from flask\_restful import Resource;

from flask import request,after\_this\_request;

from utils.cookieChecker import token\_required

from utils.dbQuery import selectQuery,insertQuery

class Bookamark(Resource):

@token\_required

def get(email,self):

bookmarks\_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE EMAIL=?',(email,))['BOOKMARKS']

bookmarks\_id=bookmarks\_id.split(',')

bookmarks=[]

if(bookmarks\_id==['']):

bookmarks\_id=[]

for x in bookmarks\_id:

data=selectQuery('SELECT DATA FROM BOOKMARK WHERE ID=?',(x,))['DATA']

bookmarks.append(data)

resp={"data":bookmarks,"id":bookmarks\_id}

return resp,200

@token\_required

def post(email,self):

req=request.json

news=req["news"]

# Query to check previous inserted

id=selectQuery('SELECT ID FROM BOOKMARK WHERE DATA=?',(news,))

if(id==False):

insertQuery('INSERT INTO BOOKMARK (DATA) VALUES (?)',(news,))

id=selectQuery('SELECT ID FROM BOOKMARK WHERE DATA=?',(news,))['ID']

else:

id=id['ID']

new\_bookmarks\_id=[]

bookmarks\_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE EMAIL=?',(email,))

if(bookmarks\_id==False):

bookmarks\_id=[]

else:

bookmarks\_id=bookmarks\_id['BOOKMARKS']

bookmarks\_id=bookmarks\_id.split(',')

if(bookmarks\_id==['']):

bookmarks\_id=[]

insertCurr=True

for x in bookmarks\_id:

new\_bookmarks\_id.append(x)

if(int(x)==id):

insertCurr=False

if(insertCurr):

new\_bookmarks\_id.append(str(id))

x=",".join([str(i) for i in new\_bookmarks\_id])

@after\_this\_request

def inserter(response):

insertQuery('UPDATE USER SET BOOKMARKS=? WHERE EMAIL=?',(x,email))

return response

return {"status":"inserted"},200

class UnBookMark(Resource):

@token\_required

def post(email,self):

req=request.json;

id=req["id"]

bookmarks\_id=selectQuery('SELECT BOOKMARKS FROM USER WHERE EMAIL=?',(email,))['BOOKMARKS']

bookmarks\_id=bookmarks\_id.split(',')

new\_bookmarks=[]

for x in bookmarks\_id:

if(x!=id):

new\_bookmarks.append(x)

if(new\_bookmarks==[]):

new\_bookmarks=''

new\_bookmarks=",".join([str(i) for i in new\_bookmarks])

@after\_this\_request

def inserter(response):

insertQuery('UPDATE USER SET BOOKMARKS=? WHERE EMAIL=?',(new\_bookmarks,email))

return response

return {"status":"deleted"},200