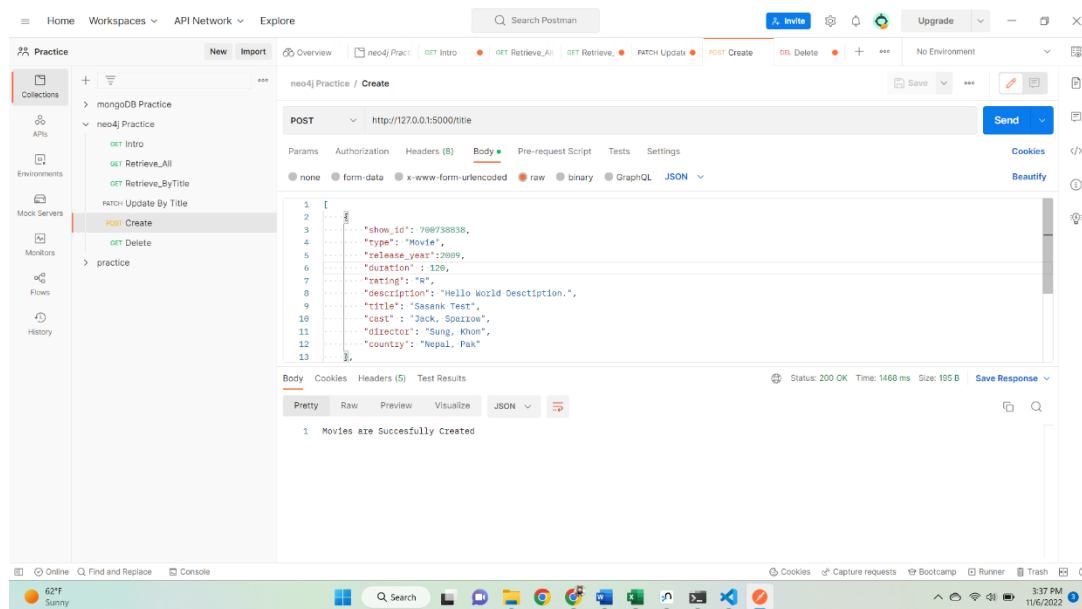
pwd = "Admin1234" //the password

Insert: Sample

```
[
  {
    "show_id": 700738838,
    "type": "Movie",
    "release_year": 2009,
    "duration": 120,
    "rating": "R",
    "description": "Hello World Description.",
    "title": "Sasank Test",
    "cast": "Jack, Sparrow",
    "director": "Sung, Jhon",
    "country": "Nepal, Pak"
  },
  {
    "rating": "R",
    "description": "In a dystopian future, an insurance adjuster for a tech company investigates a robot killed for violating protocol and discovers a global conspiracy.",
    "title": "Sasank Test2",
    "director": "Han, Ban",
    "country": "Nepal, USA"
  }
]
```

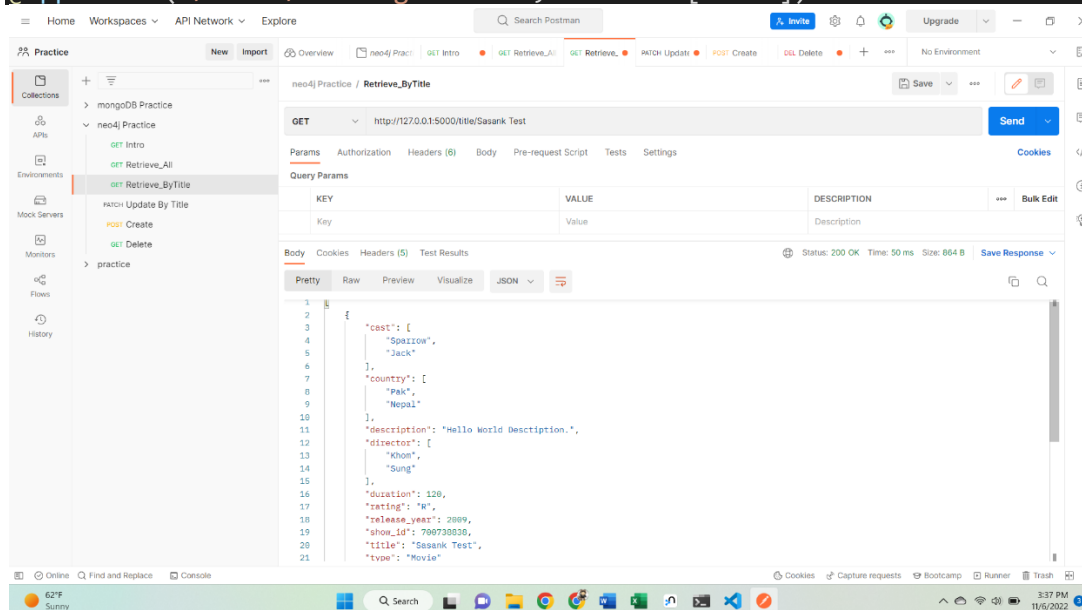
1. Insert the new movie and show.

```
@app.route('/title', methods=['POST'])
```



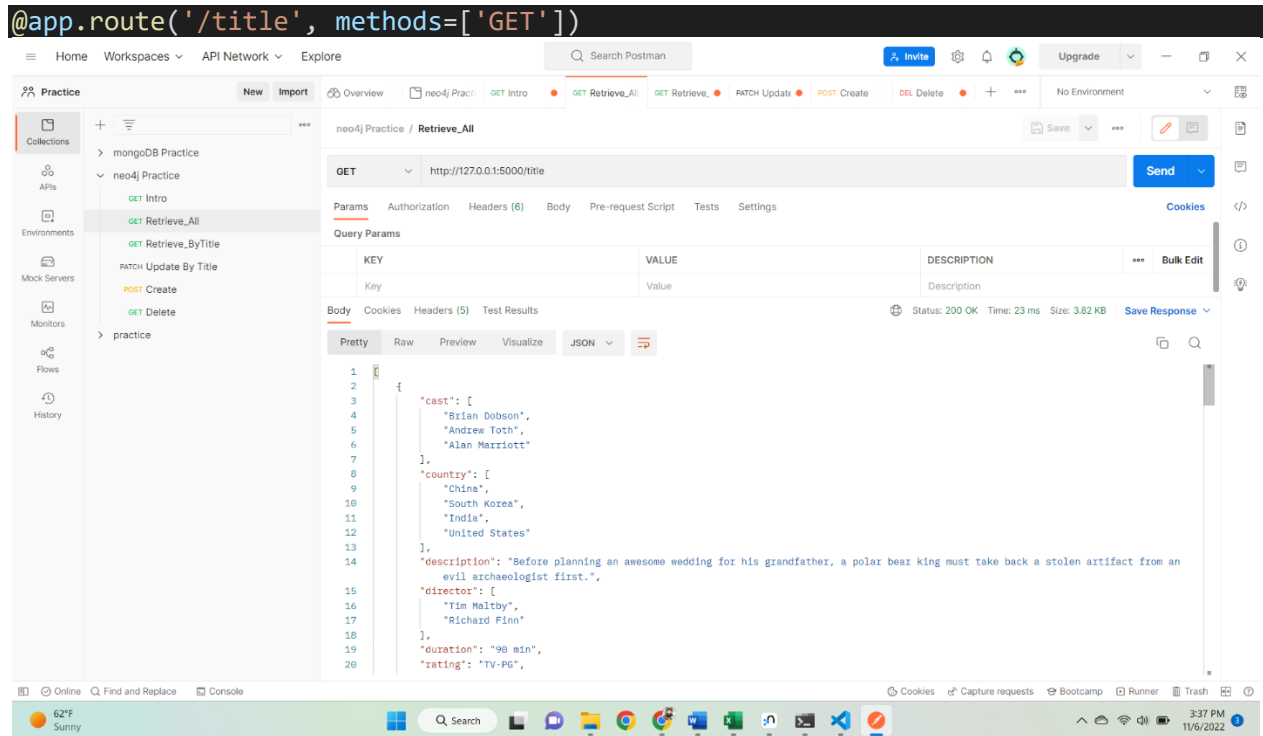
2. Display the movie and show's detail includes actors, directors and distributed country using title.

```
@app.route('/title/<string:fname>', methods=['GET'])
```



3. Retrieve all the movies and shows in database.

```
@app.route('/title', methods=['GET'])
```

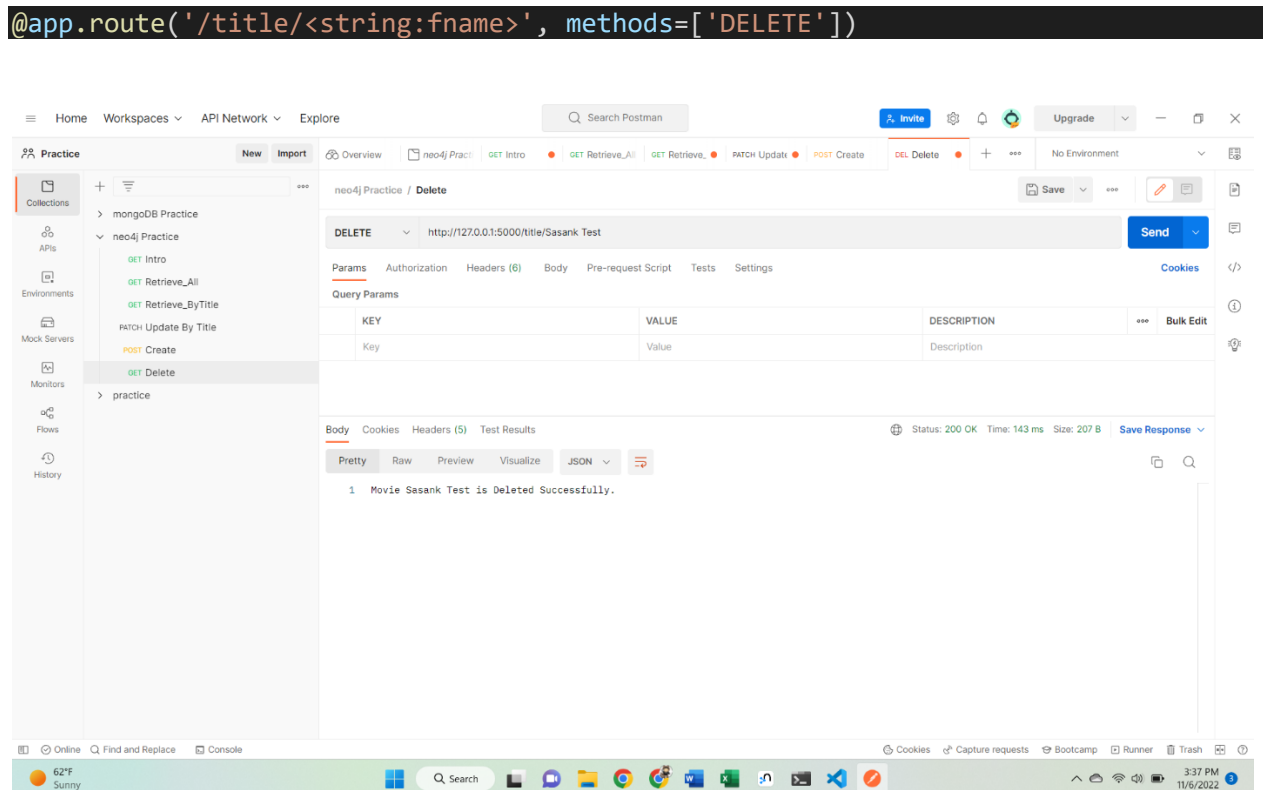


The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/title`. The response is a JSON object with the following structure:

```
{
  "cast": [
    "Brian Dobson",
    "Andrew Toth",
    "Alan Marriott"
  ],
  "country": [
    "China",
    "South Korea",
    "India",
    "United States"
  ],
  "description": "Before planning an awesome wedding for his grandfather, a polar bear king must take back a stolen artifact from an evil archaeologist first.",
  "director": [
    "Tim Maltby",
    "Richard Finn"
  ],
  "duration": "98 min",
  "rating": "TV-PG"
}
```

4. Delete the movie and show information using title.

```
@app.route('/title/<string:fname>', methods=['DELETE'])
```



The screenshot shows the Postman interface with a DELETE request to `http://127.0.0.1:5000/title/Sasank Test`. The response is a plain text message:

```
1 Movie Sasank Test is Deleted Successfully.
```

- Update the movie and show information using title. (By update only title, description, and rating)

```
@app.route('/title/<string:fname>', methods=['PATCH'])
```

The screenshot shows the Postman application interface. On the left, the 'Collections' sidebar is open, showing a collection named 'neo4j Practice' with several endpoints. The 'PATCH Update By Title' endpoint is selected. The main panel displays the details of this PATCH request. The URL is 'http://127.0.0.1:5000/title/Sasank Test2'. The request body is a JSON object:

```
{ "title": "Sasank Test82", "description": "Testing Desc updated", "rating": "Y" }
```

. The response status is '200 OK' with a time of '120 ms' and a size of '298 B'. The response body is a JSON object:

```
{ "ShowInfo": { "description": "Testing Desc updated", "rating": "Y", "title": "Sasank Test82" } }
```

The screenshot shows the Postman application interface. On the left, the 'Collections' sidebar is open, showing a collection named 'neo4j Practice' with several endpoints. The 'GET Retrieve_ByTitle' endpoint is selected. The main panel displays the details of this GET request. The URL is 'http://127.0.0.1:5000/title/Sasank Test'. The response status is '200 OK' with a time of '17 ms' and a size of '385 B'. The response body is a JSON object:

```
{ "cast": [], "country": [ "Nepal", "USA" ], "description": "Testing Desc updated", "director": [ "Han", "Ban" ], "rating": "Y", "title": "Sasank Test82" }
```