from flask import Blueprint

from backend.auth\_middleware import token\_required

import ibm\_boto3

from ibm\_botocore.client import Config, ClientError

from backend import config

cos = ibm\_boto3.resource("s3",

ibm\_api\_key\_id=config["COS\_API\_KEY\_ID"],

ibm\_service\_instance\_id=config["COS\_INSTANCE\_CRN"],

config=Config(signature\_version="oauth"),

endpoint\_url=config["COS\_ENDPOINT"]

)

files = Blueprint("files", \_\_name\_\_)

def multi\_part\_upload(bucket\_name, item\_name, file\_path):

try:

print("Starting file transfer for {0} to bucket: {1}\n".format(

item\_name, bucket\_name))

# set 5 MB chunks

part\_size = 1024 \* 1024 \* 5

# set threadhold to 15 MB

file\_threshold = 1024 \* 1024 \* 15

# set the transfer threshold and chunk size

transfer\_config = ibm\_boto3.s3.transfer.TransferConfig(

multipart\_threshold=file\_threshold,

multipart\_chunksize=part\_size

)

# the upload\_fileobj method will automatically execute a multi-part upload

# in 5 MB chunks for all files over 15 MB

with open(file\_path, "rb") as file\_data:

cos.Object(bucket\_name, item\_name).upload\_fileobj(

Fileobj=file\_data,

Config=transfer\_config

)

print("Transfer for {0} Complete!\n".format(item\_name))

except ClientError as be:

print("CLIENT ERROR: {0}\n".format(be))

except Exception as e:

print("Unable to complete multi-part upload: {0}".format(e))

@files.route('/avatar', methods=["POST"])

@token\_required

def upload\_profile\_photo(current\_user):

return "hello"