**lambda function for add items to cart**

import json

import psycopg2

import os

def lambda\_handler(event, context):

# Extract necessary parameters from event

user\_id = event['user\_id']

product\_id = event['product\_id']

quantity = event['quantity']

# Connect to RDS PostgreSQL database

conn = psycopg2.connect(

host=os.environ['DB\_HOST'],

port=os.environ['DB\_PORT'],

user=os.environ['DB\_USER'],

password=os.environ['DB\_PASSWORD'],

database=os.environ['DB\_DATABASE']

)

**# Insert cart item into database**

cur = conn.cursor()

cur.execute(

"INSERT INTO cart\_item (user\_id, product\_id, quantity) VALUES (%s, %s, %s)",

(user\_id, product\_id, quantity)

)

conn.commit()

# Return success response

response = {

'statusCode': 200,

'body': json.dumps({'message': 'Item added to cart successfully'})

}

return response

**lambda function for remove items from cart**

import json

import psycopg2

import os

def lambda\_handler(event, context):

# Extract necessary parameters from event

user\_id = event['user\_id']

product\_id = event['product\_id']

# Connect to RDS PostgreSQL database

conn = psycopg2.connect(

host=os.environ['DB\_HOST'],

port=os.environ['DB\_PORT'],

user=os.environ['DB\_USER'],

password=os.environ['DB\_PASSWORD'],

database=os.environ['DB\_DATABASE']

)

# Delete cart item from database

cur = conn.cursor()

cur.execute(

"DELETE FROM cart\_item WHERE user\_id = %s AND product\_id = %s",

(user\_id, product\_id)

)

conn.commit()

# Return success response

response = {

'statusCode': 200,

'body': json.dumps({'message': 'Item removed from cart successfully'})

}

return response

**lambda function for view cart**

import json

import psycopg2

import os

def lambda\_handler(event, context):

# Connect to PostgreSQL database

conn = psycopg2.connect(

host=os.environ['DB\_HOST'],

port=os.environ['DB\_PORT'],

dbname=os.environ['DB\_NAME'],

user=os.environ['DB\_USER'],

password=os.environ['DB\_PASSWORD']

)

# Retrieve cart items for the given user ID

user\_id = event['user\_id']

cur = conn.cursor()

cur.execute("SELECT \* FROM cart WHERE user\_id=%s", (user\_id,))

cart\_items = cur.fetchall()

# Convert result set to JSON format

cart\_data = []

for item in cart\_items:

item\_dict = {

'product\_id': item[0],

'quantity': item[1]

}

cart\_data.append(item\_dict)

response = {

'statusCode': 200,

'body': json.dumps(cart\_data)

}

return response

**lambda function for checkout**

import json

import psycopg2

import os

def lambda\_handler(event, context):

# Connect to PostgreSQL database

conn = psycopg2.connect(

host=os.environ['DB\_HOST'],

port=os.environ['DB\_PORT'],

dbname=os.environ['DB\_NAME'],

user=os.environ['DB\_USER'],

password=os.environ['DB\_PASSWORD']

)

# Retrieve cart items for the given user ID

user\_id = event['user\_id']

cur = conn.cursor()

cur.execute("SELECT \* FROM cart WHERE user\_id=%s", (user\_id,))

cart\_items = cur.fetchall()

# Insert cart items into orders table

cur.execute("INSERT INTO orders (user\_id, order\_items) VALUES (%s, %s)", (user\_id, json.dumps(cart\_items)))

conn.commit()

# Clear cart for the given user ID

cur.execute("DELETE FROM cart WHERE user\_id=%s", (user\_id,))

conn.commit()

response = {

'statusCode': 200,

'body': json.dumps({'message': 'Checkout successful'})

}

return response

**lambda function codes for Order tracking**

import os

import psycopg2

def lambda\_handler(event, context):

# Extract order\_id from the event

order\_id = event['order\_id']

# Connect to the PostgreSQL database

conn = psycopg2.connect(

host=os.environ['DB\_HOST'],

port=os.environ['DB\_PORT'],

dbname=os.environ['DB\_NAME'],

user=os.environ['DB\_USER'],

password=os.environ['DB\_PASSWORD']

)

cursor = conn.cursor()

# Execute SQL query to retrieve order tracking data

cursor.execute('SELECT status, date FROM order\_tracking WHERE order\_id = %s', (order\_id,))

rows = cursor.fetchall()

# Convert results to a dictionary

results = []

for row in rows:

results.append({'status': row[0], 'date': row[1].strftime('%Y-%m-%d %H:%M:%S')})

# Close the database connection

cursor.close()

conn.close()

# Return the results as a JSON object

return {'order\_id': order\_id, 'tracking\_data': results}

**implementation of the get\_user\_info function in Python that includes error handling:**

import requests

def get\_user\_info(user\_id):

try:

response = requests.get(f'https://example.com/users/{user\_id}')

response.raise\_for\_status()

return response.json()

except requests.exceptions.HTTPError as e:

if e.response.status\_code == 404:

print(f'User with ID {user\_id} not found')

else:

print(f'Server error: {e}')

except requests.exceptions.RequestException as e:

print(f'Request error: {e}')