**Python API**

import pymongo

from flask import Flask, request, render\_template

from flask import jsonify

app = Flask(\_\_name\_\_, template\_folder='templates')

random\_title = ""

random\_tag = ""

@app.route('/tagName=<string:tag>')

def findUsingTagName(tag):

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client['admin']

collection = db['Movie1']

new\_pipeline = collection.find({"Tag": tag})

dict\_set = []

data\_age = { "Tag": tag}

dict\_set.append(data\_age)

for item in new\_pipeline:

print(item)

list\_item = [(k, v) for k, v in item.items()]

[id\_, TagId, Tag, UserId, MovieId, Loc, Age, Title, Genre,

Rating] = list\_item # item = {'\_id': ObjectId('5de5c90b94861803ab3ac3f7'), 'TagId': '1047332', 'Tag': 'New York', 'UserId': '2', 'MovieId': '1244', 'Location': 'Georgia', 'Age': '18-29', 'Title': 'Manhattan (1979)', 'Genres': 'Comedy|Drama|Romance', 'Rating': '0'}

data\_age = {

"Title": Title[1]

}

dict\_set.append(data\_age)

# random\_tag = str(Tag[1])

return jsonify(dict\_set)

@app.route('/movieTitle=<string:title>')

def findUsingTitle(title):

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client['admin']

collection = db['Movie1']

new\_pipeline = collection.find({"Title": title})

for item in new\_pipeline:

# print(item)

list\_item = [(k, v) for k, v in item.items()]

[id\_, TagId, Tag, UserId, MovieId, Loc, Age, Title, Genre,

Rating] = list\_item # item = {'\_id': ObjectId('5de5c90b94861803ab3ac3f7'), 'TagId': '1047332', 'Tag': 'New York', 'UserId': '2', 'MovieId': '1244', 'Location': 'Georgia', 'Age': '18-29', 'Title': 'Manhattan (1979)', 'Genres': 'Comedy|Drama|Romance', 'Rating': '0'}

random\_tag = str(Tag[1])

new\_set = []

new\_pipeline = collection.find({"Tag": random\_tag}).distinct("Title")

for item in new\_pipeline:

data = {"Title": item}

new\_set.append(data)

return jsonify(new\_set)

for item in new\_pipeline:

# print(item)

list\_item = [(k, v) for k, v in item.items()]

[id\_, TagId, Tag, UserId, MovieId, Loc, Age, Title, Genre,

Rating] = list\_item # item = {'\_id': ObjectId('5de5c90b94861803ab3ac3f7'), 'TagId': '1047332', 'Tag': 'New York', 'UserId': '2', 'MovieId': '1244', 'Location': 'Georgia', 'Age': '18-29', 'Title': 'Manhattan (1979)', 'Genres': 'Comedy|Drama|Romance', 'Rating': '0'}

data\_age = {

"Title": Title[1]

}

dict\_set.append(data\_age)

# print(random\_tag)

return jsonify(dict\_set)

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

def findUsingUserId():

if request.method == 'GET':

return render\_template("hd.html")

# Else if Request method is 'POST' then...

elif request.method == 'POST':

ID = id

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client['admin']

collection = db['Movie1']

if request.form['Submit'] == 'Submit':

query1 = request.form

beat = query1['UserId']

new\_pipeline = collection.find({"UserId": beat})

dict\_set = []

data\_age = {}

for item in new\_pipeline:

print(item)

list\_item = [(k, v) for k, v in item.items()]

[id\_, TagId, Tag, UserId, MovieId, Loc, Age, Title, Genre,

Rating] = list\_item # item = {'\_id': ObjectId('5de5c90b94861803ab3ac3f7'), 'TagId': '1047332', 'Tag': 'New York', 'UserId': '2', 'MovieId': '1244', 'Location': 'Georgia', 'Age': '18-29', 'Title': 'Manhattan (1979)', 'Genres': 'Comedy|Drama|Romance', 'Rating': '0'}

data\_age = {

"Title": Title[1],

"Age": Age[1]

}

dict\_set.append(data\_age)

random\_title = str(Title[1])

return jsonify(dict\_set)

if \_\_name\_\_ == "\_\_main\_\_":

app.debug = True

app.run()

**HTML**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<title>MovieLens</title>**

**</head>**

**<body>**

**<form id="Movie" method="POST" action="">**

**<!--First Case-->**

**<div class="panel-body">**

**<h2>1) Enter UserId</h2>**

**<!--Textbox for Beats-->**

**Enter a UserId: <input type="text" name="UserId"> <br><br>**

**<!--Submit button for first question-->**

**<input type="submit" name="Submit" value="Submit">**

**<br>**

**<br>**

**<br>**

**</div>**

**</form>**

**</body>**

**</html>**

**Readme**

**Html file opens a page with text box taking the input as UserId and has a submit button.**

**Imported pymongo package in python. Used flask API**

**The output is converted into json using jsonify function.**

**Loaded the data onto studio3t and mongodb instance.**

**The data is denormalized and loaded.**