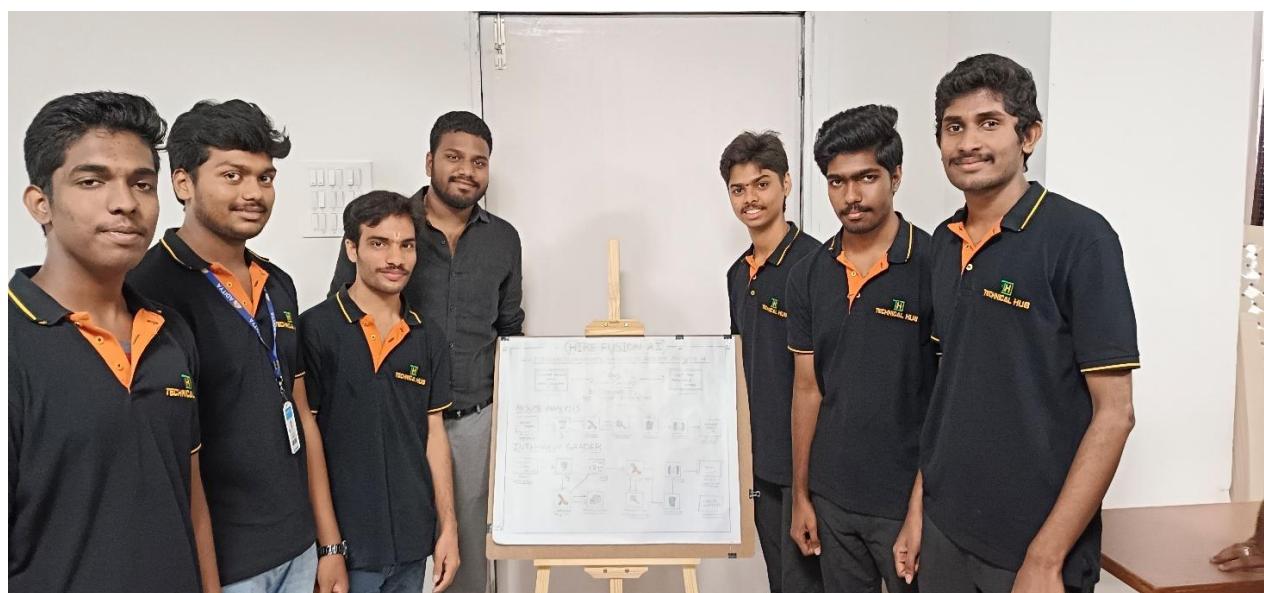


# HIREFUSION-AI

## AI POWERED CANDIDATE EVALUATION & RESUME ANALYSIS

### Team Members:



- Member 1 (**SAYYAD SAMEER**)
- Member 2 (**MURAMALLA AMBICA SAIRAM**)
- Member 3 (**KANTIPUDI VIVEK VARDHAN**)
- Member 4 (**BOGAVALLI AKASH**)
- Member 5 (**GODE SIVA RAMA KRISHNA DURGA PRASAD**)
- Member 6 (**BIKKAVOLU S SAI VEERA BHADRA AYYAN**)

**Mentor:**

- **MOHAMMAD SHAIFU ZAMA**

**Support Mentors:**

- **Aravind Pappala**
- **Durga Prasad Setti**
- **Pavan Teja Rayavarapu**
- **Bobby Pamarthi**

**Organization:**

- **Technical Hub Private Limited**
- **Aditya College Of Engineering and Technology, Surampalem**

**Academic Year: 2023-2027**

**Acknowledgement:**

We express our heartfelt gratitude to our mentor **MOHAMMAD SHAIFU ZAMA** for their continuous guidance, encouragement, and invaluable support throughout the development of this project.

We would also like to extend our sincere thanks to our supporting mentors **Aravind Pappala**, **Durga Prasad Setti**, **Pavan Teja Rayavarapu**, and **Bobby Pamarthi** for their insightful suggestions and timely help at various stages of the project.

We are especially grateful to **Mr. Babji Neelam**, CEO of **Technical Hub**, for his visionary leadership and constant encouragement that inspired us to deliver this project successfully.

Finally, we are grateful to **Technical Hub** and our institution for providing the necessary infrastructure, resources, and a collaborative environment that made the successful completion of this project possible.

## ABSTRACT

The recruitment process in organizations often involves manual screening of resumes and interviews, which is time-consuming and prone to bias. HireFusion is an AI-powered automated recruitment tool built using AWS Serverless architecture to address this challenge.

This project automates the analysis of resumes and interview videos using AI services like Amazon Textract, Rekognition, Comprehend, and Transcribe, with data stored securely in DynamoDB. The system provides real-time scoring and analytics through a web dashboard hosted on Amazon S3.

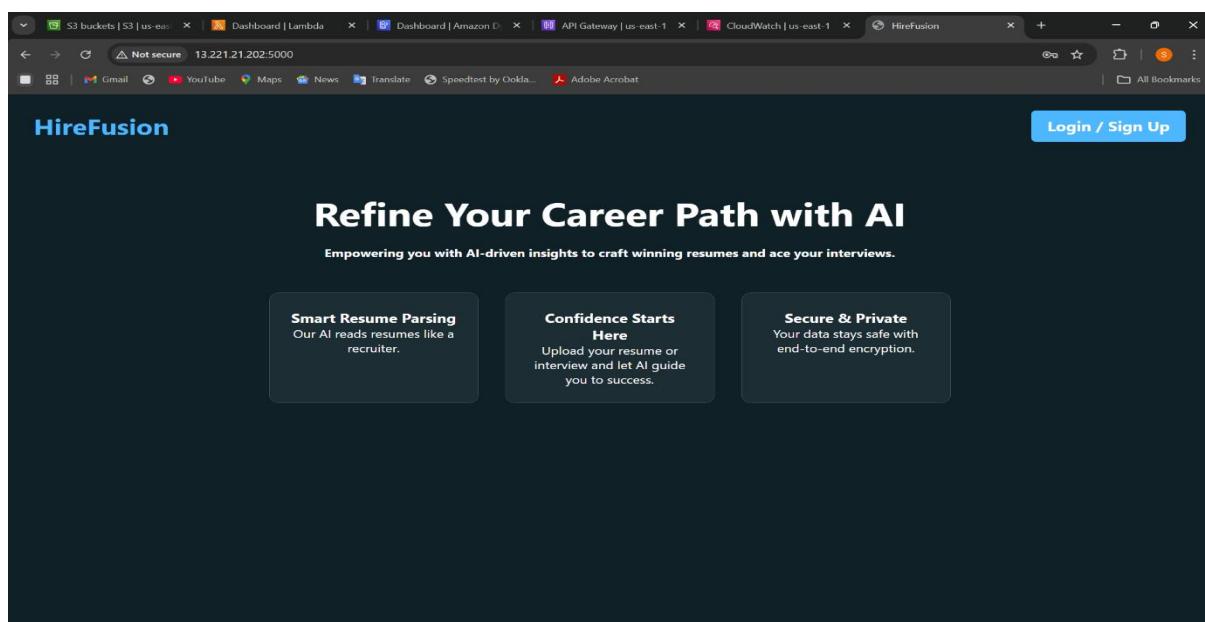
### Key features:

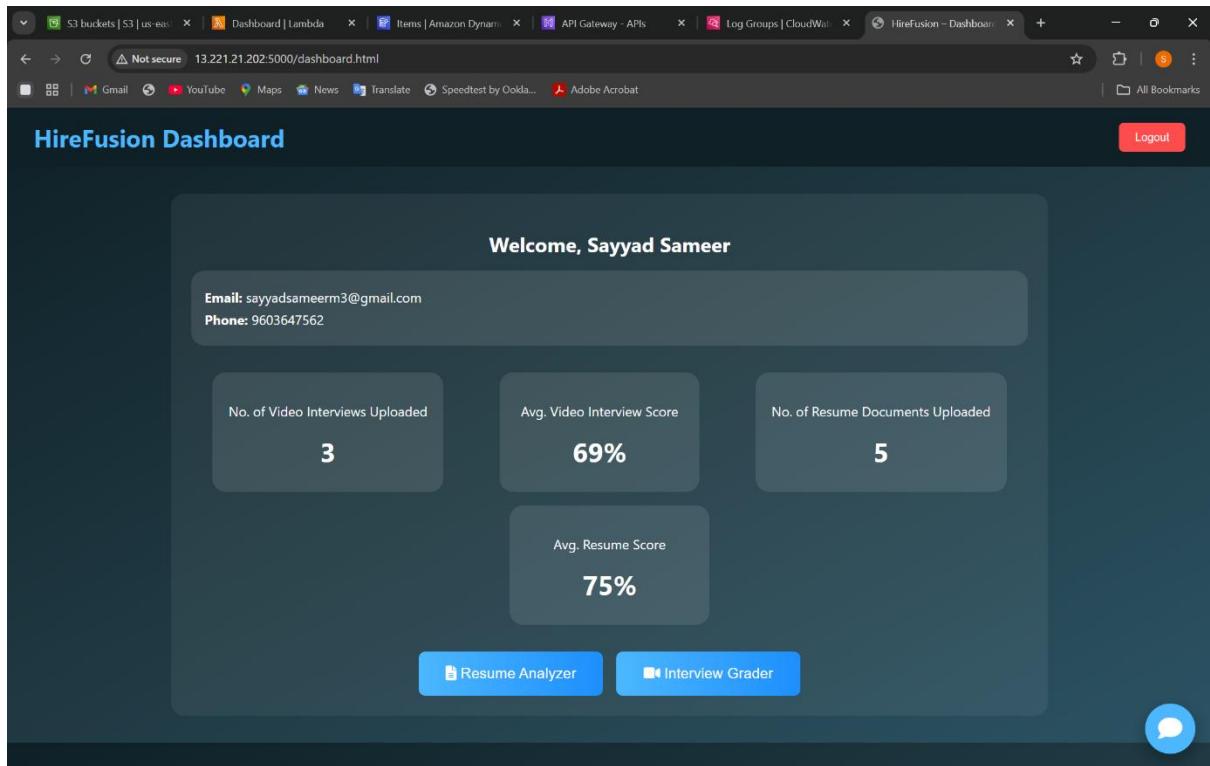
- Automated Resume Analysis: Extracts skills and calculates a score using Textract and Comprehend.
- Interview Grader: Analyzes facial expressions, hand gestures, and speech confidence using Rekognition and Transcribe.
- Serverless & Scalable: Built on AWS with Lambda triggers, event notifications, and API Gateway for APIs.
- Secure Authentication: User login monitoring with CloudWatch.

---

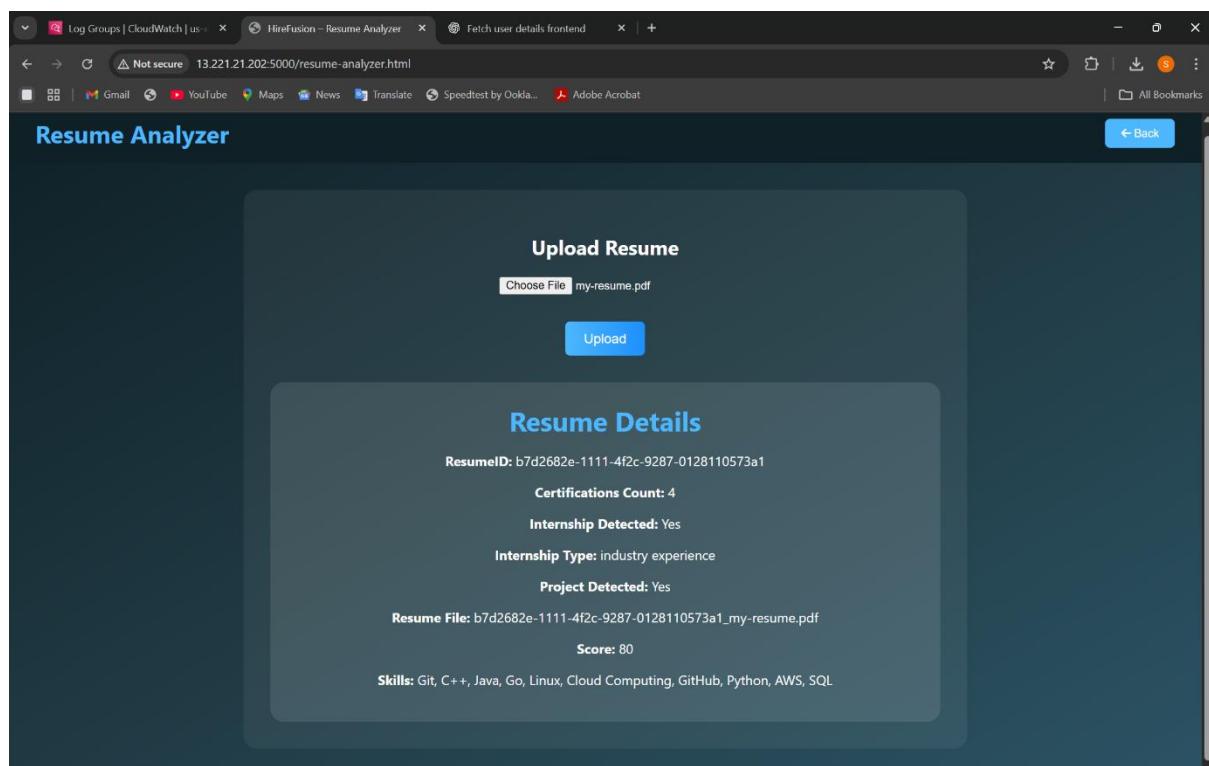
## Project Overview

**HireFusion** is an AI-driven platform designed to automate and improve the hiring process by analyzing resumes and interview videos. The system leverages multiple AWS services to provide real-time analysis, detect skills in resumes, and assess candidate performance in interview videos.

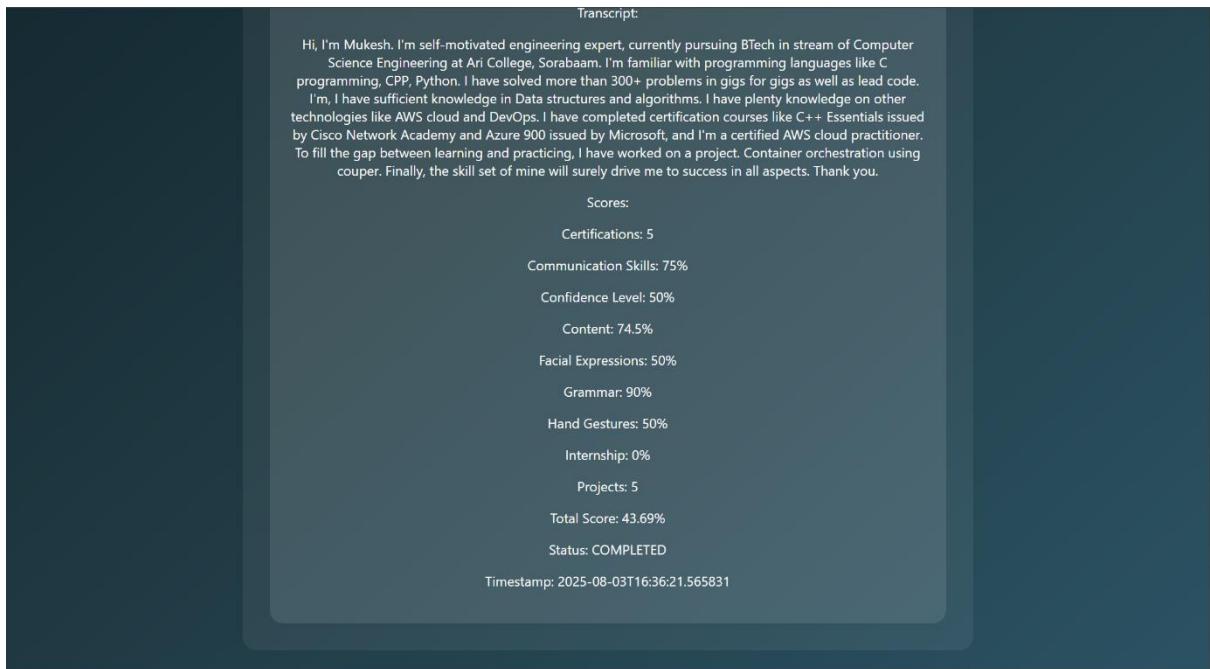
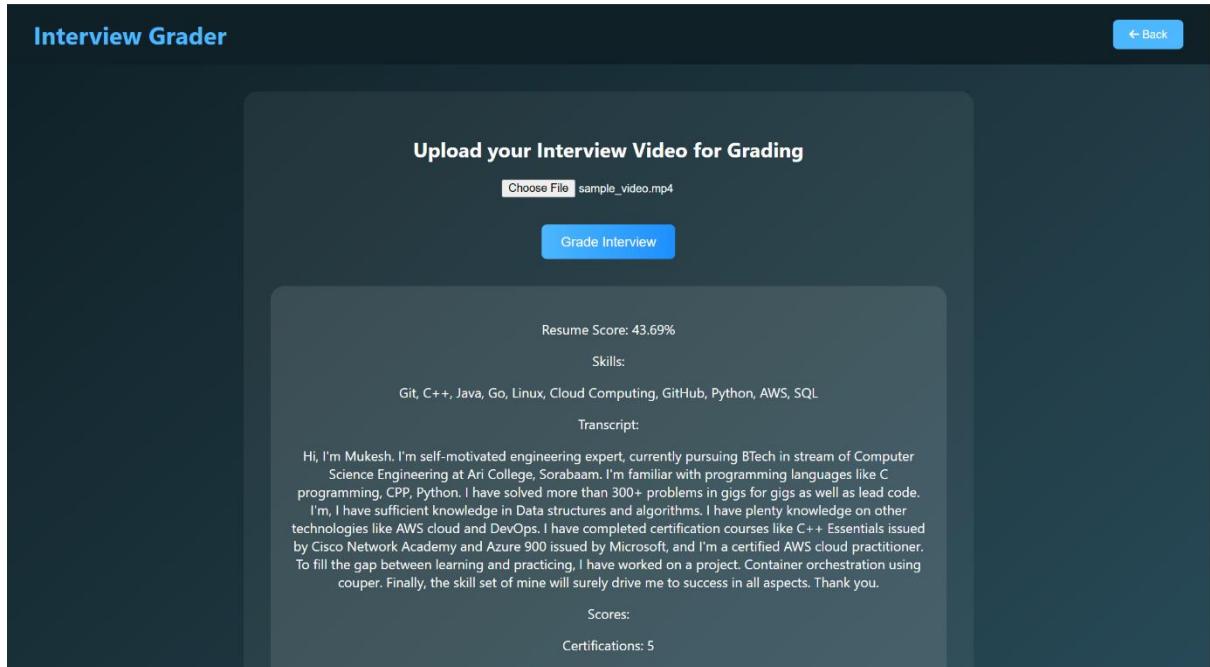




- **Resume Analyzer:** Automatically extracts and analyzes skills and relevant information from resumes.



- **Interview Grader:** Analyzes interview videos based on facial expressions, hand gestures, and transcribed text to grade candidate performance.



## **INTRODUCTION**

### **Problem Statement**

Hiring teams spend hours manually analyzing resumes and interview performance, resulting in delays and inconsistencies. Existing tools do not fully automate the process, and many require heavy infrastructure.

### **Purpose & Scope**

The purpose of this project is to create an AI-powered recruitment assistant that automates resume analysis and interview grading, reducing manual effort and improving accuracy.

#### **Scope:**

- Candidate uploads resume/video → AI extracts data & scores → Real-time dashboard display
- Fully serverless architecture with zero infrastructure management
- Easy integration with corporate HR tools in the future

#### **Objectives**

- Build a web-based platform for resume & interview uploads
- Use AWS AI services for text, speech, and facial analysis
- Provide instant scores and analytics
- Implement serverless triggers to eliminate infrastructure overhead

### **Existing System vs Proposed System**

#### **Existing System:**

- Manual screening, time-consuming
- High recruiter bias
- Requires large infrastructure for automation

#### **Proposed System:**

- Automated using AI & ML
- Reduces human error and time
- Scalable and cost-efficient with AWS Lambda

## **SYSTEM REQUIREMENTS**

### **Functional Requirements**

- Upload resumes (PDF/DOCX) and videos (MP4)
- Trigger Lambda functions via S3 Event Notifications
- Extract text and entities (skills) from resumes
- Analyze emotions, hand gestures, and speech from videos
- Store results in DynamoDB
- Display scores in the user dashboard

### **Non-Functional Requirements**

- Security: Cognito-based user authentication
- Performance: Analysis completed in <5 seconds per file
- Scalability: Serverless services auto-scale based on load
- Reliability: CloudWatch monitoring and error handling

## **Software & Hardware Requirements**

<b>Category</b>	<b>Requirement</b>
Programming Lang	Python (Lambda), JavaScript (Frontend)
AWS Services	S3, Lambda, DynamoDB, Textract, Rekognition, Transcribe, API Gateway, Cognito, CloudWatch, SNS
Frontend	HTML, CSS, JS
Hardware	AWS Cloud Infrastructure

---

## Services and Technologies Used

### AWS Lambda

- Purpose:** Host business logic for analyzing resumes and interview videos. It automatically scales based on file uploads.
- Key Features:** Serverless, event-driven, cost-efficient.

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with 'Lambda' selected under 'Functions'. The main area is titled 'Functions (3)' and lists three functions: 'processor', 'HireFusionResumeLambda', and 'HireFusionInterviewLambda'. Each function entry includes columns for 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. The 'HireFusionResumeLambda' and 'HireFusionInterviewLambda' functions were created yesterday. On the right side, there are 'Info' and 'Tutorials' tabs. The 'Tutorials' tab is active, displaying a section titled 'Create a simple web app' with a brief description and two bullet points: 'Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage' and 'Invoke your function through its function URL'. There are also 'Learn more' and 'Start tutorial' buttons.

### Amazon Textract

- Purpose:** Automatically extracts text from resumes in PDF or DOCX formats.
- Key Features:** Extracts text, tables, and forms for structured data.

The screenshot shows the Amazon Textract 'Analyze Document' page. On the left, there's a sidebar with 'Demos' and 'Analyze Document' selected. The main area has a heading 'Analyze Document' with an 'Info' link. Below it, a sub-instruction says 'Choose a sample document, or upload your own, to view the result from the Analyze Document API.' A large preview window shows a resume document titled 'form\_1005'. To the right of the preview, there are tabs for 'Raw text', 'Layout', 'Forms', 'Tables', and 'Queries'. The 'Raw text' tab is active, showing the raw text extracted from the resume. The 'Results' section below contains several boxes: 'Request for Verification of Employment' (with a note about Privacy Act Notice), 'Not be disclosed outside the agency except as required and permitted by law', 'The information requested in this form is authorized by Title 38, USC, Chapter 37 (if VA); by 12 USC, Section 1701 et. seq. (if HUD/FHA); by 42 USC, Section 1452b (if HUD/CPD); and Title 42 USC, 1471 et. seq., or 7 USC, 1921 et. seq. (if USDA/FmHA)', and 'Instructions: Lender - Complete items 1 through 7. Have applicant complete item 8. Forward directly to employer named in item 1.'

## Amazon Comprehend

- **Purpose:** Extracts entities and skills from the extracted text.
- **Key Features:** Natural Language Processing (NLP) service to detect entities like "skills" in resumes.

The screenshot shows the Amazon Comprehend console interface. On the left, a sidebar lists options like 'Real-time analysis', 'Customization', and 'Endpoints'. The main area has a text input box containing a sample text about a credit card payment and a spa visit. Below the text is a button to 'Analyze' it. The results are presented in an 'Insights' section with tabs for 'Entities', 'Key phrases', 'Language', 'PII', 'Sentiment', 'Targeted sentiment', and 'Syntax'. The 'Entities' tab is selected, showing detected entities in the analyzed text. A detailed breakdown of the analyzed text is also provided.

## Amazon Rekognition

- **Purpose:** Analyzes interview videos to detect facial expressions and emotions.
- **Key Features:** Provides facial analysis, including sentiment detection (e.g., Happy, Sad, Angry).

The screenshot shows the Amazon Rekognition console. The left sidebar includes sections for 'Custom Moderation', 'Custom Labels', 'Demos' (with 'Facial analysis' selected), 'Video Demos', and 'Metrics'. The main area features a 'Facial analysis' section with a preview image of a woman driving a car. Below the preview are options to 'Choose a sample image' or 'Use your own image' (with a file upload field). To the right, the 'Results' section displays a table of detected facial attributes with their confidence scores. The table includes rows for gender, age range, expression, and specific features like glasses and mouth status.

Attribute	Confidence Score
looks like a face	99.9 %
appears to be female	98.7 %
age range	24 - 30 years old
smiling	88.9 %
appears to be happy	99.6 %
wearing glasses	100 %
wearing sunglasses	100 %
eyes are open	100 %
mouth is open	99.8 %
does not have a mustache	99.9 %

## Amazon Transcribe

- **Purpose:** Converts speech from interview videos into text.
- **Key Features:** Real-time transcription, supports multiple languages.

The screenshot shows the Amazon Transcribe Real-time transcription interface. On the left, a sidebar lists various services: Get started, Real-time transcription (selected), Transcription jobs, Custom language model, Custom vocabulary, Vocabulary filtering, Amazon Transcribe Call Analytics, Real-time Analytics, Post-call Analytics, Category Management, Amazon Transcribe Medical, Real-time transcription, Transcription jobs, Custom vocabulary, AWS HealthScribe (New), Medical Insights (Demo), and Transcription jobs. The main content area is titled "Real-time transcription" with an "Info" link. It contains a "Transcription" section with a "Download full transcript" button and an orange "Start streaming" button. Below this is a "Transcription output" box showing the text "This is a test. But". To the right, it says "Current language: English, US". At the bottom, it shows "00:00 of 15:00 min audio stream" and a "Want to improve results?" link. A "Language settings" dropdown is also visible.

## Amazon S3

- **Purpose:** Stores resume files and interview videos uploaded by users.
- **Key Features:** Secure and scalable storage for large files.

The screenshot shows the Amazon S3 General purpose buckets interface. On the left, a sidebar lists: General purpose buckets, Directory buckets, Table buckets, Vector buckets (Preview), Access Grants, Access Points (General Purpose Buckets, FSx file systems), Access Points (Directory Buckets), Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (Dashboards, Storage Lens groups, AWS Organizations settings), and a Feature spotlight section with 11 items. The main content area shows "General purpose buckets (2)" with an "Info" link. It includes a "Create bucket" button and a "Find buckets by name" search bar. A table lists two buckets: "hirefusion-interview-videos" (Name, AWS Region: US East (N. Virginia) us-east-1, Creation date: July 29, 2025, 10:56:01 (UTC+05:30)) and "hirefusional-resumes" (Name, AWS Region: US East (N. Virginia) us-east-1, Creation date: July 29, 2025, 11:43:36 (UTC+05:30)). To the right, there are two callout boxes: "Account snapshot" (Info, Updated daily, View dashboard) which describes Storage Lens as providing visibility into storage usage and activity trends; and "External access summary - new" (Info, Updated daily) which describes External access findings to identify bucket permissions that allow public access or access from other AWS accounts.

## Amazon DynamoDB

- **Purpose:** Stores resume and video analysis results, including skills, scores, and facial analysis.
- **Key Features:** NoSQL database with high availability and low-latency access.

The screenshot shows the AWS DynamoDB console. On the left, there's a sidebar with links like Dashboard, Tables, DAX, Clusters, Subnet groups, Parameter groups, and Events. The main area is titled 'Tables (3) Info' and lists three tables: 'HireFusionTable', 'ResumeAnalysisResults', and 'VideoResumeAnalysisResults'. Each table row includes columns for Name, Status, Partition key, Sort key, Indexes, Replication Regions, and Deletion protection. A 'Create table' button is visible at the top right of the table list.

## API Gateway

- **Purpose:** Exposes Lambda functions as HTTP APIs, making them accessible from the frontend.
- **Key Features:** API management, routing, and authorization.

The screenshot shows the AWS API Gateway console. On the left, there's a sidebar with links for APIs, Custom domain names, Domain name access associations, VPC links, Usage plans, API keys, Client certificates, and Settings. The main area is titled 'APIs (1/1)' and lists one API named 'hirefusionapi'. The table columns include Name, Description, ID, Protocol, API endpoint type, and Created. A 'Create API' button is visible at the top right of the table list.

## Amazon CloudWatch

- **Purpose:** Monitors and logs application behavior, including Lambda function executions and AWS service performance.
- **Key Features:** Provides logs, metrics, and alarms to ensure smooth operation, helps track errors, performance, and optimize Lambda functions.

The screenshot shows the CloudWatch Log groups page. On the left, there's a navigation sidebar with sections like AI Operations, Alarms, Logs (selected), Metrics, Application Signals (APM), Network Monitoring, and Insights. The main area displays a table titled "Log groups (3)". The table has columns for Log group, Log class, Anomaly d..., Data p..., Sensit..., Retenti..., and Metric. Three log groups are listed:

Log group	Log class	Anomaly d...	Data p...	Sensit...	Retenti...	Metric
/aws/lambda/HireFusionInterviewLambda	Standard	Configure	-	-	Never exp...	-
/aws/lambda/HireFusionResumeLambda	Standard	Configure	-	-	Never exp...	-
/aws/lambda/processor	Standard	Configure	-	-	Never exp...	-

## Architecture Overview

The architecture of HireFusion is designed for scalability, security, and efficiency. The system operates as follows:

### Resume Analyzer Flow:

1. **User Uploads Resume:** The user uploads their resume to the S3 bucket `hirefusion-resumes`.

The screenshot shows the Amazon S3 Objects page for the bucket `hirefusionai-resumes`. The table lists one object:

Name	Type	Last modified	Size	Storage class
rajaresume.pdf	pdf	August 3, 2025, 22:08:26 (UTC+05:30)	109.2 KB	Standard

## 2. Lambda Triggered: The S3 Event Notification triggers the Resume Analyzer Lambda.

### Event notifications (1)

Send a notification when specific events occur in your bucket. [Learn more](#)

[Edit](#)
[Delete](#)
[Create event notification](#)

The screenshot shows the AWS Lambda console interface. At the top, there's a navigation bar with 'Lambda' and 'Functions' selected. Below it, the function name 'HireFusionResumeLambda' is displayed. On the left, a sidebar shows triggers: 'API Gateway' and 'S3'. The main area has tabs for 'Info' (selected) and 'Tutorials'. Under 'Tutorials', there's a section titled 'Create a simple web app' with a 'Start tutorial' button. The 'Code source' tab is selected in the bottom navigation bar. The code editor shows a Python file named 'lambda\_function.py' with the following code:

```

def generate_score(skills, text):
    cert_count = text.lower().count("certificate") + text.lower().count("certification")
    score = 10 + len(skills) * 3
    if project_flag:
        score += 10
    if internship_flag:
        score += 10
    return score

```

Below the code editor, there's a 'TEST EVENTS' section where a test event named 'resume' is defined as a JSON object:

```

{
  "BUCKET": "hirefusionai-resumes",
  "object": {
    "key": "rajaresume.pdf"
  }
}

```

The status of the function is listed as 'Succeeded'.

3. **Text Extraction:** The Lambda function uses **Textract** to extract text from the resume.
4. **Skills Extraction:** **Comprehend** detects skills and other entities from the extracted text.
5. **Score Generation:** A score is generated based on the number of skills detected in the resume.
6. **Data Storage:** The result, including skills and score, is stored in **DynamoDB**.

**Table: ResumeAnalysisResults - Items returned (6)**

ResumelD (String)	CertDetected	ResumeFile	Score	Skills
83d5ebdc-192b-4fef-a29a-92d18bb0e368		rajaresume.pdf	38	["React", "G...
8e0505da-a36e-4512...		8e0505da-a...	50	["Networki...
c2017675-3990-43f2...		c2017675-3...	82	["GitHub", "...
78cde4ee-ac8e-4019-...		78cde4ee-ac...	82	["GitHub", "...
960edaab-0866-474e...		960edaab-0...	80	["Cloud Co...
d914c2cb-b9b3-4274...		d914c2cb-b...	50	["Networki...

Attribute name	Value	Type
ResumelD - Partition key	83d5ebdc-192b-4fef-a29a-92d18bb0e368	String
CertificationsCount	0	Number
InternshipDetected	<input type="radio"/> True <input checked="" type="radio"/> False	Boolean
InternshipType	None	String
ProjectDetected	<input checked="" type="radio"/> True <input type="radio"/> False	Boolean
ResumeFile	rajaresume.pdf	String
Score	38	Number
Skills	["React", "Go", "Java", "JavaScript", "Git", "Python"]	String

Objects (1)

Name	Type	Last modified	Size	Storage class
sample_video.mp4	mp4	August 3, 2025, 22:05:31 (UTC+05:30)	3.1 MB	Standard

## 2. Lambda Triggered: The S3 Event Notification triggers the Interview Grader Lambda.

Event notifications (1)

Name	Event types	Filters	Destination type	Destination
interview	Put	.mp4	Lambda function	HireFusionInterviewLambda

The test event "resume" was successfully saved.

**HireFusionInterviewLambda**

Layers (0)

+ Add destination

API Gateway

S3

+ Add trigger

Code | Test | Monitor | Configuration | Aliases | Versions

Code source | Info

lambda\_function.py

```
1 import boto3
2 import os
3 import uuid
4 import uniclib_parse
```

Open in Visual Studio Code | Upload from

Last modified 3 days ago

Function ARN arn:aws:lambda:us-east-1:785269092008:function:HireFusionInterviewLambda

Function URL | Info

Tutorials

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Start tutorial

```

import boto3
import os
import uuid
import urllib.parse
import json

rekognition = boto3.client('rekognition')
transcribe = boto3.client('transcribe')

SNS_TOPIC_ARN = os.environ['SNS_TOPIC_ARN']

def lambda_handler(event, context):
    try:
        # Get S3 object details from event
        record = event['Records'][0]
        bucket = record['s3']['bucket']
        key = urllib.parse.unquote_plus(record['s3']['object']['key'])
    
```

3. **Facial Expression and Gesture Analysis:** The Lambda function uses **Rekognition** to detect facial expressions and gestures.
4. **Speech Transcription:** The Lambda function uses **Transcribe** to convert speech from the video into text.
5. **Skill Detection:** **Comprehend** analyzes the transcribed text for skills or sentiment.

ID	Endpoint	Status	Protocol
a26a083c-22b9-4c30-a06b-32...	arn:aws:lambda:us-east-1:7852...	Confirmed	LAMBDA

Last modified 2 days ago

**Function ARN** arn:aws:lambda:us-east-1:78526909200:processor:processor

**Function URL** -

**Tutorials**

Learn how to implement common use cases in AWS Lambda.

**Create a simple web app**

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#)

[Start tutorial](#)

**6. Score Generation:** The final score is based on facial expression, gesture analysis, and the sentiment from the transcription.

**7. Data Storage:** The analysis results are stored in DynamoDB.

**DynamoDB**

- Dashboard
- Tables
- Explore items**
- PartQL editor
- Backups
- Exports to S3
- Imports from S3
- Integrations
- Reserved capacity
- Settings

**VideoResumeAnalysisResults**

**Scan or query items**

**Completed** - Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCU consumed: 2

AnalysisID (String)	scores	status	timestamp	total_score
VideoResumeAnalysisResults	{} (1 item)	{} (1 item)	{} (1 item)	{} (1 item)

aws Search [Alt+S] United States (N. Virginia) Sayyad Sameer

DynamoDB > Explore items: VideoResumeAnalysisResults > Edit item

### Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

[Form](#) | [JSON view](#)

[Add new attribute](#)

Attribute name	Value	Type	<a href="#">Remove</a>
AnalysisID - Partition key	2363d817-1f5a-488f-8b2d-d90a8d6f6501	String	
scores	<a href="#">Insert a field</a>	Map	<a href="#">Remove</a>
certifications	5	Number	<a href="#">Remove</a>
communication_skills	75	Number	<a href="#">Remove</a>
confidence_level	50	Number	<a href="#">Remove</a>
content	74.5	Number	<a href="#">Remove</a>
facial_expressions	50	Number	<a href="#">Remove</a>
grammar	90	Number	<a href="#">Remove</a>
hand_gestures	50	Number	<a href="#">Remove</a>

aws Search [Alt+S] United States (N. Virginia) Sayyad Sameer

DynamoDB > Explore items: VideoResumeAnalysisResults > Edit item

### Edit item

[Add new attribute](#)

Attribute name	Value	Type	<a href="#">Remove</a>
internship	0	Number	<a href="#">Remove</a>
projects	5	Number	<a href="#">Remove</a>
total_score	43.69	Number	<a href="#">Remove</a>
status	COMPLETED	String	<a href="#">Remove</a>
timestamp	2025-08-03T16:36:21.565831	String	<a href="#">Remove</a>
total_score	43.69	Number	<a href="#">Remove</a>
transcript	Hi, I'm Mukesh. I'm self-motivated engineering expert, currently pursuing BTech in stream of Computer Science Engineering at Ari College, Sonabaam. I'm familiar with programming languages like C programming, CPP, Python. I have solved more than 300+ problems in gigs as well as lead code. I'm, I have sufficient knowledge in Data structures and algorithms. I have plenty knowledge on other technologies like AWS cloud and DevOps. I have completed certification courses like C++ Essentials issued by Cisco Network Academy and Azure 900 issued by Microsoft, and I'm a certified AWS cloud practitioner. To fill the gap between learning and practicing, I have worked on a project. Container orchestration using couper. Finally, the skill set of mine will surely drive me to success in all aspects. Thank you.	String	<a href="#">Remove</a>
video	s3://hirefusion-interview-videos/sample_video.mp4	String	<a href="#">Remove</a>

## 8. API Response: The score and analysis are returned to the frontend.

---

## Testing

### Testing Approach

The testing of **HireFusion-AI** focused on validating the functionality of Lambda triggers, AI service integrations, and the end-to-end flow using **CloudWatch Logs** and **DynamoDB records**. Since Postman or manual API testing tools were not used, the testing was primarily **event-driven** using S3 uploads.

---

### Test Scenarios

#### 1. Resume Analyzer Flow

- **Test Case 1:** Upload a valid PDF/DOCX resume to the hirefusion-resumes S3 bucket.
    - **Expected Result:**
      - S3 event triggers Resume Analyzer Lambda.
      - Lambda uses Textract → Comprehend → generates skills and score.
      - DynamoDB table stores the record.
      - CloudWatch Logs show successful execution.
    - **Result:** Succeeded
  - **Test Case 2:** Upload a corrupted or unsupported resume file.
    - **Expected Result:** Lambda logs error in CloudWatch, record not stored in DynamoDB.
    - **Result:** Succeeded (Error captured in logs)
- 

#### 2. Interview Grader Flow

- **Test Case 3:** Upload a valid MP4 video to the hirefusion-interview-videos S3 bucket.
  - **Expected Result:**
    - Lambda triggers Rekognition (facial analysis) and Transcribe (speech-to-text).
    - Generates score based on emotion confidence, speech sentiment.
    - Stores results in DynamoDB.

- CloudWatch Logs confirm each step.
- **Result:** Succeeded
- **Test Case 4:** Upload very large video (>100MB).
  - **Expected Result:** Lambda times out or fails gracefully; error logged in CloudWatch.
  - **Result:** Succeeded

```

 1 import json
 2 import boto3
 3 import time
 4
 5 # AWS Clients
 6 s3 = boto3.client('s3')
 7 textract = boto3.client('textract')
 8 dynamodb = boto3.resource('dynamodb')
 9 table = dynamodb.Table('ResumeAnalysisResults')
10
11 # Skills list
12 SKILL_KEYWORDS = [
13     "AWS", "Azure", "GCP", "Google Cloud",
14     "Ansible", "CI/CD", "Jenkins", "GitHub",
15     "TypeScript", "C++", "C#", "Go", "Ruby",
16     "Spring Boot", "Django", "Flask", "Elixir",
17     "Redis", "Elasticsearch", "Machine Learning",
18     "scikit-learn", "Pandas", "NumPy", "TensorFlow"
]

```

## CloudWatch Log Monitoring

- **CloudWatch Logs** were actively used to:
  - Verify **event triggers** from S3 notifications.
  - Check AI service responses from Textract, Comprehend, Rekognition, and Transcribe.
  - Identify **errors** like AccessDenied, timeout errors, or unsupported file formats.
  - Ensure DynamoDB put\_item was successful.

```

▶ 2025-07-31T14:51:36.896Z      START RequestId: 407a3d2c-e490-4773-acfe-24a5026814ac Version: $LATEST
▶ 2025-07-31T14:51:36.898Z      [DEBUG] Incoming Event: {"Records": [{"EventSource": "aws:sns", "EventVersion": "1.0", "EventSubscriptionArn": "arn:aws:sns:us-east-1:123456789012:my-topic", "Message": "Job posted for developer role in New York."}], "RequestID": "407a3d2c-e490-4773-acfe-24a5026814ac", "Timestamp": "2025-07-31T14:51:36.898Z", "Region": "us-east-1", "Source": "Lambda", "Type": "CloudWatchLogs", "LogGroup": "/aws/lambda/HireFusion-Resume-Analyzer", "LogStream": "407a3d2c-e490-4773-acfe-24a5026814ac", "LogFile": "main.log", "LogLevel": "INFO", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Incoming Event: {"Records": [{"EventSource": "aws:sns", "EventVersion": "1.0", "EventSubscriptionArn": "arn:aws:sns:us-east-1:123456789012:my-topic", "Message": "Job posted for developer role in New York."}], "RequestID": "407a3d2c-e490-4773-acfe-24a5026814ac", "Timestamp": "2025-07-31T14:51:36.898Z", "Region": "us-east-1", "Source": "Lambda", "Type": "CloudWatchLogs", "LogGroup": "/aws/lambda/HireFusion-Resume-Analyzer", "LogStream": "407a3d2c-e490-4773-acfe-24a5026814ac", "LogFile": "main.log", "LogLevel": "INFO", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Parsed SNS Message: {"JobId": "7c61ce1158480f254abfad596b0c90e5b325e163a3ac58f35fcfc4f031cfdd41", "Status": "PENDING", "Timestamp": "2025-07-31T14:51:36.898Z", "Region": "us-east-1", "Source": "Lambda", "Type": "CloudWatchLogs", "LogGroup": "/aws/lambda/HireFusion-Resume-Analyzer", "LogStream": "407a3d2c-e490-4773-acfe-24a5026814ac", "LogFile": "main.log", "LogLevel": "INFO", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Getting Rekognition results for JobId=7c61ce1158480f254abfad596b0c90e5b325e163a3ac58f35fcfc4f031cfdd41"}, "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Total Faces Detected: 131", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Facial Score=75.0, Gesture Score=70.0", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Waiting for Transcribe Job=b3fc4295-8d39-485e-8717-bdbc6bd94776", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Transcribe Status=COMPLETED", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Transcript URI=https://s3.us-east-1.amazonaws.com/aws-transcribe-us-east-1-prod/785269092008/b3fc4295-8d39-485e-8717-bdbc6bd94776/transcripts/7c61ce1158480f254abfad596b0c90e5b325e163a3ac58f35fcfc4f031cfdd41", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Transcript Extracted: Hi, I'm Mukesh. I'm self-motivated engineering expert, currently pursuing BTech in stree...", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Sentiment=NEUTRAL", "LogLine": "2025-07-31T14:51:36.898Z [DEBUG] Final Score Object: {'facial_expressions': 75.0, 'hand_gestures': 70.0, 'confidence_level': 72.5, 'communicat...", "LogLine": "2025-07-31T14:51:36.898Z [ERROR] An error occurred (ValidationException) when calling the PutItem operation: One or more parameter values were invalid.", "LogLine": "END RequestId: 407a3d2c-e490-4773-acfe-24a5026814ac Duration: 1236.83 ms Billed Duration: 1237 ms Memory Size: 128 MB", "LogLine": "REPORT RequestId: 407a3d2c-e490-4773-acfe-24a5026814ac Duration: 1236.83 ms Billed Duration: 1237 ms Memory Size: 128 MB"}}

```

## Results & Outputs

The HireFusion-AI project was successfully implemented and validated using multiple test cases. The system performed resume analysis and interview video grading accurately and stored results in DynamoDB. Below are the outputs and proof of successful execution:

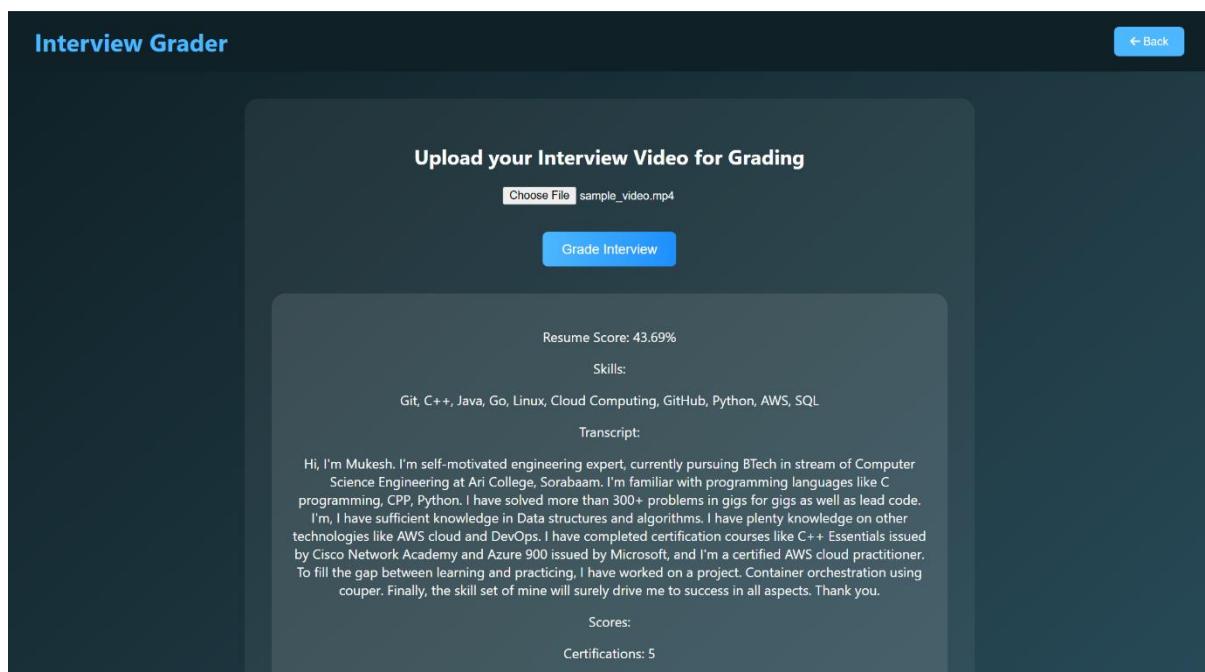
### Resume Analyzer Output

1. Users were able to upload resumes in PDF/DOCX format via the web UI.
2. The Resume Analyzer Lambda function extracted text using Amazon Textract, identified skills using Amazon Comprehend, and generated a score.
3. The final score was displayed on the web UI and stored in DynamoDB.

The screenshot shows a web browser window with the URL [13.221.21.202:5000/resume-analyzer.html](http://13.221.21.202:5000/resume-analyzer.html). The page has a dark blue header with the title 'Resume Analyzer'. Below the header, there's a 'Upload Resume' section with a 'Choose File' button set to 'my-resume.pdf' and a blue 'Upload' button. Underneath, a large rounded rectangle contains the 'Resume Details' section. It shows the 'ResumeID: b7d2682e-1111-4f2c-9287-0128110573a1', 'Certifications Count: 4', 'Internship Detected: Yes', 'Internship Type: industry experience', 'Project Detected: Yes', 'Resume File: b7d2682e-1111-4f2c-9287-0128110573a1\_my-resume.pdf', 'Score: 80', and 'Skills: Git, C++, Java, Go, Linux, Cloud Computing, GitHub, Python, AWS, SQL'.

## Interview Grader Output

1. Users uploaded interview videos (MP4) through the web interface.
2. The Interview Grader Lambda function analyzed facial expressions and gestures using Rekognition, converted audio to text using Transcribe, and extracted entities using Comprehend.
3. The final interview performance score was generated and displayed.



Transcript:

aws Search [Alt+S] United States (N. Virginia) Sayyad Sameer

DynamoDB > Explore items: VideoResumeAnalysisResults > Edit item

### Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

[Form](#) | [JSON view](#)

Attributes			<a href="#">Add new attribute</a>
Attribute name	Value	Type	
AnalysisID - Partition key	2363d817-1f5a-488f-8b2d-d90a8d6f6501	String	
scores	<a href="#">Insert a field</a>	Map	<a href="#">Remove</a>
certifications	5	Number	<a href="#">Remove</a>
communication_skills	75	Number	<a href="#">Remove</a>
confidence_level	50	Number	<a href="#">Remove</a>
content	74.5	Number	<a href="#">Remove</a>
facial_expressions	50	Number	<a href="#">Remove</a>
grammar	90	Number	<a href="#">Remove</a>
hand_gestures	50	Number	<a href="#">Remove</a>

Transcript:

aws Search [Alt+S] United States (N. Virginia) Sayyad Sameer

DynamoDB > Explore items: VideoResumeAnalysisResults > Edit item

### Edit item

You can add, remove, or edit the attributes of an item. You can nest attributes inside other attributes up to 32 levels deep. [Learn more](#)

[Form](#) | [JSON view](#)

Attributes			<a href="#">Add new attribute</a>
Attribute name	Value	Type	
internship	0	Number	<a href="#">Remove</a>
projects	5	Number	<a href="#">Remove</a>
total_score	43.69	Number	<a href="#">Remove</a>
status	COMPLETED	String	<a href="#">Remove</a>
timestamp	2025-08-03T16:36:21.565831	String	<a href="#">Remove</a>
total_score	43.69	Number	<a href="#">Remove</a>
transcript	<p>Hi, I'm Mukesh. I'm self-motivated engineering expert, currently pursuing BTech in stream of Computer Science Engineering at Ari College, Sorabaam. I'm familiar with programming languages like C programming, CPP, Python. I have solved more than 300+ problems in gigs for gigs as well as lead code. I'm, I have sufficient knowledge in Data structures and algorithms. I have plenty knowledge on other technologies like AWS cloud and DevOps. I have completed certification courses like C++ Essentials issued by Cisco Network Academy and Azure 900 issued by Microsoft, and I'm a certified AWS cloud practitioner. To fill the gap between learning and practicing, I have worked on a project. Container orchestration using couper. Finally, the skill set of mine will surely drive me to success in all aspects. Thank you.</p>	String	<a href="#">Remove</a>
video	s3://hirefusion-interview-videos/sample_video.mp4	String	<a href="#">Remove</a>

## **Team Member Roles & Contributions**

### **Member 1 ( SAYYAD SAMEER )– AWS Lambda & Event-Driven Backend Developer**

- Developed and deployed core AWS Lambda functions for Resume Analyzer and Interview Grader.
- Integrated event-driven architecture using S3 event notifications to automatically trigger Lambdas.
- Configured API Gateway endpoints for frontend-backend communication.
- Integrated all AWS services (Textract, Rekognition, Transcribe, Comprehend, DynamoDB).
- Managed CloudWatch logs, alarms, and error monitoring.
- Led cloud architecture design and AWS deployment.

### **Member 2 ( BOGAVALLI AKASH ) – Frontend Engineer**

- Built the user interface for resume and video uploads using HTML, CSS, and JavaScript.
- Integrated API calls for fetching scores and results from Lambda via API Gateway.
- Developed responsive layouts and designed user-friendly components for the dashboard and results page.
- Integrated Cognito-based user authentication and session control.

### **Member 3 ( KANTIPUDI VIVEK VARDHAN )– Cloud & IAM Configuration Specialist**

- Set up and configured all AWS services including S3 buckets, and DynamoDB tables.
- Created and managed IAM roles and permissions for secure Lambda and API access.
- Ensured security policies and least privilege principle were followed across services.
- Managed cross-service permissions and deployment setup.

**Member 4 (MURAMALLA AMBICA SAIRAM )– AI Scoring Strategy & NLP Logic Designer**

- Designed the logic and algorithm to calculate scores from AI service outputs (skills, expressions, transcription).
- Fine-tuned emotion-based grading rules (e.g., Happy = +10, Low confidence = -5).
- Configured AWS Comprehend usage for entity extraction (e.g., skills, sentiments).
- Provided mappings for sentiment/emotion weights and contributed to the business logic behind candidate evaluation.

**Member 5 ( GODE SIVA RAMA KRISHNA DURGA PRASAD )– Testing & Debugging Lead**

- Validated Lambda outputs using CloudWatch logs and manual test uploads.
- Monitored system behavior under different input scenarios (PDFs with no text, silent videos, etc.).
- Reported and documented system bugs, bottlenecks, and recommended improvements.
- Verified successful data storage in DynamoDB and correct API responses.

**Member 6 ( BIKKAVOLU S SAI VEERA BHADRA AYYAN ) – Documentation & Presentation Manager**

- Wrote project documentation, including architecture diagrams, flowcharts, AWS usage tables, and setup instructions.
  - Prepared the final presentation (PPT), README files, and user guide.
  - Organized screenshots, test outputs, and GitHub folder structure.
  - Led coordination for final project submission and demonstration.
-

## Learnings and Challenges

### Key Learnings:

- **Serverless Architecture:** Gained hands-on experience with AWS Lambda and other serverless services like S3, API Gateway, and DynamoDB. I learned how to create efficient, cost-effective applications with minimal infrastructure management.
- **AI Services:** I learned how to integrate AWS AI services like **Textract**, **Comprehend**, and **Rekognition** for real-world applications like document processing and video analysis.
- **Event-Driven Design:** I understood the importance of event-driven architecture in cloud applications, especially for triggering actions based on file uploads in S3 using event notifications.
- **Monitoring and Debugging:** I learned how to leverage **CloudWatch Logs** to monitor Lambda executions, track errors, and optimize the performance of AWS Lambda functions.

### Challenges:

- **Complexity of AI Integration:** Integrating multiple AI services like **Rekognition** and **Comprehend** for analyzing videos and resumes posed challenges, such as handling various data formats and optimizing performance.
- **Error Handling:** Ensuring that all Lambda functions handled errors gracefully, especially in complex scenarios where the processing could fail at any step (e.g., file format issues, API rate limits).

---

## Advantages of the System

- **Automation:** The system automates resume analysis and interview grading, making the hiring process faster and more efficient.
- **Scalability:** The serverless architecture ensures that the system can scale with the increasing number of resumes and videos uploaded.
- **AI-Driven:** By leveraging AWS AI services, the system performs complex tasks like skill extraction and emotion detection without human intervention.
- **Cost Efficiency:** Using AWS Lambda and other serverless services ensures minimal operational costs, as resources are only consumed when required.

## **Limitations:**

- Basic scoring system – does not fully consider experience or context.
  - Video analysis may miss subtle emotions or poor audio quality affects Transcribe.
  - Currently supports English only.
  - No feedback loop from users to improve accuracy.
- 

## **Future Enhancements**

- **Improved Scoring Mechanisms:** Currently, scoring is based on skills or facial expressions. A more complex scoring system could be developed that incorporates various factors such as experience, language proficiency, and sentiment.
  - **User Feedback:** Implement a feedback mechanism where users can validate the results of the analysis (resume or interview) to continuously improve the accuracy of the system.
  - **Expanded Support for Video Formats:** The system currently supports common video formats. Adding support for more formats and integrating multi-language transcription for international users would enhance the system.
  - Multi-language transcription.
  - More advanced scoring models using ML.
  - Real-time interviewer feedback system.
-

## **Summary**

The HireFusion-AI project was developed with the aim of revolutionizing the recruitment process by leveraging cloud computing and artificial intelligence. The system focuses on two primary modules:

1. Resume Analyzer – Automates the process of extracting information from resumes using Amazon Textract and Amazon Comprehend. It detects candidate skills, generates scores, and stores results in DynamoDB for future reference.
2. Interview Grader – Evaluates interview videos using Amazon Rekognition (for facial expressions and gestures), Amazon Transcribe (for speech-to-text conversion), and Comprehend (for sentiment and keyword analysis).

The entire project is built on AWS serverless architecture for scalability and cost-efficiency. S3 Event Notifications trigger Lambda functions, which process the uploaded files and store results in DynamoDB. The processed data is then displayed to users through APIs exposed via API Gateway, and user authentication is securely managed using AWS Cognito.

To ensure smooth operations, CloudWatch was integrated for detailed monitoring, logging, and setting alarms in case of system failures. The project's modular design allows future integrations with external HR tools and AI models.

The system significantly reduces manual effort in candidate evaluation, eliminates recruiter bias, and delivers consistent, data-driven results. This project demonstrates how modern recruitment processes can be automated using AI + Cloud technologies to streamline hiring.

---

## Conclusion

In conclusion, HireFusion-AI is a robust and scalable platform designed to transform traditional recruitment practices. By utilizing AWS serverless services, the system ensures minimal infrastructure management while providing high availability and reliability. The integration of AI services like Textract, Rekognition, Comprehend, and Transcribe enables automated, intelligent analysis of resumes and interviews, reducing hiring time and improving accuracy.

Key takeaways of the system include:

- Efficiency: Automates resume screening and interview evaluation, reducing recruiter workload.
- Accuracy: Data-driven scoring reduces bias and human error.
- Scalability: Serverless architecture allows the system to handle an increasing number of users without performance issues.
- Extensibility: Future enhancements, such as ML-based scoring models and real-time feedback, can be easily integrated.

This project successfully showcases the potential of cloud-based AI solutions for HR automation. Our team learned hands-on how to integrate multiple AWS services into a unified system, implement event-driven architecture, and monitor applications using CloudWatch. With further enhancements, HireFusion-AI can evolve into a full-fledged recruitment platform capable of supporting organizations worldwide.