**SOURCE CODE**

**Course-end Project 2: Real-time Data Management**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PRODUCER LAMBDA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import json

import boto3

import uuid

def lambda\_handler(event, context):

client = boto3.client('kinesis')

response = client.put\_record(

StreamName='Simplilearn',

Data=json.dumps(event),

PartitionKey=str(uuid.uuid4())

)

print(response)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TEST EVENT FOR PRODUCER LAMBDA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"id": "b2bfe236-c738-4916-9015-ce47d49d666d",

"canFly": true,

"name": "blue jay",

"colors": [

"black",

"white",

"blur",

"grey"

]

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CONSUMER LAMBDA\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import base64

import json

import boto3

def lambda\_handler(event, context):

for record in event['Records']:

#Kinesis data is base64 encoded so decode here

payload=base64.b64decode(record["kinesis"]["data"])

res = json.loads(payload)

write\_to\_db(res)

print("Object successfully stored in DB.")

def write\_to\_db(data):

dynamodb = boto3.resource('dynamodb', region\_name="us-east-1")

table = dynamodb.Table("Simplilearn-table")

table.put\_item(

Item=data

)