CSCI 5410
Serverless Data Processing
Fall(2024)

Lab Activity Report

User Management Web App

BannerId: B00984592

Contents

1. Introduction 3

2. Application Overview 3

3. Thought Process and Design 3

4. Application Functionality 4

5. Screenshots 5

6. Deployed URL 9

References: 10

1. Introduction

The aim of this lab activity is to develop a serverless user management web application using AWS services such as Lambda, S3, and DynamoDB. The web app allows users to perform CRUD (Create, Read, Update, Delete) operations on user data, including uploading profile images, with all data stored in DynamoDB[2] and images in S3[3]. The application has been deployed using Netlify, and it meets the requirements set for the activity.

- **Live Application URL**: https://6706b2ce077967000809e68c--playful-elf-43f236.netlify.app/

2. Application Overview

This user management application allows users to:

- Create a new user with details such as name, email, and profile picture.
- Edit user information (name and email).
- Delete users.
- View a list of all users with their details and profile pictures.

The user data is stored in DynamoDB, and profile images are uploaded to Amazon S3. Four AWS Lambda functions are used for handling the CRUD operations.

3. Thought Process and Design

The application was designed with a focus on simplicity and scalability. Using AWS services like Lambda, DynamoDB, and S3 provided a serverless architecture that reduces maintenance overhead and ensures high availability.

- **Lambda Functions**:
    - **CreateUser**: Accepts user details (name, email, and image) and stores them in DynamoDB. The image is uploaded to S3.
    - **GetUser**: Retrieves user details from DynamoDB.

  - **UpdateUser**: Updates user details such as name and email in DynamoDB.
  - **DeleteUser**: Deletes a user from DynamoDB and the corresponding image from S3.

- **Frontend**: The frontend is built using React and Bootstrap to handle the user interface and provide a clean and responsive design.

Challenges Faced:

- Managing file uploads and converting them into a format (Base64) suitable for storing in S3 was a bit challenging.
- Making sure that proper policies are attached to the bucket to get the access.
- Configuring the lambdas to access 2 services(Dynamo DB and S3) of the amazon.

4. Application Functionality

The application fulfills the following requirements from the lab activity:

1. **Four Lambda Functions**: The app has four Lambda functions to handle each CRUD operation[1].
2. **S3 for Storage**: S3 is used for storing user profile images, ensuring scalable and reliable image storage[3].
3. **DynamoDB Interactions**: DynamoDB is used to store and manage user information, interacting with two use cases (Create/Update and Get/Delete)[2].

5. Screenshots

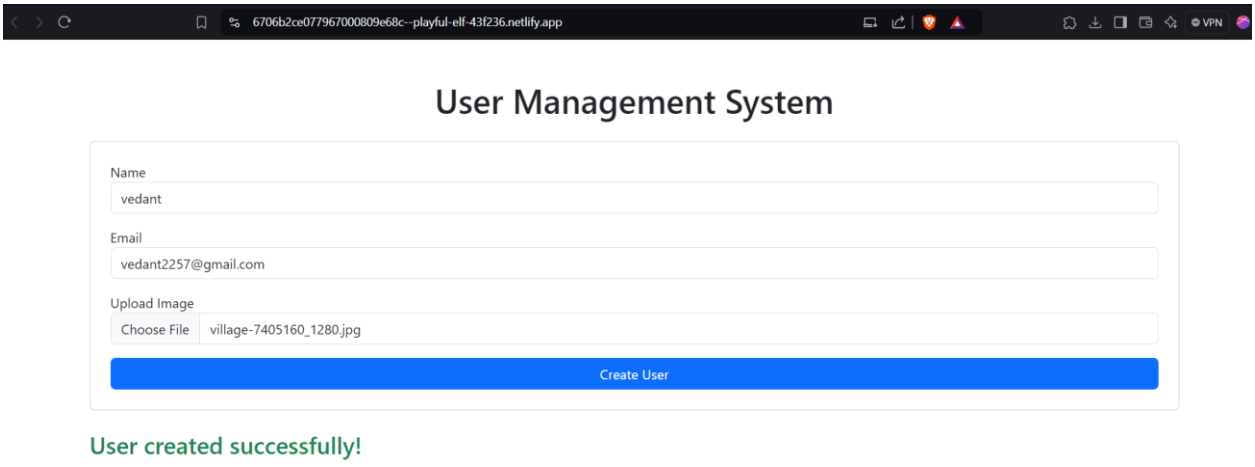

Figure 1: User details forms with image

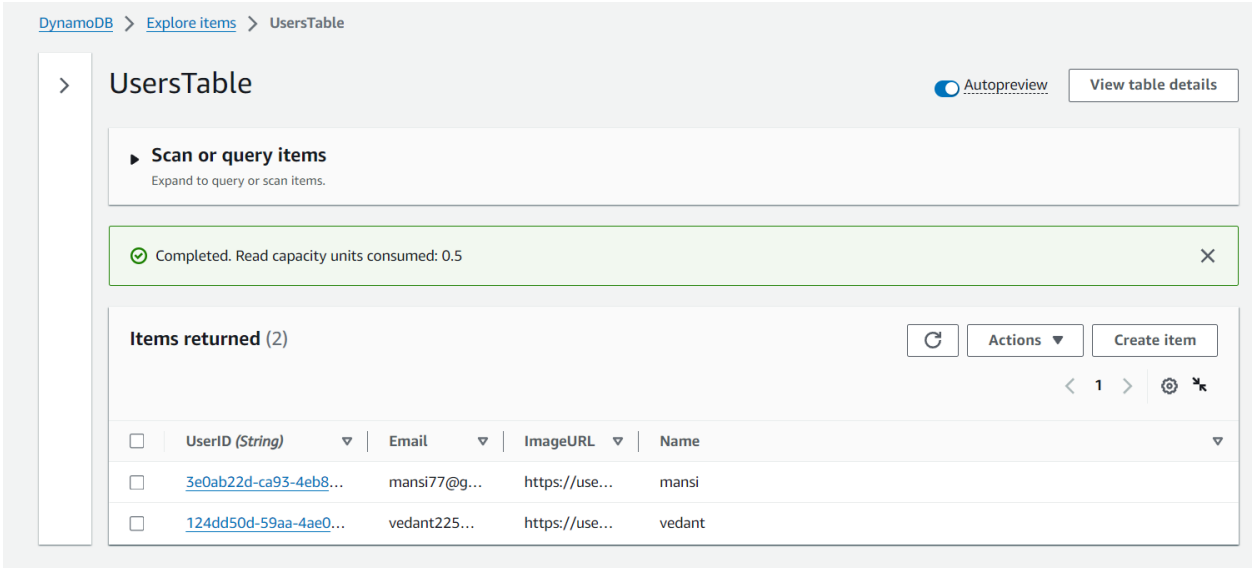

Figure 2: Inserted data in the dynamo DB tables named UserTable

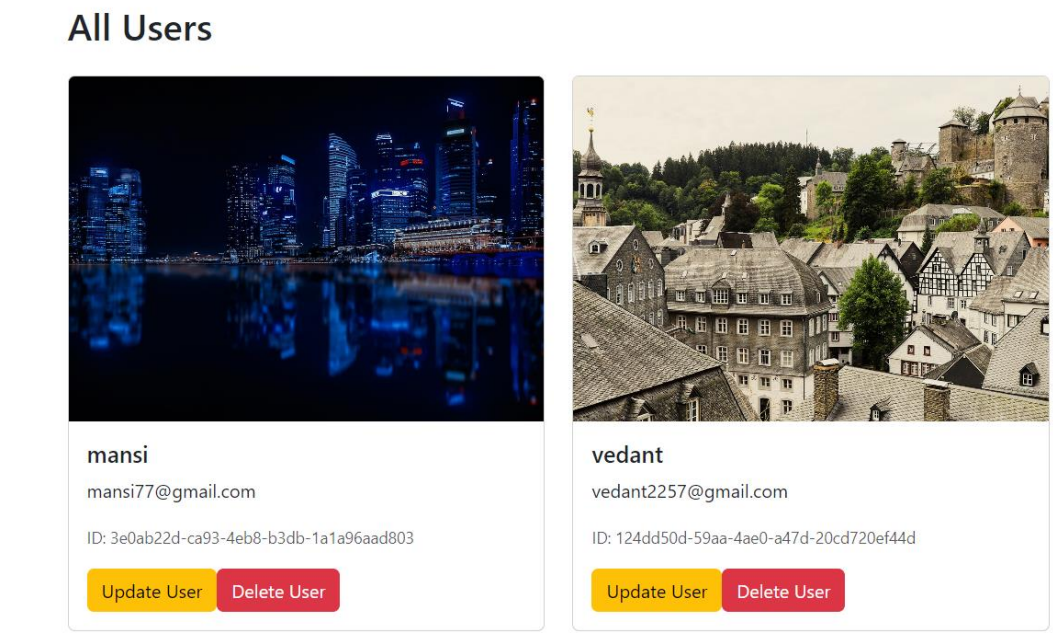

Figure 4: Users card showing the users details

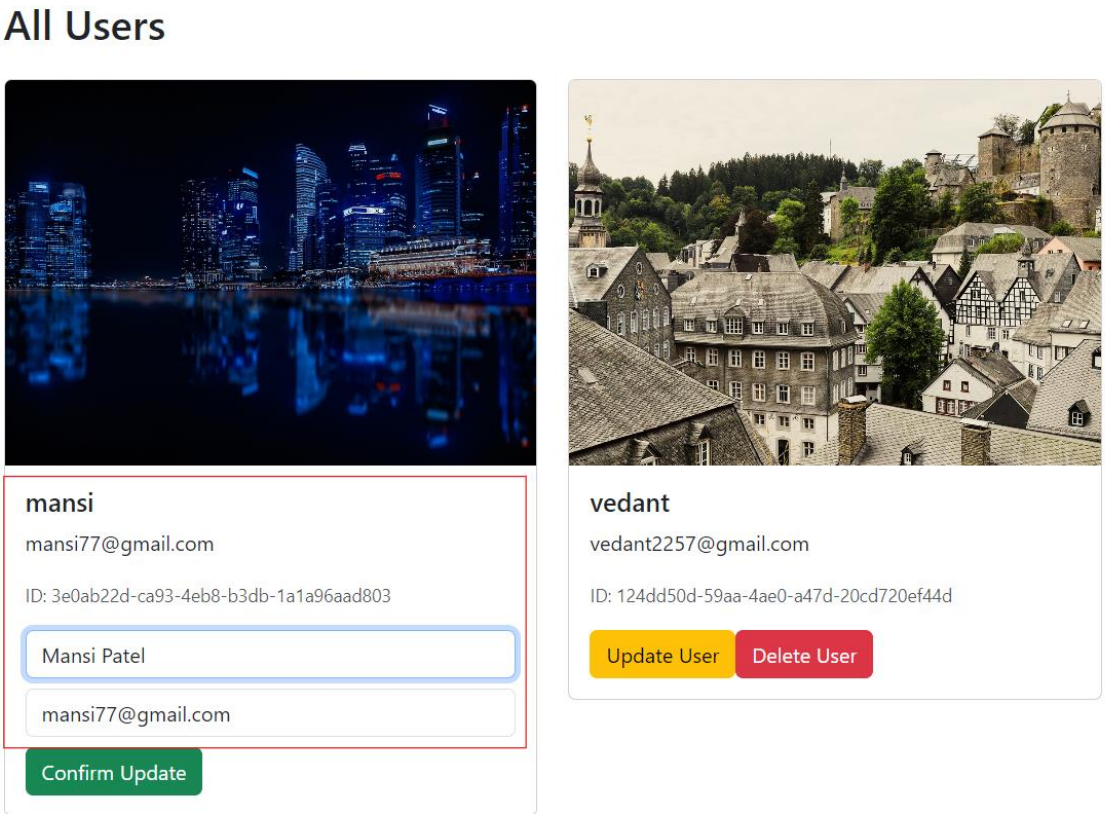

Figure 3: Updating the user

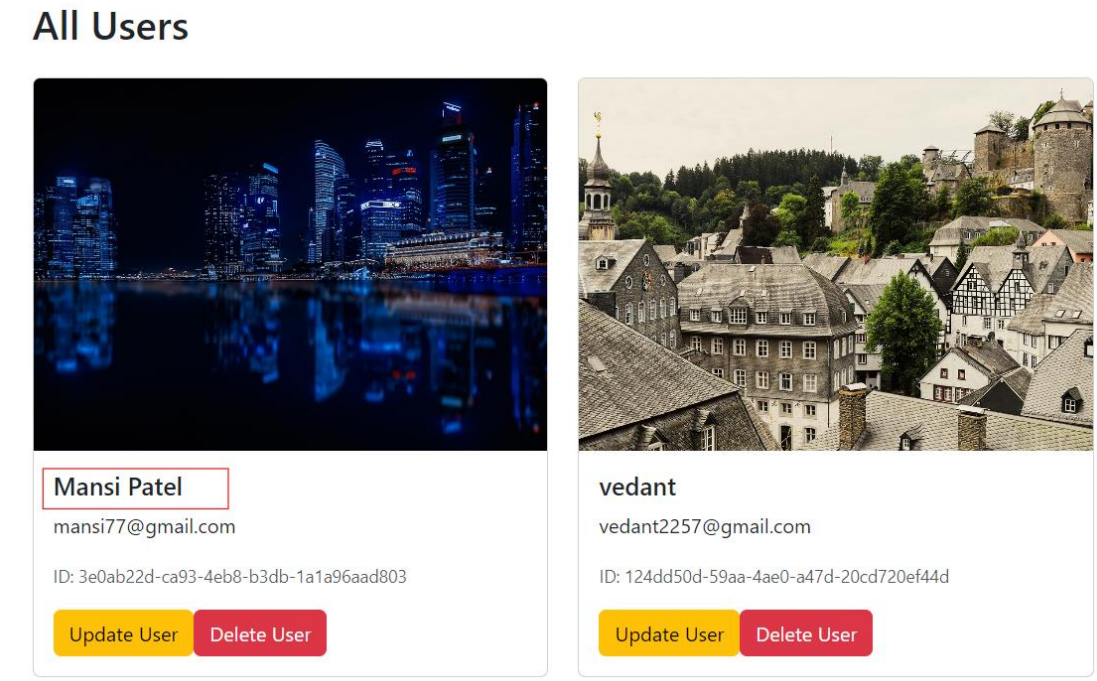

Figure 6: User Updated

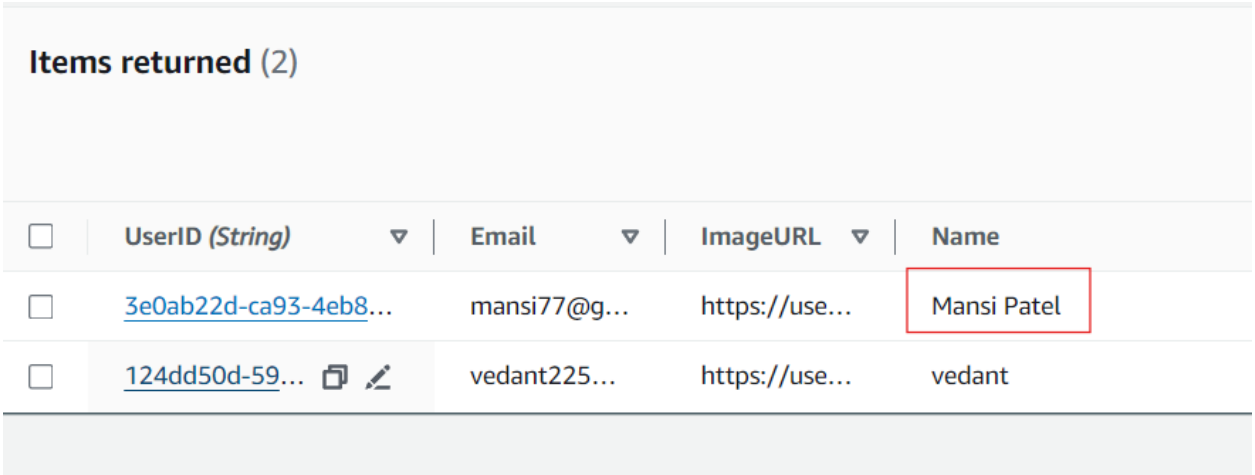

Figure 5:Details Updated in the table

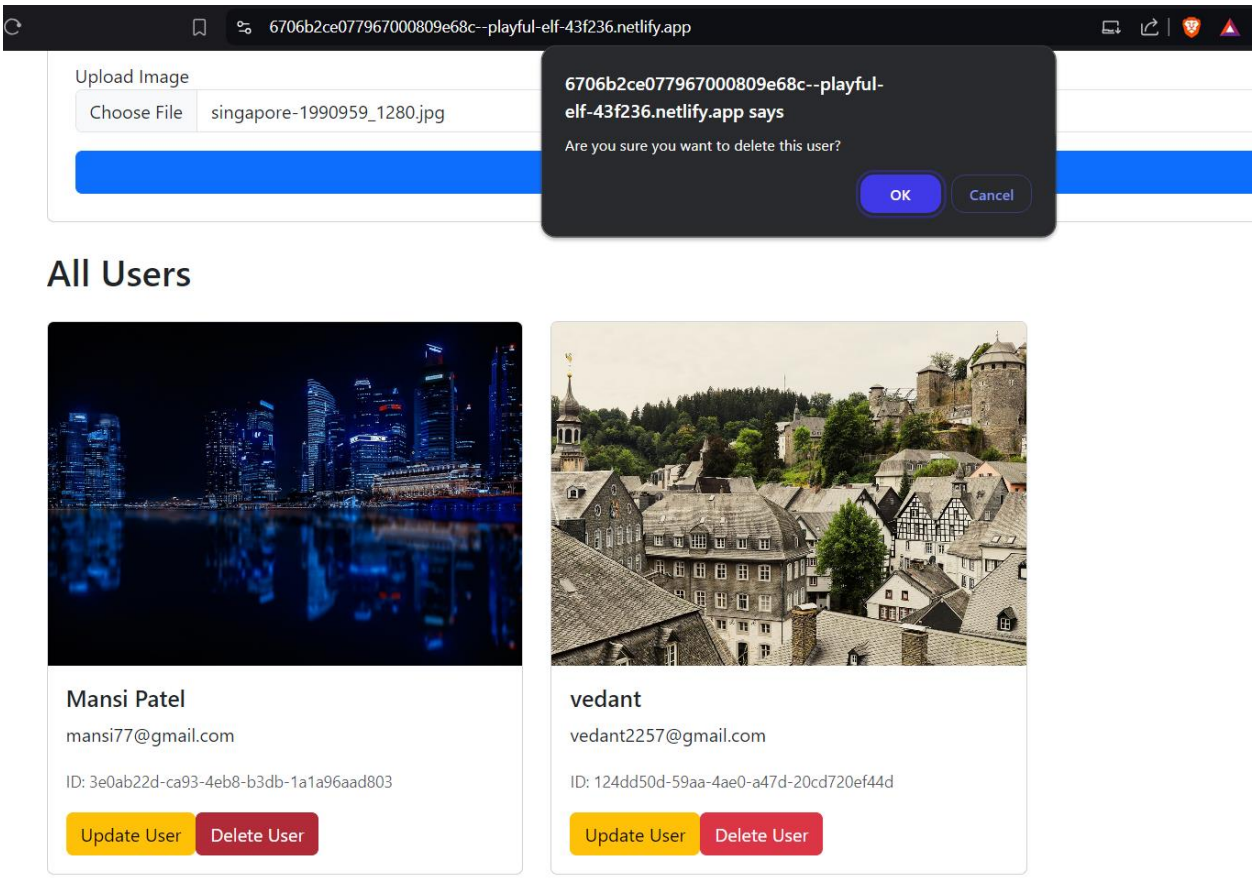

Figure 7: Deleting the user

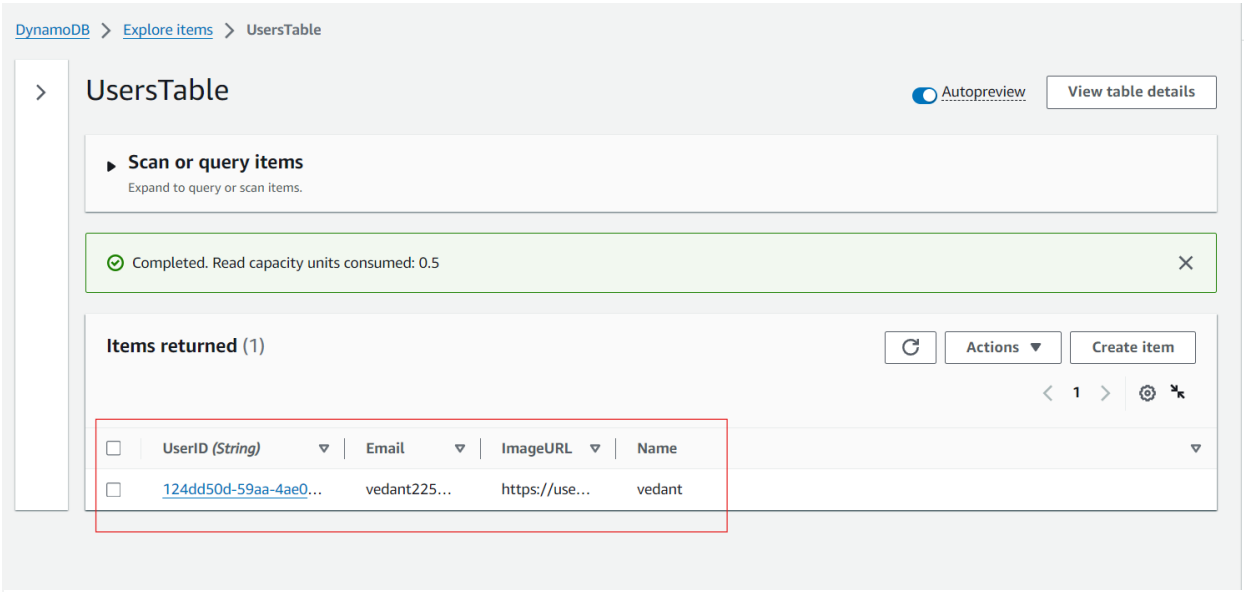

Figure 8: Details removed from the table

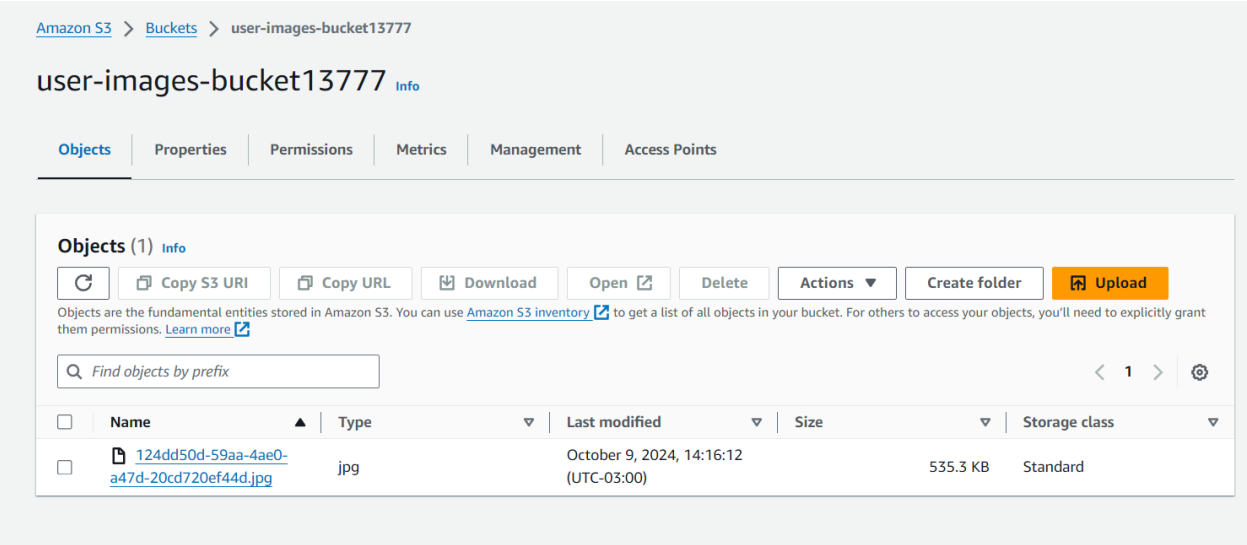

Figure 9: S3 buckets containing all of the images

6. Deployed URL

The live application is deployed at:
https://6706b2ce077967000809e68c--playful-elf-43f236.netlify.app/

References:

[1] Amazon Web Services, Inc., "AWS Lambda Developer Guide," [Online]. Available: https://docs.aws.amazon.com/lambda/latest/dg/welcome.html. [Accessed: 09-Oct-2024].

[2] Amazon Web Services, Inc., "Amazon DynamoDB Developer Guide," [Online]. Available: https://docs.aws.amazon.com/dynamodb/. [Accessed: 09-Oct-2024].

[3] Amazon Web Services, Inc., "Amazon S3 User Guide," [Online]. Available: https://docs.aws.amazon.com/s3/. [Accessed: 09-Oct-2024].

[4] Amazon Web Services, Inc., "AWS Lambda," [Online]. Available: https://docs.aws.amazon.com/lambda/. [Accessed: 09-Oct-2024].