**App.py**

from flask import Flask, request, render\_template

import pickle

import requests

import pandas as pd

from patsy import dmatrices

movies = pickle.load(open('model/movie.pkl', 'rb'))

similarity = pickle.load(open('model/similarity.pkl', 'rb'))

def fetch\_poster(movies\_id):

url = "https://api.themoviedb.org/3/movie/550?api\_key=f26f00586aa6ba538f35f141093402c4".format(movie\_id)

data = requests.get(url)

data = data.json()

poster\_path = data['poster\_path']

full\_path = "https://image.tmdb.org/t/p/w500/kqjL17yufvn9OVLyXYpvtyrFfak.jpg" + poster\_path

return full\_path

def recommend(movie):

index = [movies['title'] == movie].index[0]

distances = sorted(list(enumerate(similarity[index])), reverse= True, key=lambda x: x[1])

recommend\_movies\_name = []

recommend\_movies\_poster = []

for i in distances[1:6]:

movie\_id = movie.iloc[i[0]].movie\_id

recommend\_movies\_poster.append(fetch\_poster(movie\_id))

recommend\_movies\_name.append(movies.iloc[i[0]].title)

return recommend\_movies\_name, recommend\_movies\_poster

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template("index.html")

@app.route('/about')

def about():

return render\_template("about.html")

@app.route('/contact')

def contact():

return render\_template("contact.html")

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

def recommendation():

movies\_list = movies['title'].values

status = False

if request.method == "POST":

try:

if request.form:

movies\_name = request.form['movies']

print(movies\_name)

print(recommend\_movies\_poster)

recommend\_movies\_name,recommend\_movies\_poster = recommend(movies\_name)

print(recommend\_movies\_name)

status = True

return render\_template("prediction.html", movies\_name = recommend\_movies\_name, poster = recommend\_movies\_poster, movies\_list = movies\_list, status = status)

except Exception as e:

error = {'error':e}

return render\_template("prediction.html", error = error, movies\_list = movies\_list, status = status)

else:

return render\_template("prediction.html", movies\_list = movies\_list, status = status)

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

app.debug = True

app.run()