CLOUD COMPUTING MINI PROJECT

Group Details:

1. Ishita Hardasmalani – 2103058
2. Varun Jajoo – 2103069
3. Dimple Hemnani - 2103059

Project Title : Facial Recognition System on AWS

Problem:

Many organizations face challenges in implementing effective access control systems that strike a balance between security, convenience, and scalability. Traditional methods such as keycards or passwords are prone to security breaches, while manual verification processes can be time-consuming and error-prone. Additionally, as organizations grow or evolve, scalability becomes a concern, with traditional systems often struggling to accommodate changing needs. Our proposed solution is to develop a Facial Recognition System on AWS that integrates various AWS services to provide secure access control. This system will allow registered employees and authorized visitors to gain access to designated areas based on their facial biometrics.

Requirement Specification:

1. Image Storage and Management:

- Utilize Amazon S3 buckets to securely images.

- Implement versioning and access control policies to manage image data effectively.

2. Facial Recognition Engine:

- Leverage Amazon Rekognition for accurate and real-time facial recognition.

- Train the system to recognize authorized personnel and distinguish them from unauthorized individuals.

3. Identity Management and Authentication:

- Maintain a DynamoDB database to store employee and visitor metadata, including identity information and access permissions.

- Implement Lambda functions for registration processes, integrating with DynamoDB and Amazon Rekognition.

4. Access Control and Logging:

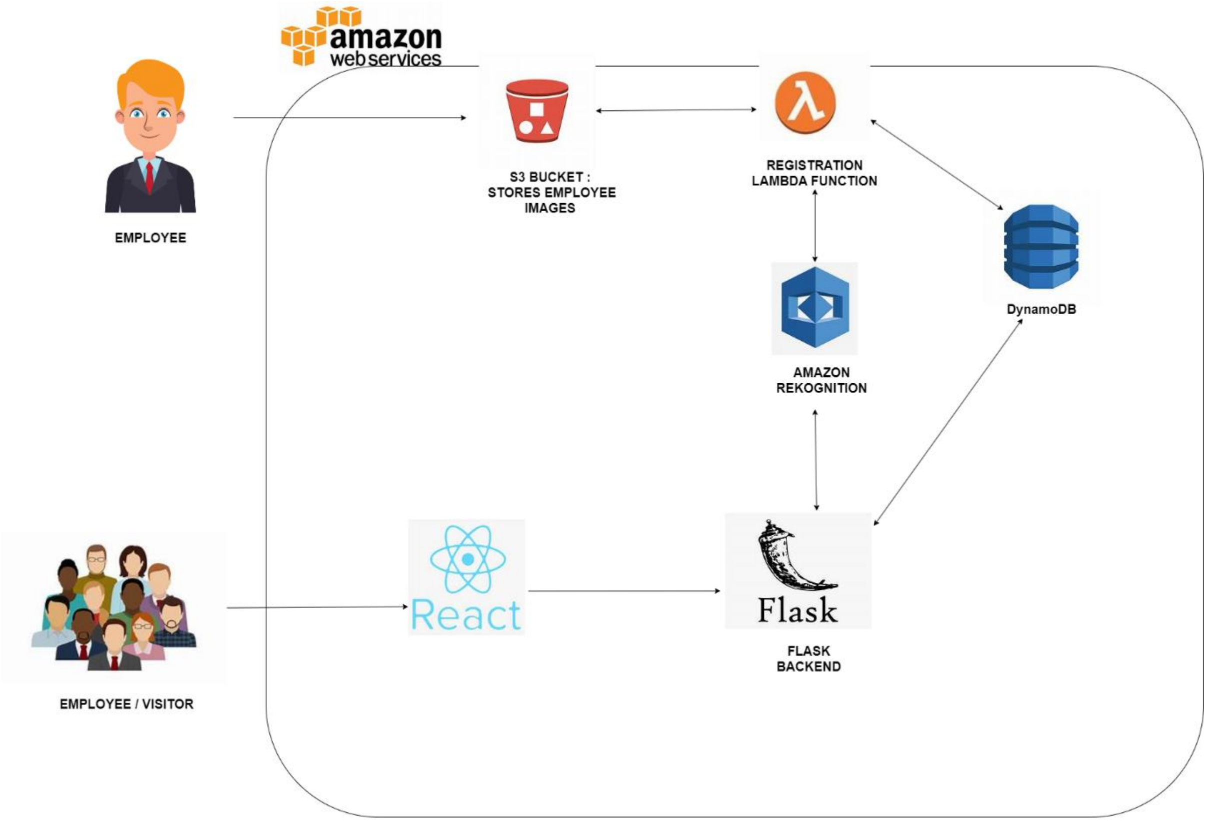
- Utilize AWS IAM roles and policies to enforce least privilege access and restrict unauthorized actions.

- Log access attempts and system activities using Amazon CloudWatch for auditing and compliance purposes.

5. User Interface:

- Develop a user-friendly front-end application using ReactJS to facilitate registration, authentication , and system monitoring.

Block Diagram:



Main Code:

Lambda Function: *lambda.py*

```
from _future_ import print_function
import boto3
from decimal import Decimal
import json
import urllib

print('Loading function')

dynamodb = boto3.client('dynamodb')
s3 = boto3.client('s3')
rekognition = boto3.client('rekognition')

def index_faces(bucket, key):
 response = rekognition.index_faces(
       Image={"S3Object":
          {"Bucket": bucket,
          "Name": key}},
          CollectionId="famouspersons")
   return response

def update_index(tableName,faceId, fullName):
   response = dynamodb.put_item(
       TableName=tableName,
       Item={
          'RekognitionId': {'S': faceId},
          'FullName': {'S': fullName}
          }
       )
def lambda_handler(event, context):
   bucket = event['Records'][0]['s3']['bucket']['name']
   print("Records: ",event['Records'])
   key = event['Records'][0]['s3']['object']['key']
   print("Key: ",key)
   # key = key.encode()
   # key = urllib.parse.unquote_plus(key)

   try:
        # Calls Amazon Rekognition IndexFaces API to detect faces in S3 object
       # to index faces into specified collection
        response = index_faces(bucket, key)
         # Commit faceId and full name object metadata to DynamoDB
          if response['ResponseMetadata']['HTTPStatusCode'] == 200:
          faceId = response['FaceRecords'][0]['Face']['FaceId']
           ret = s3.head_object(Bucket=bucket,Key=key)
          personFullName = ret['Metadata']['fullname']
          update_index('face_recognition',faceId,personFullName)
       print(response)
      return response
   except Exception as e:
       print(e)
       print("Error processing object {} from bucket {}. ".format(key, bucket))
       raise e
```

Flask Backend : *testing.py*

```
from flask import Flask, request, jsonify
from flask_cors import CORS
import boto3
import io
from PIL import Image
app = Flask(__name__)
CORS(app)  # Add CORS to your Flask app
rekognition = boto3.client('rekognition', region_name='ap-south-1')
dynamodb = boto3.client('dynamodb', region_name='ap-south-1')

@app.route('/upload_photo', methods=['POST'])
def upload_photo():
   if 'file' not in request.files:
      return jsonify({'error': 'No file part'})

   file = request.files['file']

   if file.filename == '':
      return jsonify({'error': 'No selected file'})

   try:
      image = Image.open(file)
      stream = io.BytesIO()
      image.save(stream, format="JPEG")
      image_binary = stream.getvalue()
```

```
        response = rekognition.search_faces_by_image(
            CollectionId='famouspersons',
            Image={'Bytes': image_binary}
        )

        found = False
        response_data = {'matches': []}
        for match in response['FaceMatches']:
            face = dynamodb.get_item(
                TableName='face_recognition',
                Key={'RekognitionId': {'S': match['Face']['FaceId']}}
            )

            if 'Item' in face:
                response_data['matches'].append({
                    'faceId': match['Face']['FaceId'],
                    'confidence': match['Face']['Confidence'],
                    'fullName': face['Item']['FullName']['S']
                })
                found = True

        if found:
            return jsonify(response_data)
        else:
            return jsonify({'error': 'Person cannot be recognized'})

    except Exception as e:
        return jsonify({'error': str(e)})
if __name__ == '__main__':
    app.run(debug=True)
```

putimages.py

```
import boto3
s3 = boto3.resource('s3')
# Get list of objects for indexing
images=[
        ('varun.jpg','Varun Jajoo'),
       ('ishita.jpg','Ishita Hardasmalani')
  ]

# Iterate through list to upload objects to S3
for image in images:
    file = open(image[0],'rb')
    object = s3.Object('famouspersonsvarun','index/'+ image[0])
    ret = object.put(Body=file,
                    Metadata={'FullName':image[1]})
```

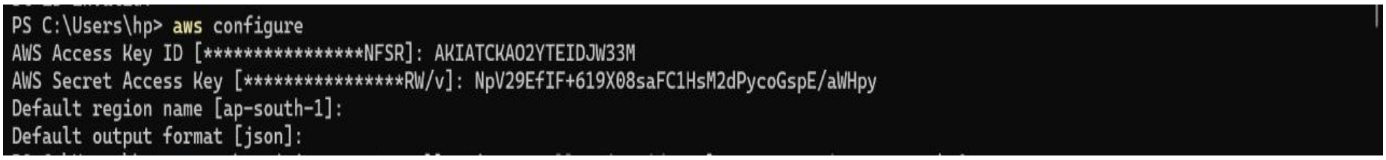
testing.py

```
import boto3
s3 = boto3.resource('s3')
# Get list of objects for indexing
images=[('image1.jpg','Elon Musk'),
      ('image2.jpg','Elon Musk'),
      ('image3.jpg','Bill Gates'),
      ('image4.jpg','Bill Gates'),
      ('image5.jpg','Sundar Pichai'),
      ('image6.jpg','Sundar Pichai')
      ]

# Iterate through list to upload objects to S3
for image in images:
    file = open(image[0],'rb')
    object = s3.Object('famouspersons-images','index/'+ image[0])
    ret = object.put(Body=file,
                    Metadata={'FullName':image[1]})
```

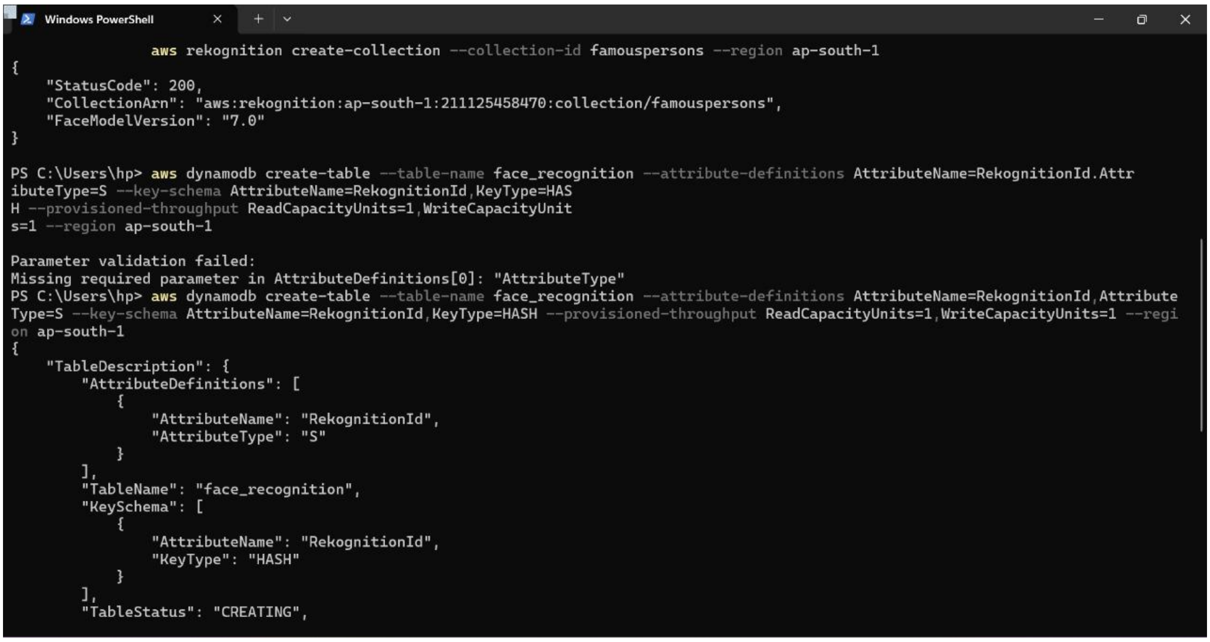
Screenshots:

1. Connecting AWS CLI with the AWS account



```
PS C:\Users\hp> aws configure
AWS Access Key ID [****************NFSR]: AKIATCKAO2YTEIDJW33M
AWS Secret Access Key [****************RW/v]: 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
Default region name [ap-south-1]:
Default output format [json]:
```

2. Creation of DynamoDB Table

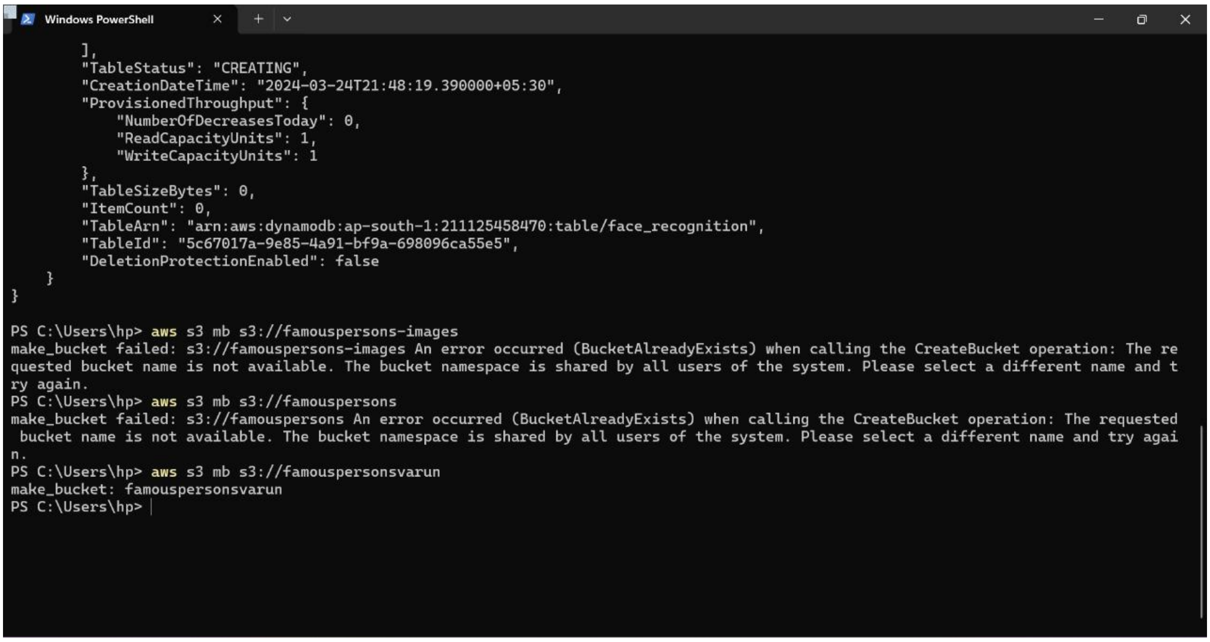


```
                aws rekognition create-collection --collection-id famouspersons --region ap-south-1
{
    "StatusCode": 200,
    "CollectionArn": "aws:rekognition:ap-south-1:211125458470:collection/famouspersons",
    "FaceModelVersion": "7.0"
}

PS C:\Users\hp> aws dynamodb create-table --table-name face_recognition --attribute-definitions AttributeName=RekognitionId.Attr
ibuteType=S --key-schema AttributeName=RekognitionId,KeyType=HAS
H --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnit
s=1 --region ap-south-1

Parameter validation failed:
Missing required parameter in AttributeDefinitions[0]: "AttributeType"
PS C:\Users\hp> aws dynamodb create-table --table-name face_recognition --attribute-definitions AttributeName=RekognitionId,Attribute
Type=S --key-schema AttributeName=RekognitionId,KeyType=HASH --provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1 --regi
on ap-south-1
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "RekognitionId",
                "AttributeType": "S"
            }
        ],
        "TableName": "face_recognition",
        "KeySchema": [
            {
                "AttributeName": "RekognitionId",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "CREATING",
```

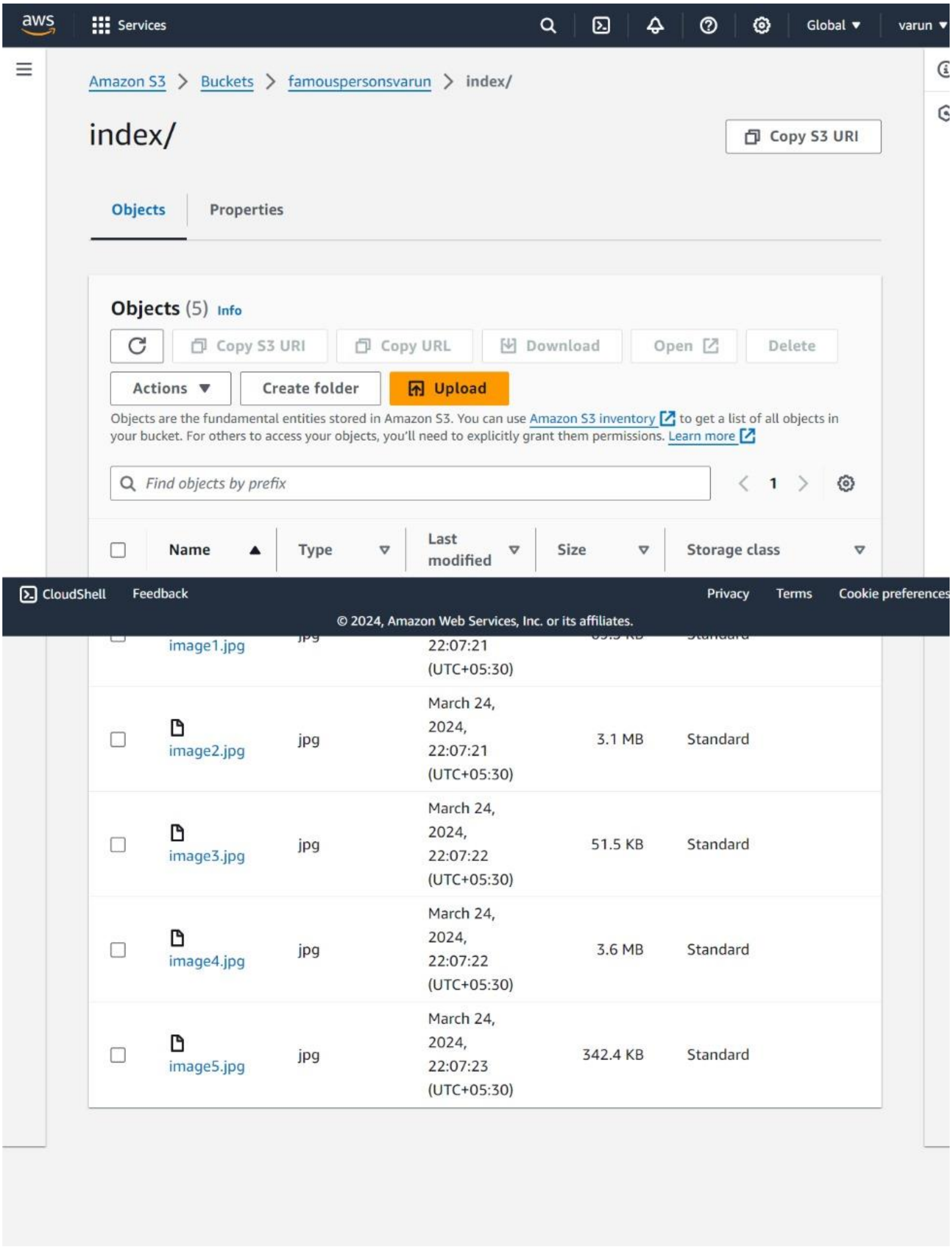
3.Creation of S3 Buckets



```
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2024-03-24T21:48:19.390000+05:30",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 1,
            "WriteCapacityUnits": 1
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:ap-south-1:211125458470:table/face_recognition",
        "TableId": "5c67017a-9e85-4a91-bf9a-698096ca55e5",
        "DeletionProtectionEnabled": false
    }
}

PS C:\Users\hp> aws s3 mb s3://famouspersons-images
make_bucket failed: s3://famouspersons-images An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The re
quested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and t
ry again.
PS C:\Users\hp> aws s3 mb s3://famouspersons
make_bucket failed: s3://famouspersons An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The requested
 bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try agai
n.
PS C:\Users\hp> aws s3 mb s3://famouspersonsvarun
make_bucket: famouspersonsvarun
PS C:\Users\hp>
```

4. Contents of S3 Bucket



Amazon S3 > Buckets > famouspersonsvarun > index/

index/

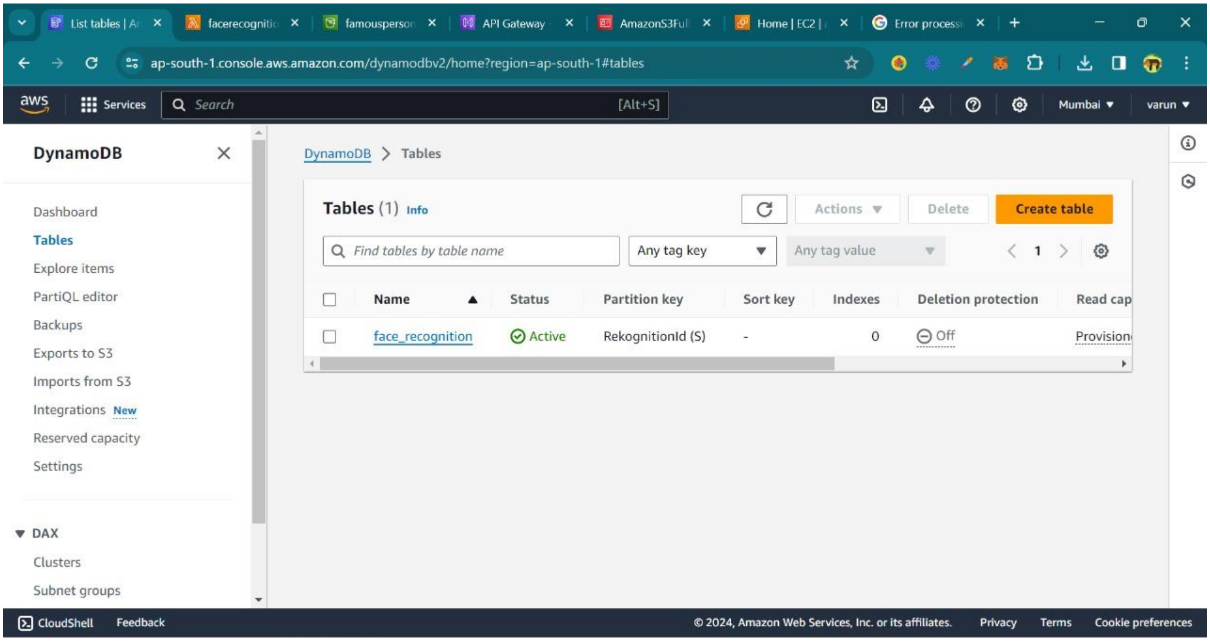
Objects (5) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	image1.jpg	jpg	22:07:21 (UTC+05:30)	[illegible]	Standard
□	image2.jpg	jpg	March 24, 2024, 22:07:21 (UTC+05:30)	3.1 MB	Standard
□	image3.jpg	jpg	March 24, 2024, 22:07:22 (UTC+05:30)	51.5 KB	Standard
□	image4.jpg	jpg	March 24, 2024, 22:07:22 (UTC+05:30)	3.6 MB	Standard
□	image5.jpg	jpg	March 24, 2024, 22:07:23 (UTC+05:30)	342.4 KB	Standard

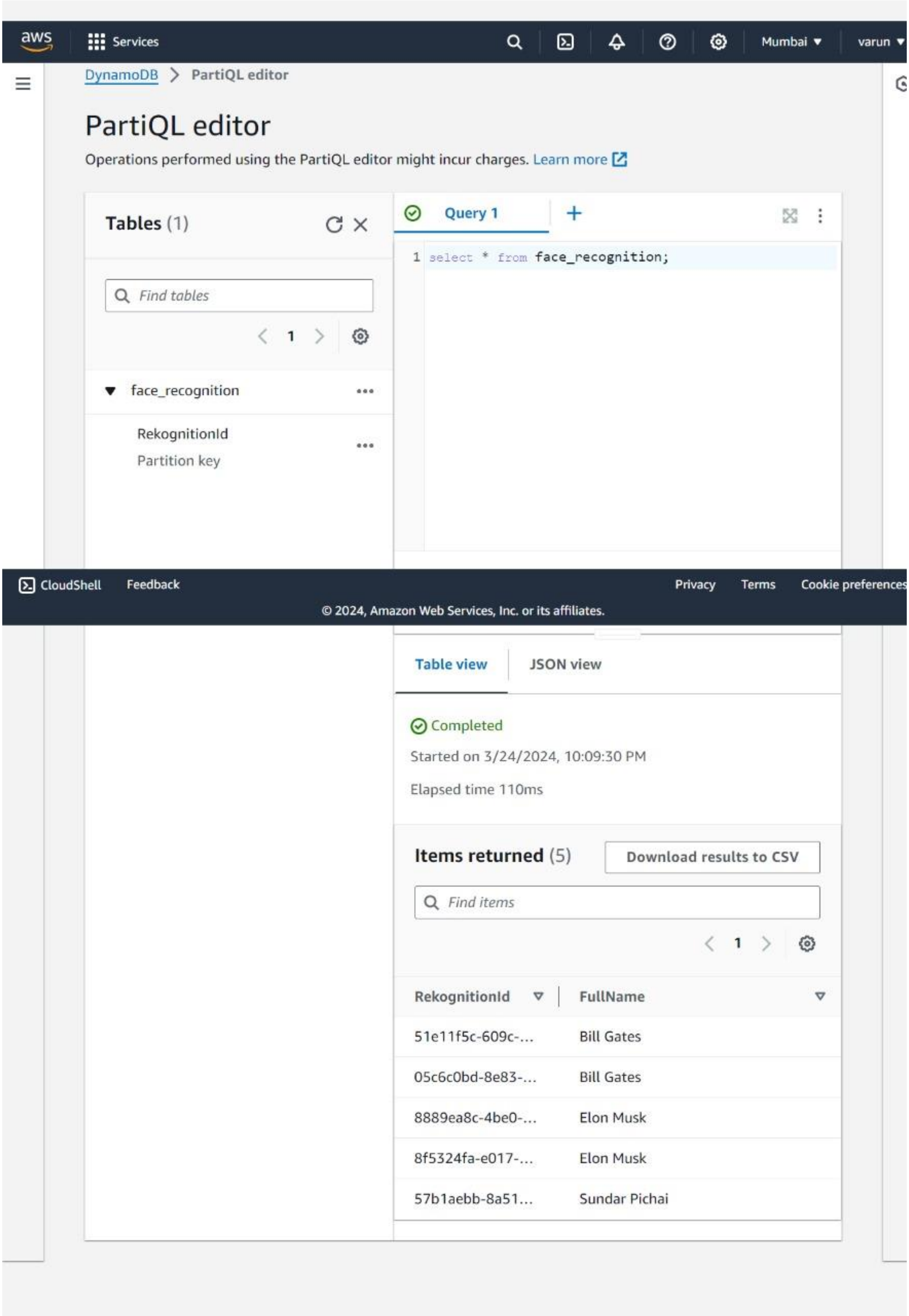
© 2024, Amazon Web Services, Inc. or its affiliates.

5. DynamoDB



□	Name ▲	Status	Partition key	Sort key	Indexes	Deletion protection	Read cap
□	face_recognition	✓ Active	RekognitionId (S)	-	0	⊖ Off	Provision

6. Contents of DynamoDb along with generated Rekognition Id



PartiQL editor

Operations performed using the PartiQL editor might incur charges. Learn more

```
select * from face_recognition;
```

© 2024, Amazon Web Services, Inc. or its affiliates.

✓ Completed

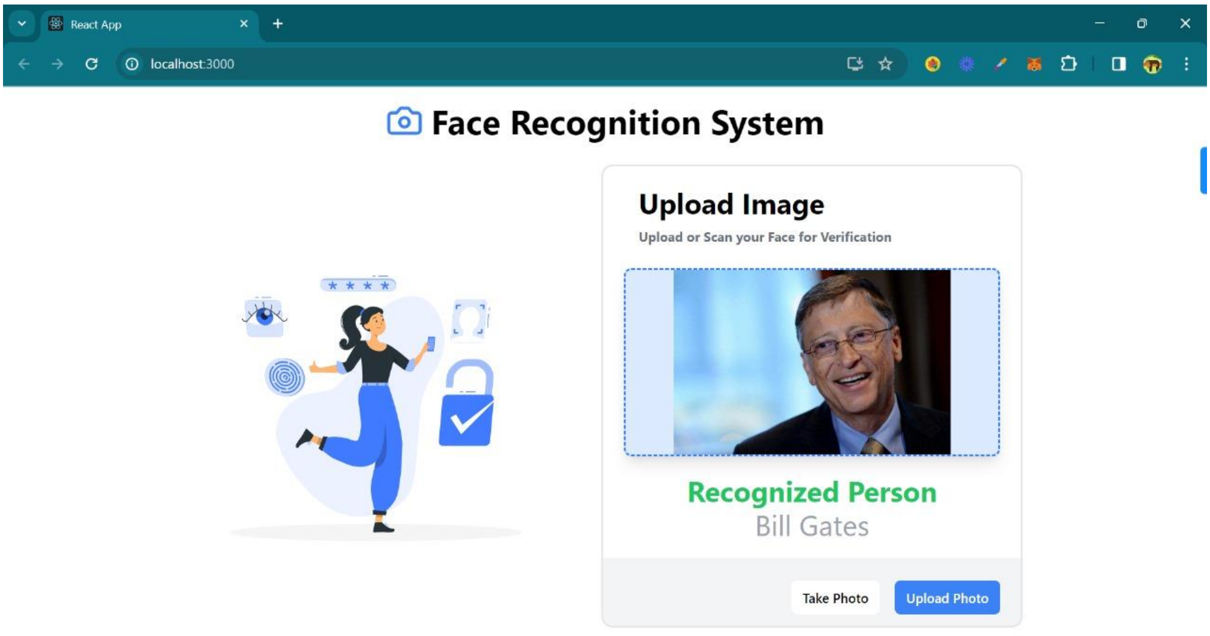
Started on 3/24/2024, 10:09:30 PM

Elapsed time 110ms

Items returned (5)

RekognitionId	FullName
51e11f5c-609c-...	Bill Gates
05c6c0bd-8e83-...	Bill Gates
8889ea8c-4be0-...	Elon Musk
8f5324fa-e017-...	Elon Musk
57b1aebb-8a51...	Sundar Pichai

7. User Interface



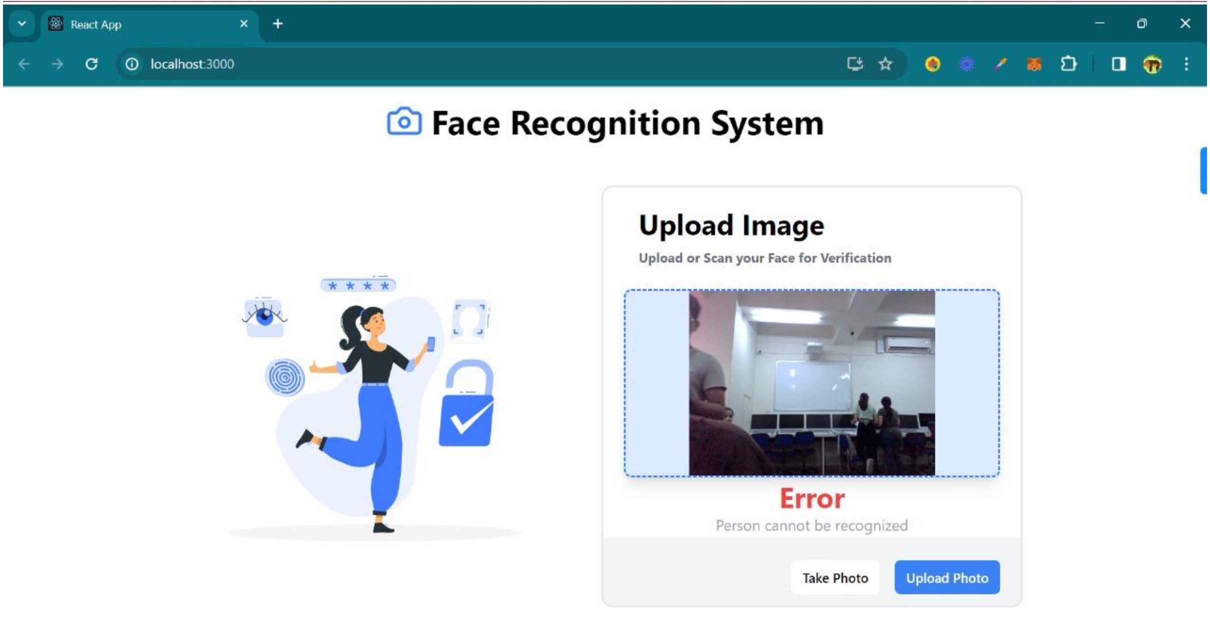
Face Recognition System

Upload Image

Upload or Scan your Face for Verification

Recognized Person

Bill Gates



Face Recognition System

Upload Image

Upload or Scan your Face for Verification

Error

Person cannot be recognized

Conclusion:

In conclusion, our exploration of facial recognition system has revealed a world of boundless possibilities and profound implications. We have witnessed how this transformative technology, fueled by advancements in artificial intelligence and machine learning, is reshaping industries, revolutionizing security measures, and redefining user experiences.