

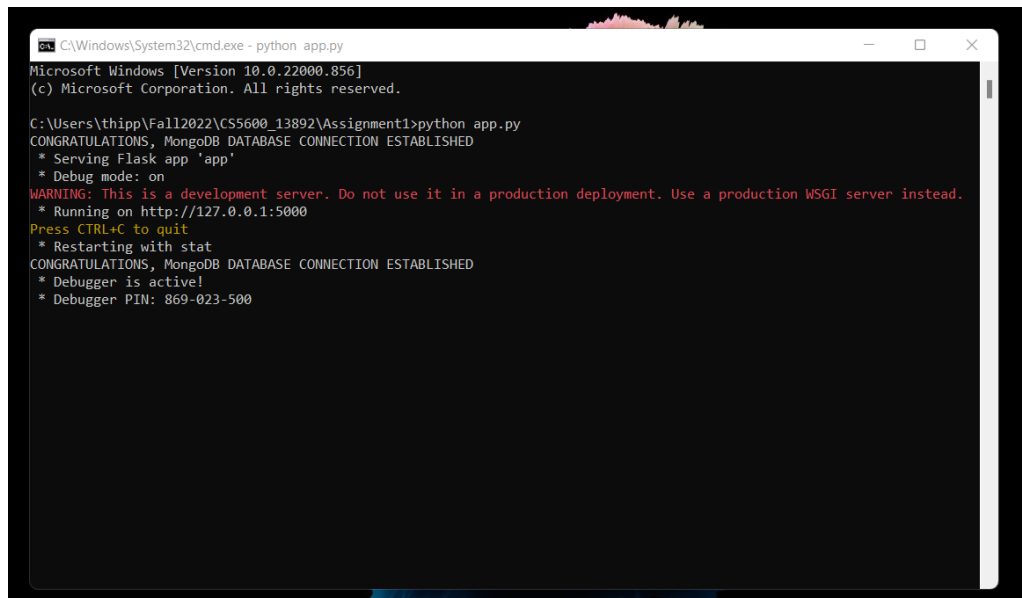
CS5600 Programming Assignment1 (CRN 13892)

First Name: Venkata Lakshmi Sasank

Last Name: Tipparaju

Student# 700738838

Navigate to file's location in Command Prompt: and then **python app.py**



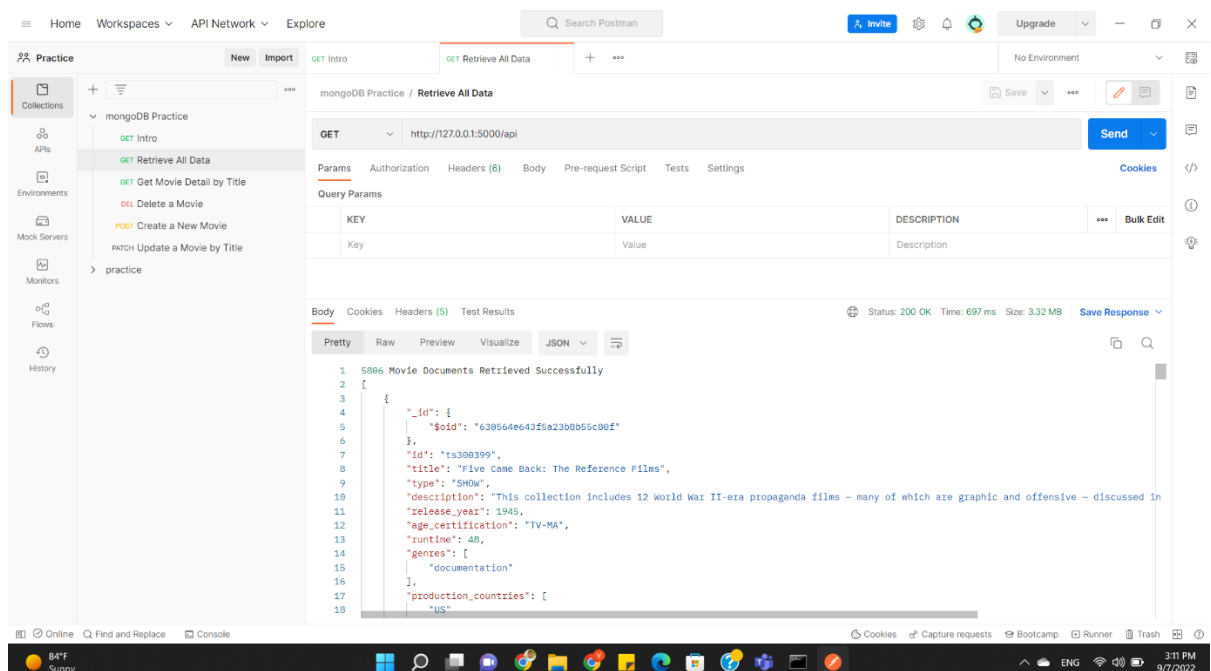
```
C:\Windows\System32\cmd.exe - python app.py
Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thipp\Fall2022\CS5600_13892\Assignment1>python app.py
CONGRATULATIONS, MongoDB DATABASE CONNECTION ESTABLISHED
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
CONGRATULATIONS, MongoDB DATABASE CONNECTION ESTABLISHED
* Debugger is active!
* Debugger PIN: 869-023-500
```

Open Postman and do the following operations

1. Retrieving all MongoDB documents from **database['netflix']**

Method: **GET**, <http://127.0.0.1:5000/api>



2. Create a movie document in netflix collection

Method: **POST**, <http://127.0.0.1:5000/api>

*For Insert Operation sample movie data of 2 documents is provided below
please provide input in body - raw - json type*

```
[
  {
    "title": "Test Movie",
    "type": "Show",
    "id": "tm84618",
    "description": "This is sample movie for demo.",
    "release_year": 2022,
    "age_certification": "A",
    "genres": [
      "crime",
      "drama"
    ],
    "production_countries": [
      "US"
    ],
    "runtime": 113,
    "imdb_score": 8.3
  },
  {
    "title": "Test Movie2",
    "type": "Show",
    "id": "tm84622",
    "description": "This is sample movie for demo.",
    "release_year": 2022,
    "age_certification": "A",
    "runtime": 113,
    "imdb_score": 8.4
  }
]
```

Home Workspaces API Network Explore Search Postman Invite Upgrade

Practice New Import GET Intro GET Retrieve All Data POST Create a New Movie No Environment

mongoDB Practice / Create a New Movie

POST http://127.0.0.1:5000/api

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "title": "Test Movie",
3   "type": "Show",
4   "id": "tm84618",
5   "description": "This is sample movie for demo.",
6   "release_year": 2022,
7   "age_certification": "A",
8   "genres": [
9     "crime",
10    "drama"
11  ],
12  "production_countries": [
13    "US"
14  ],
15  "runtime": 113,
16  "imdb_score": 8.3
17 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 13 ms Size: 887 B Save Response

Pretty Raw Preview Visualize JSON

```
1 2 Movie Documents Created Successfully
2 {
3   "_id": {
4     "$oid": "6318fb0ca73f61915887f17e"
5   },
6   ...
7 }
```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash 84°F Sunny 3:12 PM 9/7/2022

3. Get Movie Detail by Title:
Method: GET, <http://127.0.0.1:5000/api/movietitle>

Home Workspaces API Network Explore Search Postman Invite Upgrade

Practice New Import GET Intro GET Retrieve All Data POST Create a New Movie GET Get Movie Detail by Title No Environment

mongoDB Practice / Get Movie Detail by Title

GET http://127.0.0.1:5000/api/Test Movie

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 12 ms Size: 619 B Save Response

Pretty Raw Preview Visualize JSON

```
1 1 Movie Documents with: Test Movie is Retrieved Successfully
2 {
3   "_id": {
4     "$oid": "6318fb0ca73f61915887f17e"
5   },
6   "title": "Test Movie",
7   "type": "Show",
8   "id": "tm84618",
9   "description": "This is sample movie for demo.",
10  "release_year": 2022,
11  "age_certification": "A",
12  "genres": [
13    "crime",
14    "drama"
15  ],
16  "production_countries": [
17    "US"
18  ],
19 }
```

Online Find and Replace Console Cookies Capture requests Bootcamp Runner Trash 84°F Sunny 3:12 PM 9/7/2022

4. Update Movie by Title:

Method: **PATCH**, <http://127.0.0.1:5000/api/movietitle>

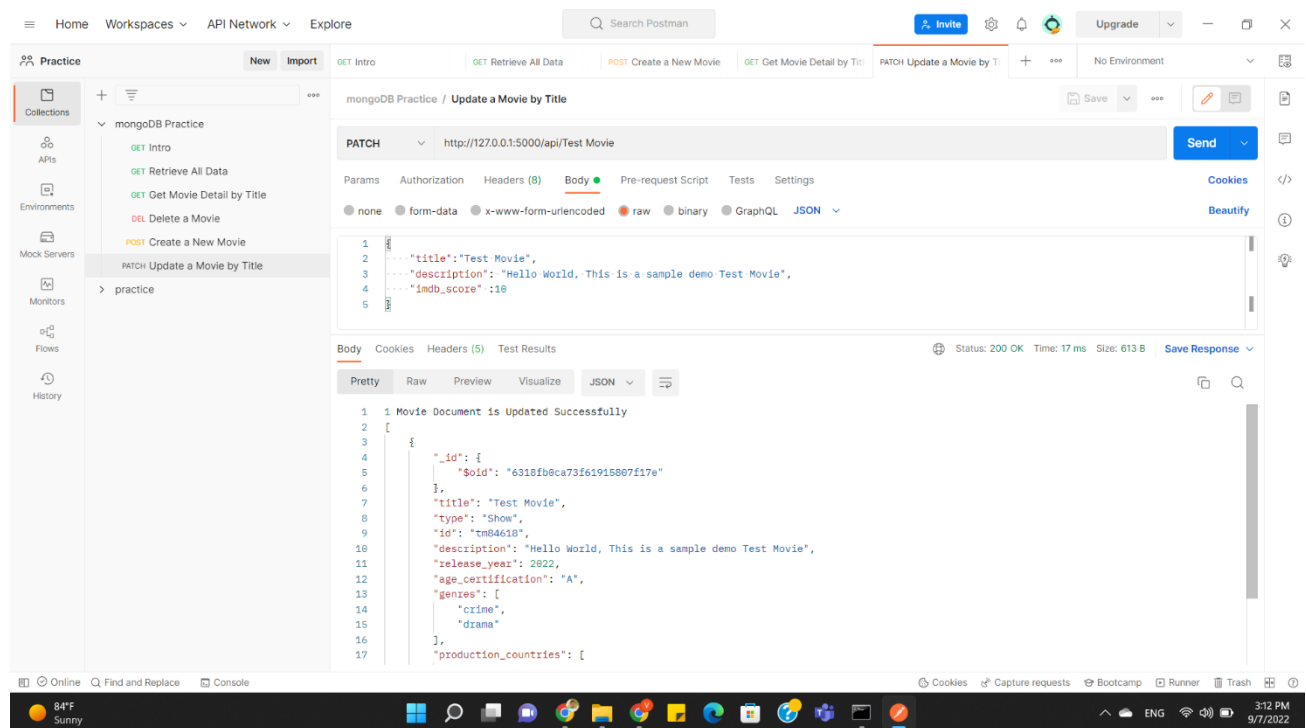
For update operation sample data, after inserting above documents - we are updating

use: Title: Test Movie -- [http://127.0.0.1:5000/api/Test Movie](http://127.0.0.1:5000/api/Test%20Movie)

by updating only title, description, imdb_score - please provide input in body - raw - json type

--

```
{
  "title": "Test Movie",
  "description": "Hello World, This is a sample demo Test Movie",
  "imdb_score": 10
}
```



5. Delete a movie document

Method: **DELETE**, <http://127.0.0.1:5000/api/movietitle>

Home Workspaces API Network Explore Search Postman Invite Upgrade

Practice New Import GET Intro GET Retrieve All Data POST Create a New Movie GET Get Movie Detail by Title PATCH Update a Movie **DEL Delete a Movie** No Environment

mongoDB Practice

- GET Intro
- GET Retrieve All Data
- GET Get Movie Detail by Title
- DEL Delete a Movie**
- POST Create a New Movie
- PATCH Update a Movie by Title

practice

mongoDB Practice / Delete a Movie

DELETE http://127.0.0.1:5000/api/Test Movie Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 15 ms Size: 584 B Save Response

Pretty Raw Preview Visualize JSON

```
1 Movie with title: "Test Movie" is Successfully Deleted.
2 {
3   "_id": {
4     "$oid": "6318fb8ca73f61915887f17e"
5   },
6   "title": "Test Movie",
7   "type": "Show",
8   "id": "tm84618",
9   "description": "Hello World. This is a sample demo Test Movie",
10  "release_year": 2022,
11  "age_certification": "A",
12  "genres": [
13    "crime",
14    "drama"
15  ],
16  "production_countries": [
17    "US"
```

-----XXX End of Document XXX-----