**Facial Recognition System on AWS with Rekognition, Lambda, S3 and DynamoDB**

*A Project Based Learning Report Submitted in partial fulfilment of the requirements for*

*the award of the degree*

*of*

**Bachelor of Technology**

**in The Department of Electronics and Communication Engineering**

**CLOUD & SERVERLESS COMPUTING**

**(22CEC3305A)**

Submitted by

**2210040061: Charishma Kolli**

**2210040072: Sai Dheeraj**

**2210040073: Shivanand**

**2210040099: Narigela Apoorva**

**2210040109: Jahangeer**

*Under the guidance of*

**Ms.Saritha**

**Course Coordinator**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**K L Deemed to be UNIVERSITY**

*Aziznagar, Moinabad , Hyderabad ,*

*Telangana , Pincode: 500075*

2024-2025

**DECLARATION**

We hereby declare that the project entitled “Facial Recognition System on AWS with Rekognition, Lambda, S3 and DynamoDB” which is being submitted as project-based learning of 6th semester in Electronics and Communication Engineering, Aziznagar, Hyderabad in authentic record of genuine work done under the guidance of Ms.Saritha, Assistant Professor, Department of Computer Science and Engineering ,KL University, Aziznagar, Hyderabad.

Date:

Place: Hyderabad

|  |  |
| --- | --- |
| Charishma Kolli | 2210040061 |
| Sai Dheeraj | 2210040072 |
| Shivanand | 2210040073 |
| Narigela Apoorva  Jahangeer | 2210040099  2210040109 |

**K L Deemed to be UNIVERSITY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

*Certificate*

This is Certified that the project entitled **“**ToDesign a facial recognition system on AWS with Rekognition, Lambda, S3 and DynamoDB**”** which is an experimental & Simulation work carried out by Charishma Kolli, Sai Dheeraj, Shivanand, Narigela Apoorva, Jahangeer in partial fulfillment of the course requirements for the award of grades in the subject of **CLOUD AND SERVERLESS COMPUTING**, during the year **2024-2025**. The project has been approved as it satisfies the academic requirements.

**Ms.Saritha Dr. Arpita Gupta**

**Course Coordinator Head of the Department**

**CONTENTS**

1. Introduction

2. AWS Services Used as part of the project

3. Steps involved in solving project problem statement

4. Stepwise Screenshots with brief description

5. Learning Outcomes

6. Conclusion

7. References

**Chapter-1**

1. **INTRODUCTION**

*Mini Project Title*

Facial recognition technology has rapidly evolved, becoming a critical component in security, authentication, and user experience applications. Amazon Web Services (AWS) provides a suite of cloud-based tools that enable developers to design and deploy facial recognition systems efficiently. By leveraging **Amazon Rekognition, AWS Lambda, Amazon S3, and Amazon DynamoDB**, organizations can build scalable, secure, and cost-effective facial recognition solutions.

**Amazon Rekognition** is a powerful AI service that allows developers to analyze images and videos, identify faces, and even detect emotions and attributes. It eliminates the need for complex machine learning models, making it an ideal choice for businesses looking to implement facial recognition with minimal development effort. **AWS Lambda** provides serverless compute power to process images dynamically, reducing infrastructure management. **Amazon S3** serves as the storage layer, holding image data securely, while **Amazon DynamoDB** acts as the NoSQL database for storing metadata, user information, and facial recognition results.

This system can be used for various real-world applications, such as identity verification, access control, customer engagement, and security monitoring. By integrating these AWS services, developers can create a robust, highly available, and serverless facial recognition system without worrying about scalability or infrastructure maintenance. This guide will walk through the architecture, implementation, and best practices for designing such a system on AWS.

**Chapter - 2**

**2. AWS SERVICES USED AS PART OF THE PROJECT**

1. **AWS Rekognition:**

**Amazon Rekognition** is at the core of the system, offering AI-powered image and video analysis capabilities. It enables facial detection, recognition, and comparison, allowing the system to match detected faces against a stored dataset with high accuracy. Rekognition also provides additional features such as emotion analysis and facial attribute detection, making it a powerful tool for various applications.

**2.AWS Lambda**

**AWS Lambda** plays a crucial role in the system's architecture by providing serverless computing power to process images dynamically. Whenever a new image is uploaded to Amazon S3, a Lambda function is triggered to invoke **Rekognition,** analyze the image, and process the results. This eliminates the need for managing dedicated servers, reducing operational costs while ensuring high availability.

**3.AWS S3**

**Amazon S3** acts as the primary storage solution for images and facial datasets. It offers secure, scalable, and cost-effective storage, allowing users to upload and retrieve images efficiently. S3 also integrates seamlessly with other AWS services, enabling event-driven workflows that trigger further processing when new images are added.

**4.Amazon DynamoDB**

**Amazon DynamoDB** is used as a NoSQL database. It stores metadata, user profiles, and recognition results in a highly available and low-latency manner. DynamoDB’s scalability ensures that the system can handle large amounts of data efficiently without performance bottlenecks.

**Chapter - 3 3. STEPS INVOLVED IN SOLVING PROJECT PROBLEM STATEMENT**

**1. Set Up an Amazon S3 Bucket for Image Storage**

* Create an Amazon S3 bucket to store user images and reference images for facial recognition.
* Enable proper permissions and access control to ensure security and compliance.

**2. Enable Amazon Rekognition for Face Analysis**

* Configure Amazon Rekognition to detect and recognize faces in images.
* Set up a Rekognition collection to store and manage facial feature vectors for matching.

**3. Implement AWS Lambda for Serverless Processing**

* Develop a Lambda function that triggers when a new image is uploaded to S3.
* The function should call Amazon Rekognition to analyze the image and compare faces with stored records.
* Process the response and extract relevant data such as match confidence, face ID, and attributes.

**4. Store and Manage Data Using Amazon DynamoDB**

* Create a DynamoDB table to store user metadata, face IDs, and recognition results.
* Ensure efficient indexing and query patterns to support fast lookups and retrievals.

**5. Implement Logging and Monitoring**

* Enable AWS CloudWatch for monitoring logs, metrics, and system performance.
* Set up alerts to detect failures or unauthorized access.

**6. Test and Optimize the System**

* Run multiple test cases with different images to ensure accuracy and performance.
* Optimize Lambda execution time and DynamoDB queries to improve efficiency.

**Chapter – 4 4. STEPWISE SCREENSHOTS WITH BRIEF DESCRIPTION**

**Chapter – 5**

**5. LEARNING OUTCOMES**

**Chapter – 6**

**6. CONCLUSION**

**Chapter - 7**

**7. REFERENCES**

**1.Amazon Documentation:**

Build Your Own Face Recognition Service Using Amazon Rekognition

<https://aws.amazon.com/blogs/machine-learning/build-your-own-face-recognition-service-using-amazon-rekognition/>

**2. Build a Facial Recognition App on AWS from Scratch**

<https://youtu.be/FsXyfxLDDIs?si=apRrK3z9oI5G49Lv>

**3. Facial Recognition Service Using Amazon Rekognition AI Service**

<https://youtu.be/k2Q_dQRdbvk?si=kO04uwLVBOd3Fzhb>