# Hourly Attendance Monitoring System

## 1.Introduction

### 1.1 Overview

In this project we will building smart hourly attendance monitoring system using Amazon Web services , such as Dynamo DB , Amazon Rekognition and many others . By building this system we can save plenty of time and efforts compared to traditional method of attendance monitoring system.

### 1.2 Purpose

The main purpose of building this system , is to save time and efforts that is spent by thousands of people all over the world . This system also prevent attendance proxy with better accuracy and at the same time by consuming less efforts **.**

## 2 . Literature Survey :

### 2.1 Existing Problem :

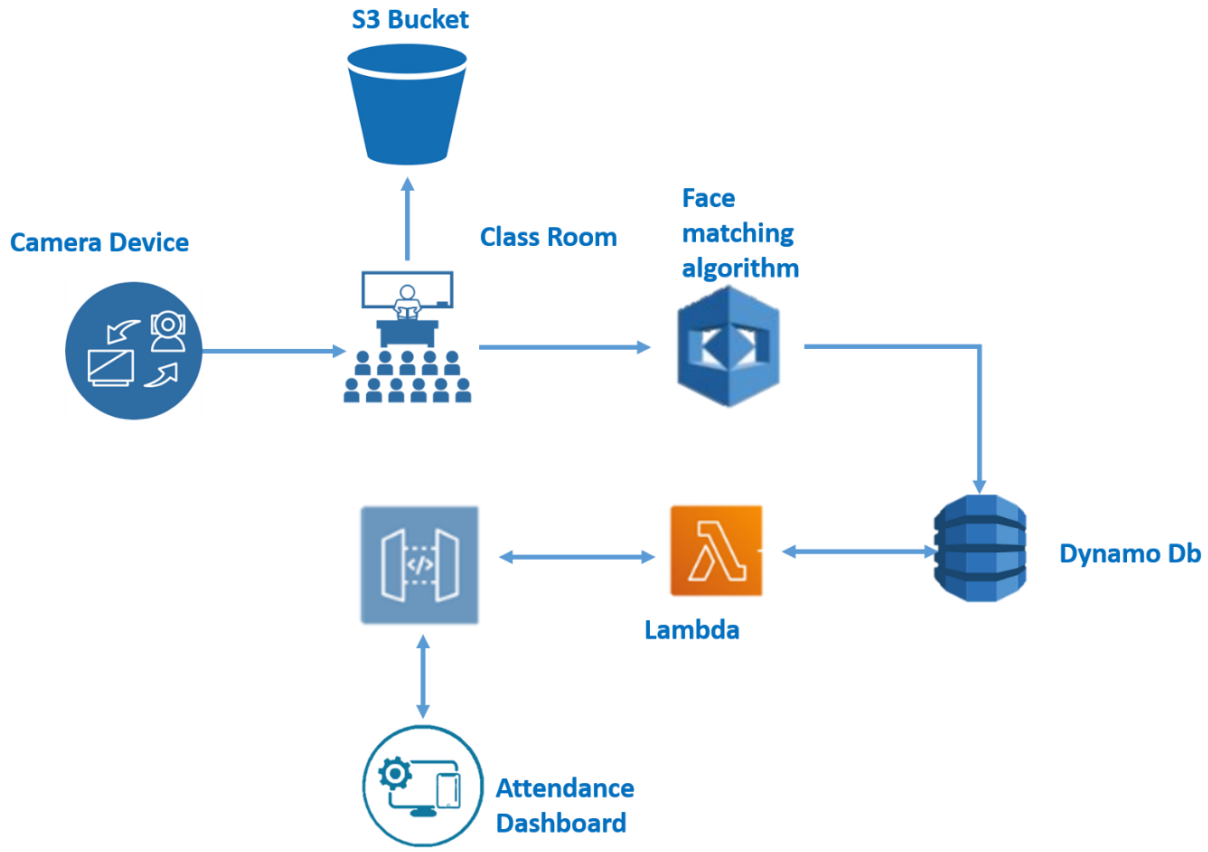
Till now in many schools teachers were taking attendance using a notebook and pen for every hour , later they consolidate the attendance of all the students in another notebook for each month . This process continues every year which is a huge time consuming and tedious process . Nowadays teachers were taking attendance manually using a mobile app which is also again a time consuming process . Hence I have developed a system which could solve these problem effectively .

## 2.1 Proposed Solution

Maintaining attendance is essential in all the institutes for checking the attendance percentage of Students. Every institute has its method in this regard. Some are taking attendance manually on the register for every hour and later they will upload every hour data of a class to the server or file-based approach and some have adopted methods of automatic attendance using some biometric techniques. However, these methods are inefficient and time-consuming, AI can definitely find a solution to this problem.

# 3 Theoretical Analysis

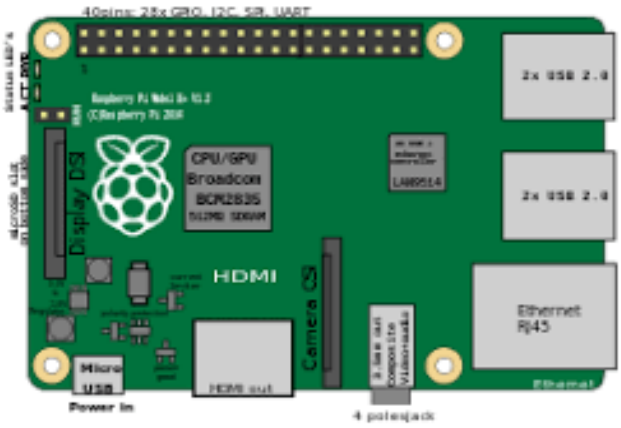
## 3.1 Block Diagram



## 3.2 Hardware/Software Designing

### Hardware

When we are implementing this project in real time we should use a Raspberry pi and a CCTV Camera . This camera will camera will be connected to raspberry pi and the opencv code and the Amazon rekognition will be made to run in the raspberry pi .

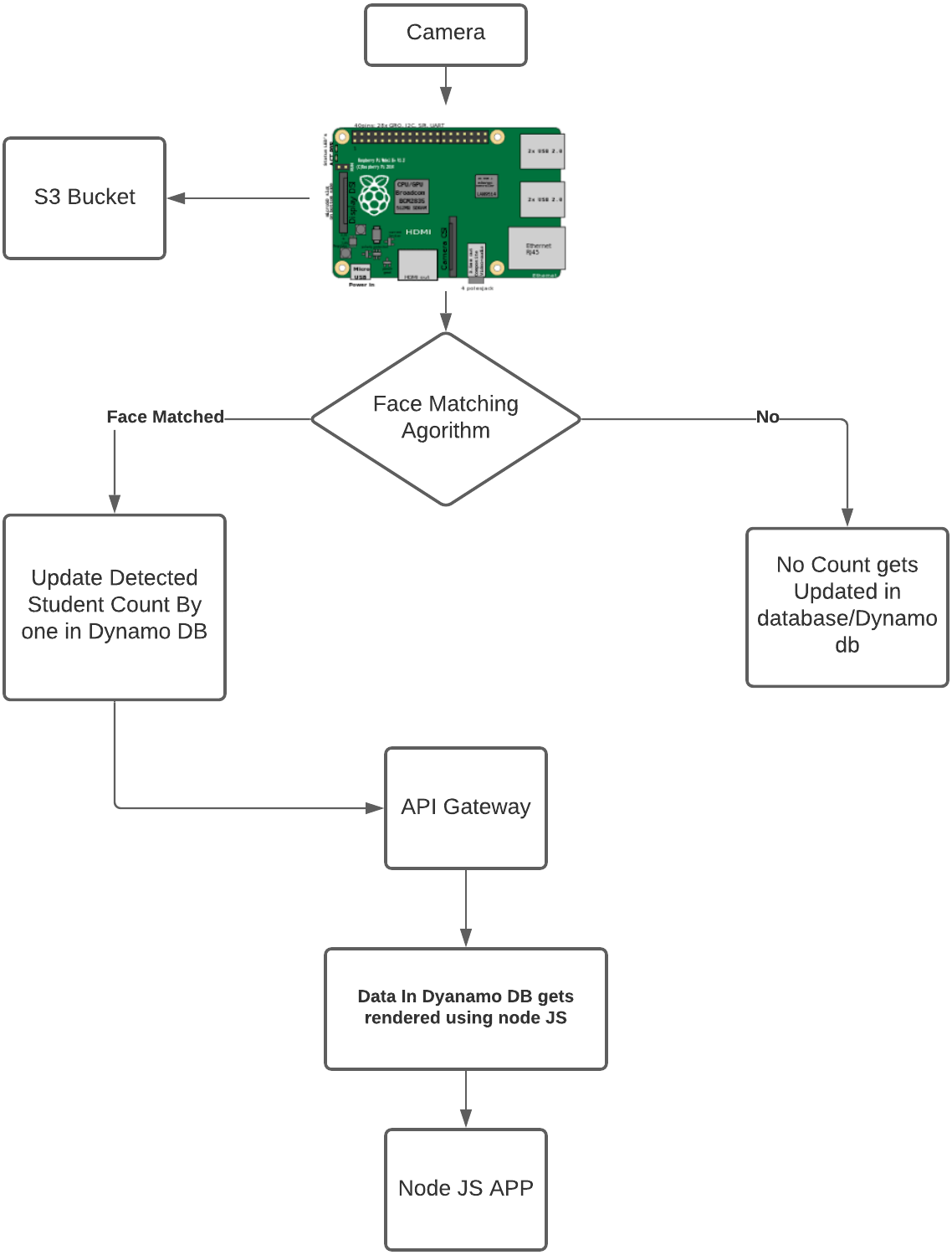



### Software

Software designing is done using Amazon Web Services as

- Amazon Rekognition
- Dynamo DB
- Lambda function
- S3 bucket
- API Gateway
- web frameworks such as Node js .

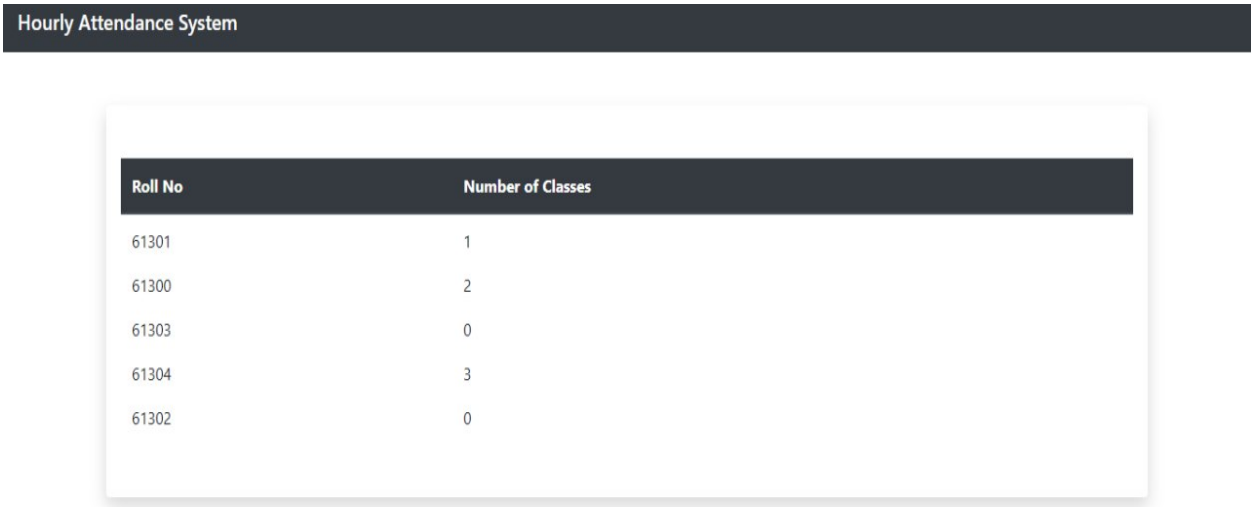
# 5.Flowchart


Camera
S3 Bucket
Face Matching Agorithm
Face Matched
No
Update Detected Student Count By one in Dynamo DB
No Count gets Updated in database/Dynamo db
API Gateway
Data In Dyanamo DB gets rendered using node JS
Node JS APP

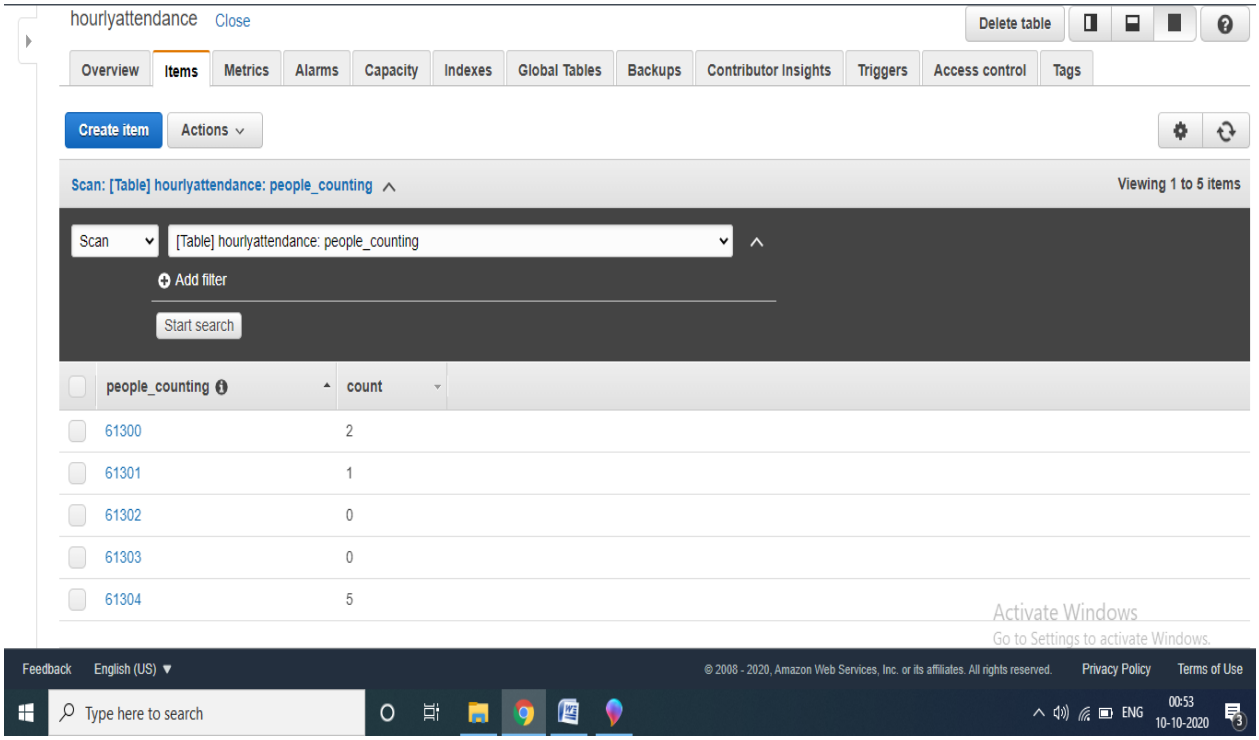
## 6. Result :

Below is the images of the output that I obtained

## NodeJS Dashboard

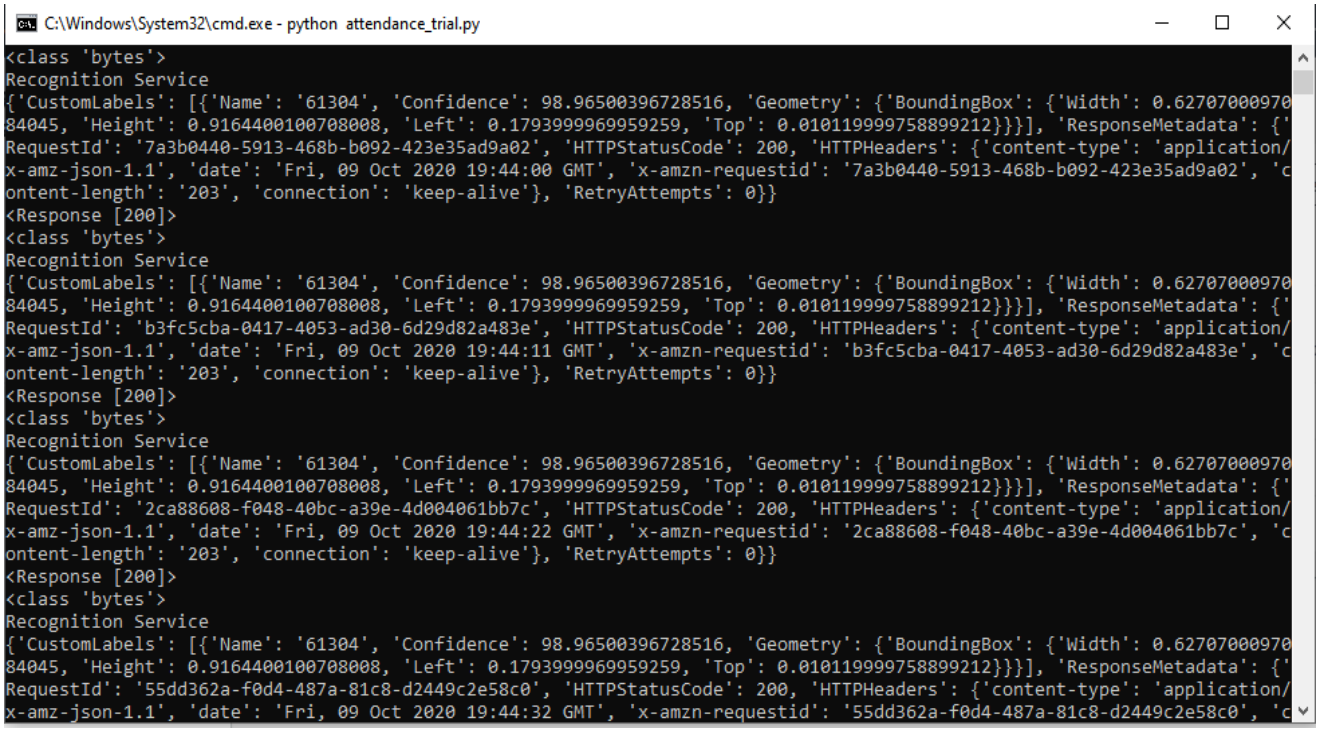



## DynamoDB Table:



## Command Prompt Output :

Output printed in command prompt for every one hour




## 7. Advantages and Disadvantages:

## Advantages:

- Less time consumption
- No need of manual operation
- Prevents attendance proxy
- User friendly process
- More efficient approach

## Disadvantages

- Initial high cost

- Require skilled labour for maintenance in software as well as hardware side

## 8.Applications:

Though this hourly attendance system focuses mainly on capturing attendance in schools and colleges , the behind this idea can be used for many application such as

- workplace safety monitoring
- Replacing biometric attendance scheme by facial rekognition method
- Used in police investigation for comparing ac quest face with faces stored in database .
- Used in healthcare field .
- More useful in Industry 4.0 concepts

## 9.Conclusion:

This is type of smart hourly attendance monitoring system will be very useful in monitoring attendance in school and college without human intervention and consumes great time as well as efforts . Though the initial cost is high it will more efficient and user friendly than conventional method .

## 10.Future Scope:

Though this system will be useful in school and colleges , but the concept behind this idea can be applicable various sectors such as agriculture(animal intervention can be detected) , healthcare(Identifying whether the staff or patients are wearing mask  or not), Industry 4.0(Workplace safety monitoring) and

many other sectors also . Hence this type of concept has huge scope in future.

# 11.Bibliography

# Appendix

## A.Source Code

### AWS Rekognition Code

```
import numpy as np
import cv2
import time
import boto3
import requests

while(True):


    client=boto3.client('rekognition',
                        aws_access_key_id="AxxxxxxxxxxxxxxxxxxN",

aws_secret_access_key="PX1UxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxZ",

aws_session_token="FwoGZXIvxxxxxxxxxxxxxxxxxxx7gD4P8wobiahxyBFQFxxtYrvtZ
UjygySu0tQOm0U+fi1TI9XXTzLy+a2X6zbkHH+SBdxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxT
cFbwVih5Bgy44LvxxxxxxxxxxxxxxxxxxxxxPF+qSidYQxGxxxxxxxxxxxxxxxtbRdHz9Y/R2
r6gQvaoAU7y4mqYDvETxxxxxxxxxxxxxxxxxxxxxxo5fCC/AUyLXa3WwQCXOG0PZ3",
                        region_name='us-east-1')

    with open(r'E:\Co-curricular\Smartintern AI project\cam.jpg','rb') as
source_image:
        source_bytes=source_image.read()
    print(type(source_bytes))

    print("Recognition Service")
    response = client.detect_custom_labels(

ProjectVersionArn='arn:aws:rekognition:us-east-1xxxxxxxxxxxx:project/xxx
xxxxxx/version/xxxxxxxxxxxxxxxxxxxx.xxxxxxxx/xxxxx',

    Image={
        'Bytes':source_bytes

```

```
    },

    )

    print(response)
    if not len(response['CustomLabels']):
        print('Animal not identified')

    else:
        str=response['CustomLabels'][0]['Name']

url="https://i36bw98emf.execute-api.us-east-1.amazonaws.com/people_counting?rollno="+str
        resp = requests.get(url)
        print(resp)
//time.sleep(60*60)- it triggers amazon rekognition for every 1 hour
    time.sleep(60*60);

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

```

## Opencv Code to capture image Every one hour :

```
import numpy as np
import cv2
import time

while(True):

    cap = cv2.VideoCapture(0)
    framerate = cap.get(5)

    # Capture frame-by-frame
    ret, frame = cap.read()
    cap.release()
    # Our operations on the frame come here

```

```
15
16
17      filename = 'E:/Co-curricular/Smartintern AI project/image.png'
18
19      cv2.imwrite(filename, frame)
20  //time.sleep(60*60) - capture image for every 1 hour
21      time.sleep(60*60)
22      if cv2.waitKey(1) & 0xFF == ord('q'):
23          break
24
25  # When everything done, release the capture
26  cap.release()
27  cv2.destroyAllWindows()
28
```

## AWS Lambda function code(Python)

```
1   import json
2   import boto3
3
4   dynamo= boto3.resource('dynamodb')
5   table = dynamo.Table('hourlyattendance')
6
7   def lambda_handler(event, context):
8       resp =
    table.get_item(Key={"people_counting":event['people_counting']})
9       print(resp['Item']['count'])
10      count = resp['Item']['count']
11      count= count+1
12      inp = {"people_counting":event['people_counting'],"count":count}
13      table.put_item(Item=inp)
14      # TODO implement
15      # table.put_item(Item=event)
16      return "successful"
```

## Reading data from DynamoDB Code(NodeJS)

```
1   console.log('starting function');
2
3   const AWS = require('aws-sdk');
4   const docClient = new AWS.DynamoDB.DocumentClient({region:
```

```
    'us-east-1'});
5
6   exports.handler = function(e, ctx, callback){
7
8       var scanningParameters = {
9           TableName: 'hourlxxxxxxxxxx',
10          Limit: 100
11      };
12
13      docClient.scan(scanningParameters,function(err, data){
14          if(err){
15              callback(err, null);
16          }else{
17              callback(null, data);
18          }
19      });
20  }
```