KARNAVATI UNIVERSITY

UNITEDWORLD SCHOOL OF COMPUTATIONAL INTELLIGENCE (USCI)

Summative Assessment (SA)

Submitted by
Tanya Chhabahdiya
(Enrl. No.: 20210701007)

Course Code and Title: 21BSCS35C03 – AWS Cloud

B.Sc. (Hons.) Computer Science / Data Science / AIML
V Semester – July – Nov 2023

USCI

Nov/Dec 2023

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The Video Indexing project is a sophisticated web application leveraging HTML, CSS, and JavaScript, hosted on Amazon Web Services. It enables users to securely upload and analyze videos using AWS services like S3 for storage and Rekognition for comprehensive content analysis, including label detection, text recognition, facial analysis, and celebrity identification.

1.1 PROJECT DESCRIPTION

Video indexing refers to the process of analyzing and cataloging video content to extract relevant information and make it easily searchable and retrievable. It involves the extraction of metadata, which includes various features such as visual, auditory, and textual elements, from the video data. The goal is to create an index or database of the video content that enables efficient and effective content retrieval based on specific criteria.

The Video Indexing project harnesses the capabilities of Amazon Rekognition and an array of AWS services to create a sophisticated web application for video content analysis. The foundation of the project lies in its HTML, CSS, and JavaScript-based web interface, providing users with a seamless experience to upload videos. These videos are then securely stored in an Amazon S3 bucket, ensuring durability, high availability, and easy scalability of storage.

To fortify the project against unauthorized access and manipulation, IAM roles and permissions are meticulously configured. This ensures that only authenticated and authorized users can interact with the application, maintaining the integrity and confidentiality of the stored video content.

Upon uploading a video, Amazon Rekognition takes center stage, leveraging its robust capabilities to analyze the content comprehensively. Labels are automatically detected, offering insights into the video's content. Text within the videos is recognized, enabling applications in transcription and content categorization. The facial recognition feature not only identifies faces but also gauges the associated emotions, providing a nuanced understanding of the video's sentiment.

One of the standout features is the ability to identify faces of celebrities within the videos. This is made possible through seamless integration with AWS SNS, which notifies users of the analysis results in real-time. In the background, AWS SQS efficiently manages asynchronous processing tasks, enhancing the overall responsiveness and scalability of the system.

The synergy of AWS services ensures a robust and scalable architecture. The project is not only user-friendly but also versatile, finding applications in diverse domains such as content moderation, sentiment analysis, and celebrity recognition. With security at its core, the Video Indexing project sets a benchmark in leveraging cutting-edge technologies to deliver a comprehensive and efficient solution for video content analysis on the cloud.

1.2 ARCHITECTURE DIAGRAM

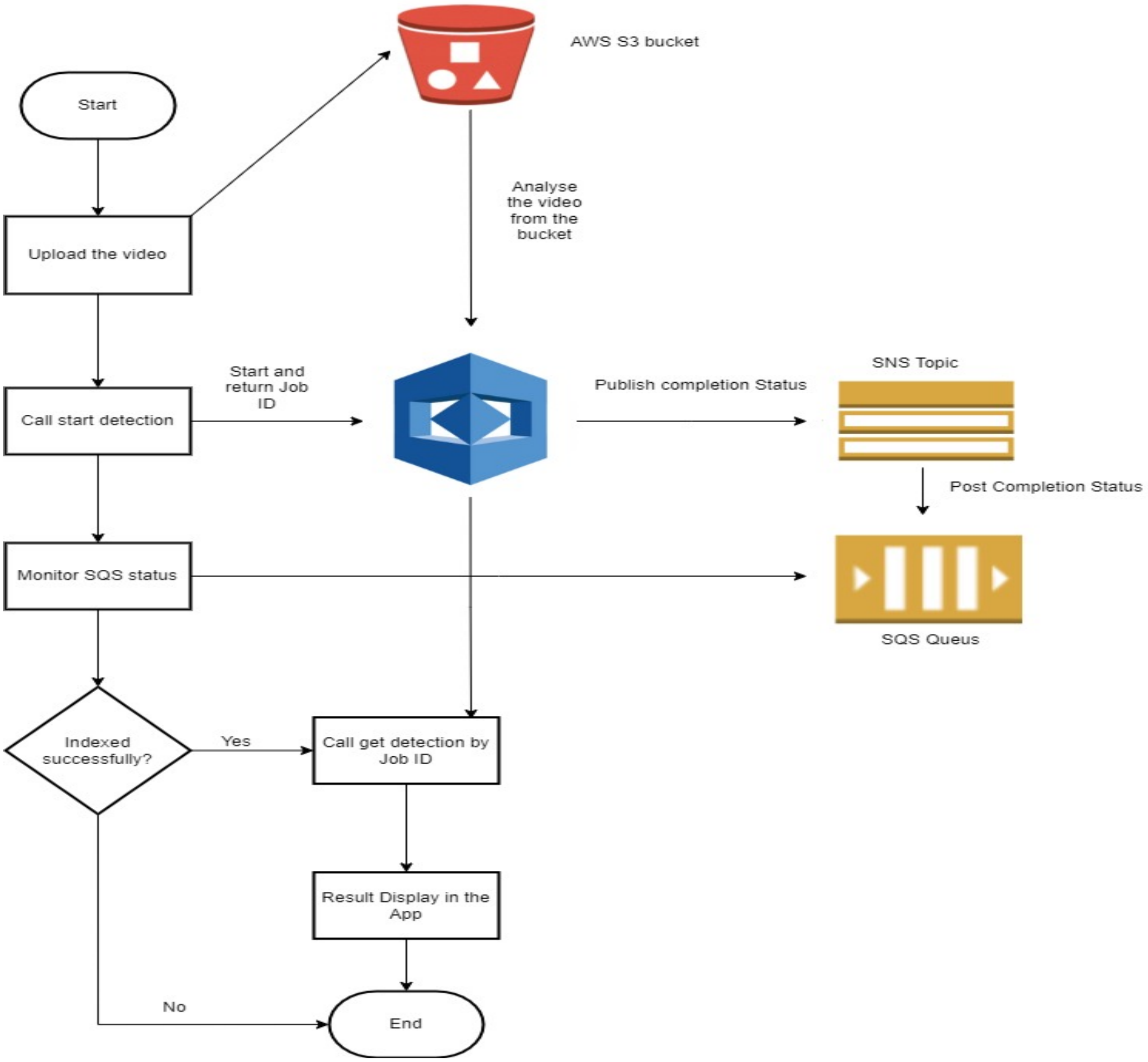


Fig 1.1 Hierarchical diagram of Application

The architecture depicts a video indexing system built on Amazon Web Services (AWS). The system uses a web application as the front-end, where users can upload videos for analysis. Once uploaded, the video is stored in an S3 bucket, which is a secure and scalable storage solution.

An SNS topic is then notified about the new video, and it in turn sends a message to an SQS queue. This queue acts as a buffer, holding the video analysis jobs until a worker process is available to pick them up.

When a worker process is ready, it takes a message from the SQS queue and starts analyzing the video using Amazon Rekognition. Rekognition is a service that can detect objects, scenes, text, faces, and even celebrities in videos. The analysis results are then displayed to the user. This could include things like a list of objects detected in the video, a transcript of the spoken text, or a list of the faces that were recognized.

Overall, this architecture provides a secure and scalable way to analyze videos using Amazon Rekognition. The SQS queue helps to ensure that videos are processed even if there are spikes in traffic, and the data store allows you to store the analysis results for later use.

Chapter 2

RESOURCE SPECIFICATIONS

To build the Video Indexing project, you'll need a combination of tools, platforms, and specific hardware and software specifications.

2.1 TOOLS AND PLATFORMS

1. **Amazon Web Services (AWS):**
    - **Amazon S3:** For secure and scalable storage of uploaded videos.
    - **Amazon Rekognition:** Leveraged for video content analysis, including label detection, text recognition, facial analysis, and celebrity identification.
    - **IAM (Identity and Access Management):** Used to configure security permissions and roles, ensuring secure access to AWS resources.
    - **AWS SNS (Simple Notification Service):** For real-time notifications about the results of video analysis.
    - **AWS SQS (Simple Queue Service):** Manages asynchronous processing, enhancing system responsiveness.
2. **Web Technologies:**
    - **HTML, CSS, JavaScript:** The foundation for the web application's user interface and interactivity.

2.2 HARDWARE AND SOFTWARE SPECIFICATIONS

Hardware Specifications:

1. **Client-side (User Devices):**
    - Standard modern web browsers (Chrome, Firefox, Safari, etc.): Ensure compatibility with the web application.
2. **Server-side:**

- AWS Cloud Infrastructure: Utilize AWS EC2 instances or serverless computing services for hosting the web application and handling backend processes.

Software Specifications:

1. **Development Environment:**
   - Text Editor or Integrated Development Environment (IDE): Use tools like Visual Studio Code, Sublime Text, or others for coding.
   - Git: Version control for collaborative development.
2. **Backend Development:**
   - Node.js, Express.js (optional): For server-side logic and handling HTTP requests.
   - AWS SDK for JavaScript: Facilitates interaction with AWS services programmatically.
3. **Database (if applicable):**
   - DynamoDB, Amazon RDS, or other AWS Database Services: If you need to store additional metadata or user information.

Chapter 3

DEVELOPMENT PROCESS

The development process for the Video Indexing project begins with setting secure IAM roles and permissions in AWS, followed by configuring S3 buckets and SNS for secure video storage and real-time notifications. A crucial step involves understanding the AWS SDK documentation to facilitate programmatic interactions with AWS services. The web application is then constructed using HTML, CSS, and JavaScript for user-friendly video uploads. Integration with the AWS SDK ensures secure communication, and HTTPS and user authentication enhance overall security. Specific video analysis functionalities, such as label detection, face recognition, and celebrity identification, are implemented using Amazon Rekognition, with results seamlessly displayed in the application interface. The iterative development process emphasizes testing and documentation to ensure the robustness and reliability of the Video Indexing project.

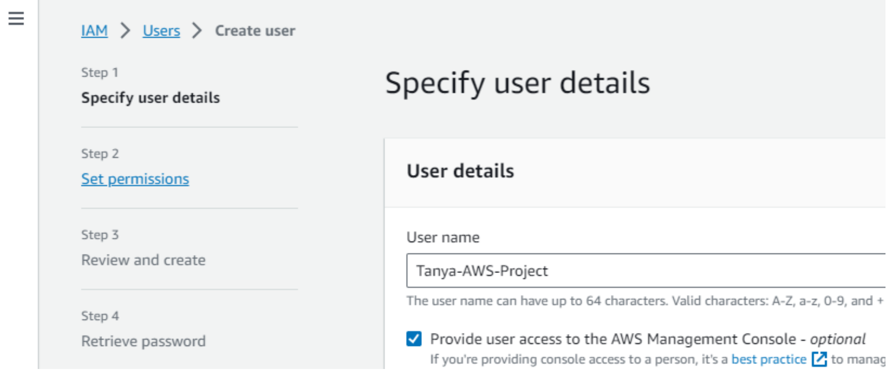
3.1 IMPLEMENTATION

1. Set Security Permissions

   For the security permissions, IAM (AWS Identity and Access Management) is used. In IAM a user and role for the user is created through which they can access the service through the web application publically.

   The steps are:

   1) Create a user



Specify user details

User details

User name

Tanya-AWS-Project

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and +

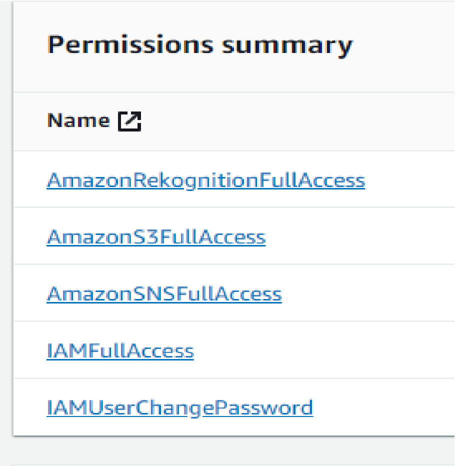
☑ Provide user access to the AWS Management Console - *optional*

If you're providing console access to a person, it's a best practice to manag

Fig 3.1 Create IAM User

2) Set permissions

3) Allow several permission policies

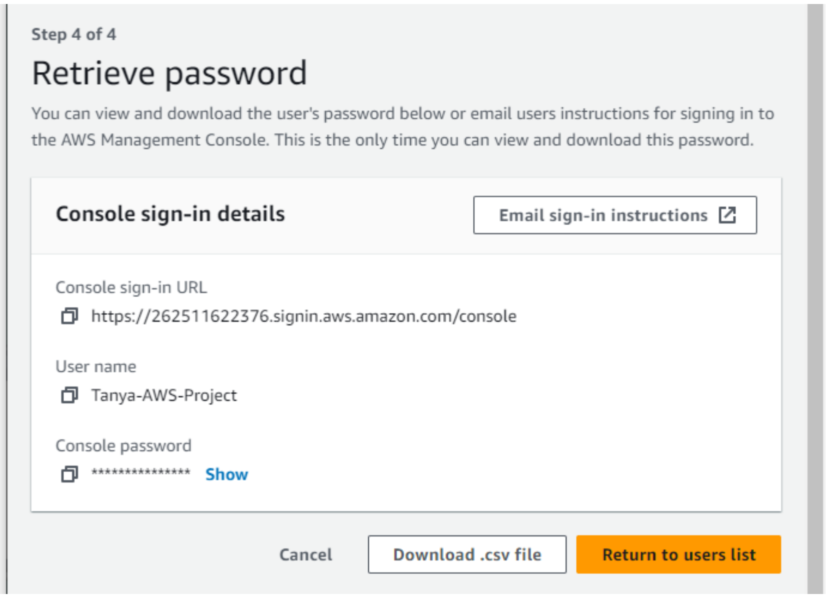


Permissions summary

Name
AmazonRekognitionFullAccess
AmazonS3FullAccess
AmazonSNSFullAccess
IAMFullAccess
IAMUserChangePassword

Fig 3.2 Policies

4) Download the credentials for future use.



Step 4 of 4

Retrieve password

You can view and download the user's password below or email users instructions for signing in to the AWS Management Console. This is the only time you can view and download this password.

Console sign-in details

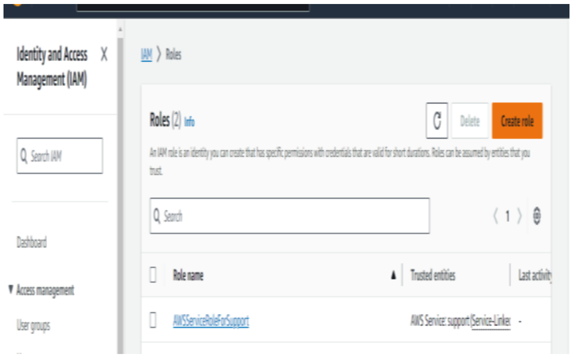
Console sign-in URL
https://262511622376.signin.aws.amazon.com/console

User name
Tanya-AWS-Project

Console password
*************** Show

Fig 3.3 Download Credentials

5) Create a role



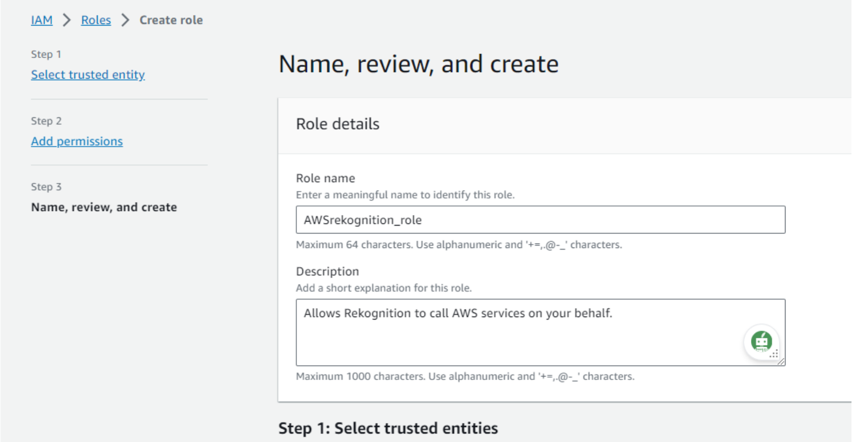
Roles (2) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name	Trusted entities	Last activity
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-

Fig 3.4 Create Role

6) Fill necessary details.



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
AWSrekognition_role
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows Rekognition to call AWS services on your behalf.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Fig 3.5 Fill Details

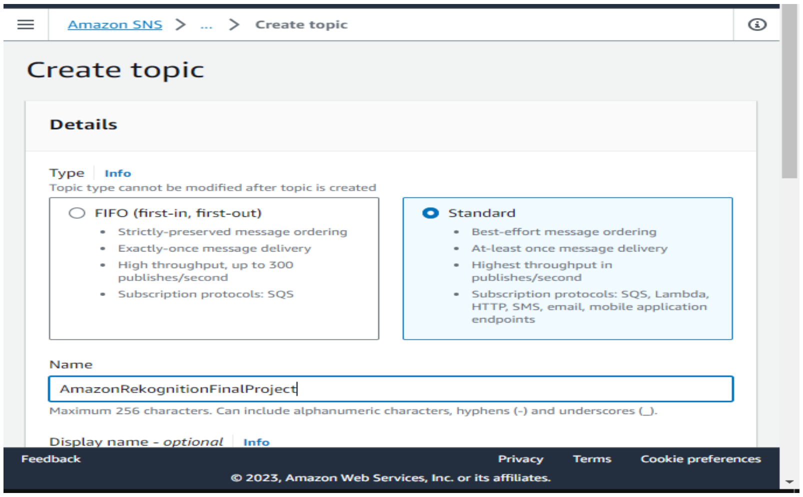
2. Configure AWS SNS and AWS S3

   In order to store the video and have a notification for the process we need to use SNS and S3. Before using the services some configuration has to be done in order to use them late in the web application development phase.

   Steps are:

   - SNS
     - Create topic
     - Use standard option

- ○ Select the name of service for which the SNS is used.



Amazon SNS > ... > Create topic

Create topic

Details

Type Info

Topic type cannot be modified after topic is created

○ FIFO (first-in, first-out)
- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

● Standard
- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

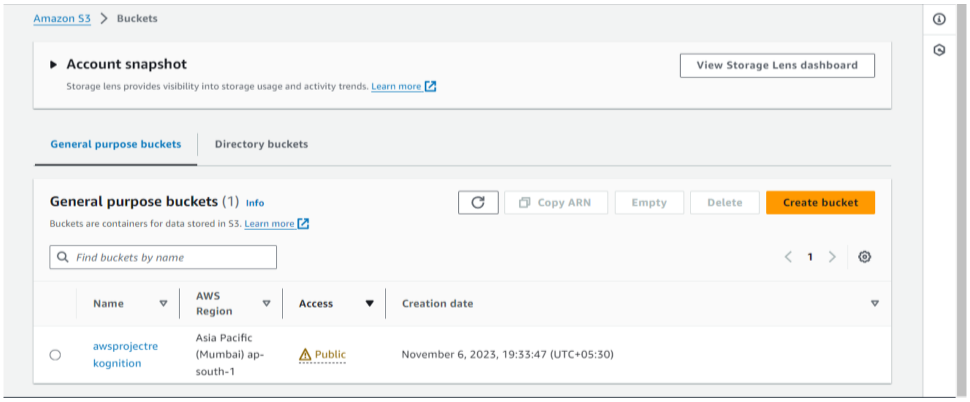
AmazonRekognitionFinalProject

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional* Info

Fig 3.6 Configure SNS

- S3
  - ○ Create the bucket
  - ○ Allow public access for objects
  - ○ Update the bucket and CORS policy



Amazon S3 > Buckets

▸ Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Find buckets by name

	Name	AWS Region	Access	Creation date
○	awsprojectrekognition	Asia Pacific (Mumbai) ap-south-1	⚠ Public	November 6, 2023, 19:33:47 (UTC+05:30)

Fig 3.7 Configure S3

3. Understanding the AWS SDK Documentation

   In order to use command like get and start we use SDK Documentation for Rekognition API. The code for JavaScript is used for building the web application

   The steps are:

   - Familiarize yourself with the AWS SDK documentation for the programming language of choice (e.g., JavaScript).

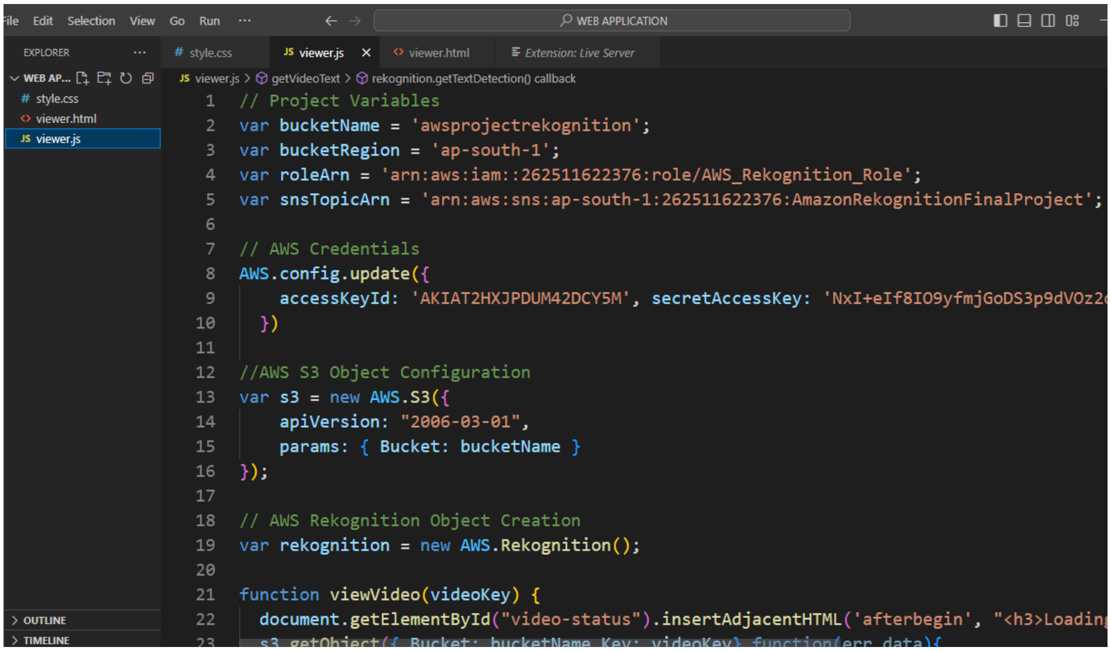
- Learn how to authenticate and interact with AWS services programmatically using the SDK.
- Understand key concepts like AWS credentials, service endpoints, and error handling.

4. Configuring a Web application for Video Indexing

   To deploy the API a web application is created using HTML, CSS and JavaScript.

   Steps are:

   - Create the HTML page containing buttons for the functions like upload video, detect labels etc.
   - Create a stylesheet for design
   - Using the reference from the SDK Documentation For Rekognition API build the JavaScript file.
   - Add credential that were generated while setting IAM permission.
   - Add bucket name, region name, SNS ARN and Role ARN in the JavaScript file.



```
// Project Variables
var bucketName = 'awsprojectrekognition';
var bucketRegion = 'ap-south-1';
var roleArn = 'arn:aws:iam::262511622376:role/AWS_Rekognition_Role';
var snsTopicArn = 'arn:aws:sns:ap-south-1:262511622376:AmazonRekognitionFinalProject';

// AWS Credentials
AWS.config.update({
    accessKeyId: 'AKIAT2HXJPDUM42DCY5M', secretAccessKey: 'NxI+eIf8IO9yfmjGoDS3p9dVOz2
  })

//AWS S3 Object Configuration
var s3 = new AWS.S3({
    apiVersion: "2006-03-01",
    params: { Bucket: bucketName }
});

// AWS Rekognition Object Creation
var rekognition = new AWS.Rekognition();

function viewVideo(videoKey) {
  document.getElementById("video-status").insertAdjacentHTML('afterbegin', "<h3>Loading
  s3.getObject({ Bucket: bucketName,Key: videoKey},function(err,data){
```

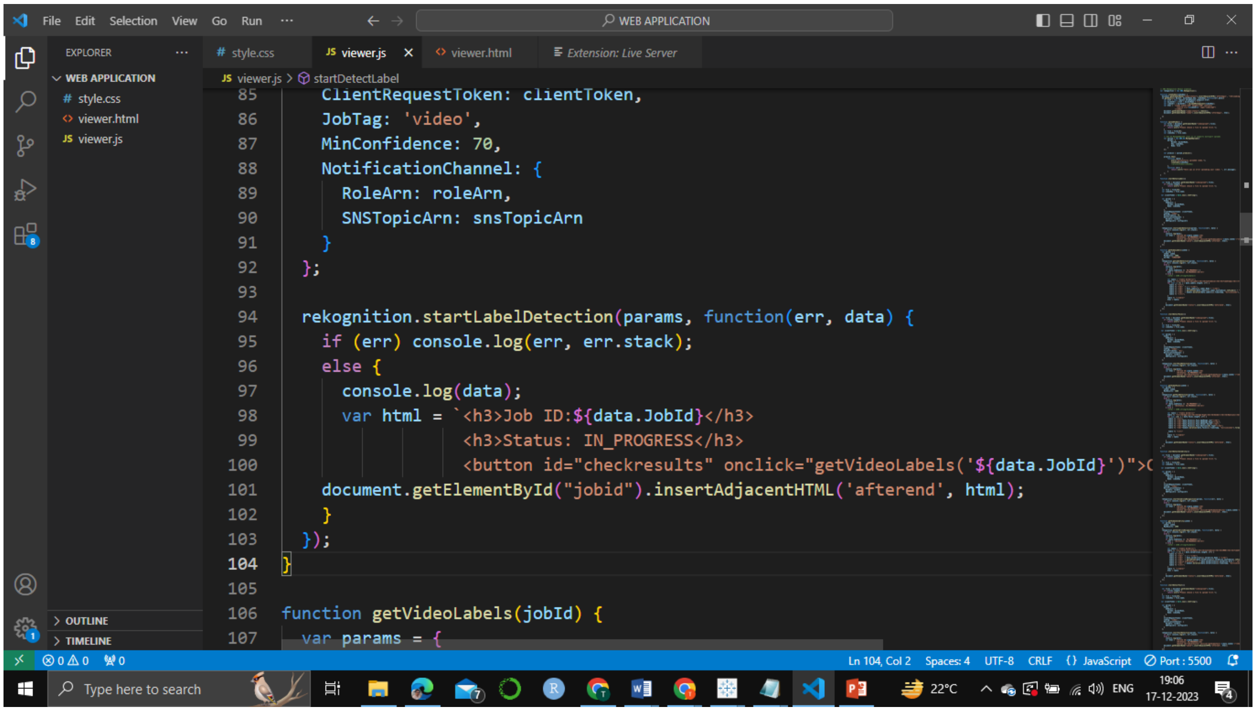
Fig 3.8 Create Web Application

5. Detect Labels, text and faces in a Video

Code for getting labels text and faces will be used from the SDK Documentation in the JavaScript for enabling these services.

The steps are:

- Utilize Amazon Rekognition to detect labels within uploaded videos.
- Integrate the AWS SDK to send video analysis requests to Rekognition.
- Parse and display the label detection results in the web application interface.
- Extend Rekognition usage to include face detection within videos.
- Leverage the AWS SDK to request face detection analysis.
- Present face detection results on the web interface.
- Implement celebrity recognition using Amazon Rekognition.
- Use the AWS SDK to trigger celebrity detection analysis for videos.
- Display recognized celebrities in the application interface.



```
    ClientRequestToken: clientToken,
    JobTag: 'video',
    MinConfidence: 70,
    NotificationChannel: {
      RoleArn: roleArn,
      SNSTopicArn: snsTopicArn
    }
  };

  rekognition.startLabelDetection(params, function(err, data) {
    if (err) console.log(err, err.stack);
    else {
      console.log(data);
      var html = `<h3>Job ID:${data.JobId}</h3>
                  <h3>Status: IN_PROGRESS</h3>
                  <button id="checkresults" onclick="getVideoLabels('${data.JobId}')">C
    document.getElementById("jobid").insertAdjacentHTML('afterend', html);
    }
  });
}

function getVideoLabels(jobId) {
  var params = {
```

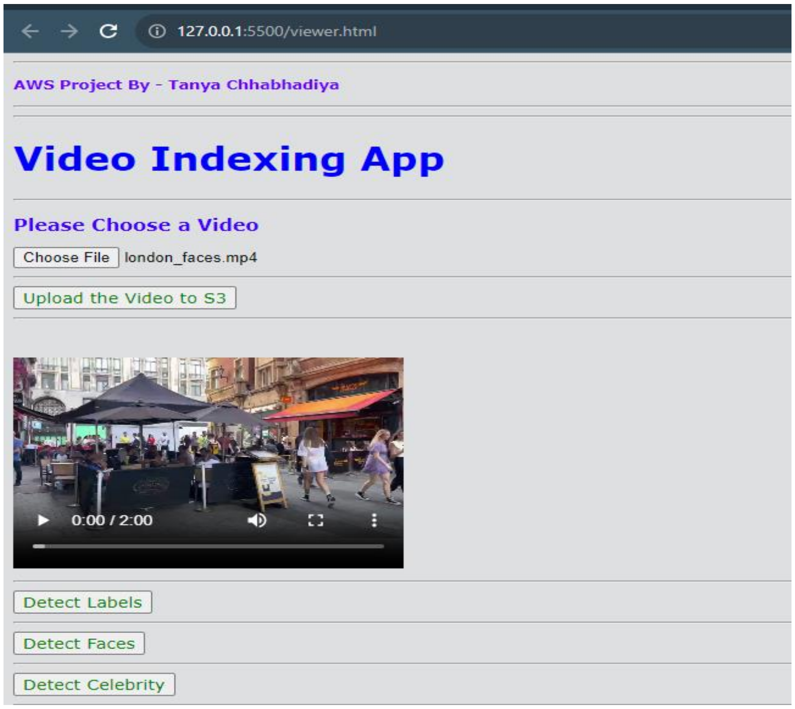
Fig 3.9 Code Detections in JavaScript

Chapter 4

RESULT AND CONCLUSIONS

4.1 RESULTS

The Video Indexing project delivers a powerful and user-friendly solution for comprehensive video content analysis. Users can seamlessly upload videos through an intuitive web interface, and the application securely stores them in an Amazon S3 bucket, enforcing robust IAM roles and permissions. Leveraging the capabilities of Amazon Rekognition and integrated AWS services like SNS and SQS, the project enables the detection of labels, faces with emotions, text, and even celebrities within the videos. Real-time notifications keep users informed of analysis results. The web application, crafted with HTML, CSS, and JavaScript, offers an accessible platform for content moderation, sentiment analysis, and celebrity recognition. With an emphasis on security, scalability, and a rich feature set, the Video Indexing project stands as a sophisticated and versatile tool for efficiently analyzing and indexing video content on the cloud.



127.0.0.1:5500/viewer.html

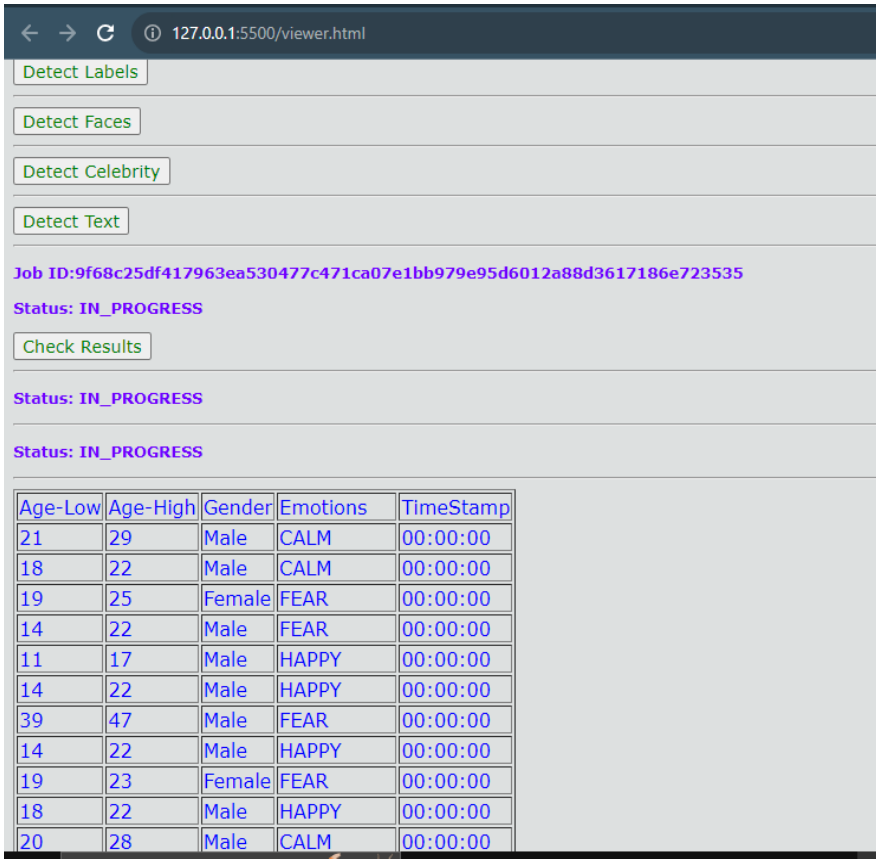
AWS Project By - Tanya Chhabhadiya

Video Indexing App

Please Choose a Video

Choose File london_faces.mp4

Fig 4.1 Video Indexing Application



Job ID:9f68c25df417963ea530477c471ca07e1bb979e95d6012a88d3617186e723535

Status: IN_PROGRESS

Status: IN_PROGRESS

Status: IN_PROGRESS

Age-Low	Age-High	Gender	Emotions	TimeStamp
21	29	Male	CALM	00:00:00
18	22	Male	CALM	00:00:00
19	25	Female	FEAR	00:00:00
14	22	Male	FEAR	00:00:00
11	17	Male	HAPPY	00:00:00
14	22	Male	HAPPY	00:00:00
39	47	Male	FEAR	00:00:00
14	22	Male	HAPPY	00:00:00
19	23	Female	FEAR	00:00:00
18	22	Male	HAPPY	00:00:00
20	28	Male	CALM	00:00:00

Fig 4.2 Results from the Application

4.2 SCOPE OF PROJECT

The Video Indexing project opens up promising avenues for future development and expansion. One potential avenue involves incorporating machine learning advancements to enhance the accuracy of video content analysis, enabling the system to adapt and improve over time. Integration with additional AWS services and complementary technologies could further extend the project's capabilities, such as incorporating natural language processing for improved text recognition or integrating with Amazon Comprehend for sentiment analysis. Exploring compatibility with emerging video formats and standards ensures the project stays ahead in the evolving landscape of multimedia content. Additionally, enhancing the user interface and experience, perhaps through interactive visualizations of analysis results, could contribute to a more engaging and insightful platform. Continuous monitoring, feedback incorporation, and staying abreast of technological advancements will be crucial for the sustained evolution and success of the Video Indexing project.

REFERENCES

1. https://docs.aws.amazon.com/AWSJavaScriptSDK/latest/AWS/Rekognition.html
2. https://medium.com/feedium/searching-on-medium-the-medium-search-function-3186cef4effd
3. https://aws.amazon.com/rekognition/video-features/
4. https://aws.amazon.com/rekognition/
5. https://docs.aws.amazon.com/sns/