AUTOMATED RESUME KEYWORD EXTRACTION (A.R.K.E) TOOL

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1.0 PROBLEM STATEMENT

Recruiters are currently facing an overwhelming number of resume submissions. Manually scanning resumes to locate key skills, job titles, and educational background is slow, tiring and error-prone, creating delays in the hiring process. Until the issue is solved, the efficiency and productivity of the human resource of the recruiting company could reduce sharply over a short period of time.

1.1.1 AIM OF THE PROJECT

This project, Automated Resume Keyword Extraction Tool (A.R.K.E), seeks to present a solution of manual processing of resumes by producing a tool using Amazon Web Services (AWS) to automate the process of scanning through resumes to locate key skills, job titles, and educational background to make the hiring process faster and smooth.

1.1.2 SPECIFIC OBJECTIVES

Before the development of any software application, code, or website, the developers ought to know what purpose their code, website, or software application is going to serve and the various objectives to be accomplished while interacting with their work. The objectives of this project are as listed below:

- The tool should be able to take resume uploads in pdf format.
- Process uploads and extract relevant key phrases.
- Store relevant details in its database.
- Provide end users with the ability to search, query, filter etc the database to shortlist candidates.
- The tool should also allow end-users download upload resumes.

1.1.3 REQUIREMENTS

The requirements under this section would be grouped into two (2), functional requirements and the non-functional requirements.

Functional Requirements

- Resume uploads
- Text Extraction
- Analysis on extracted text
- Data Storage
- Candidate Pre-Screening

Non Functional Requirements

- Performance
- Reliability
- Security
- Usability

1.1.4 PROJECT DELIVERABLES

The project aims to produce the following services as its end result to its end-users;

- A complete documentation of the project.
- A fully functional automated resume keyword extraction tool for resume processing and candidate pre-screening.

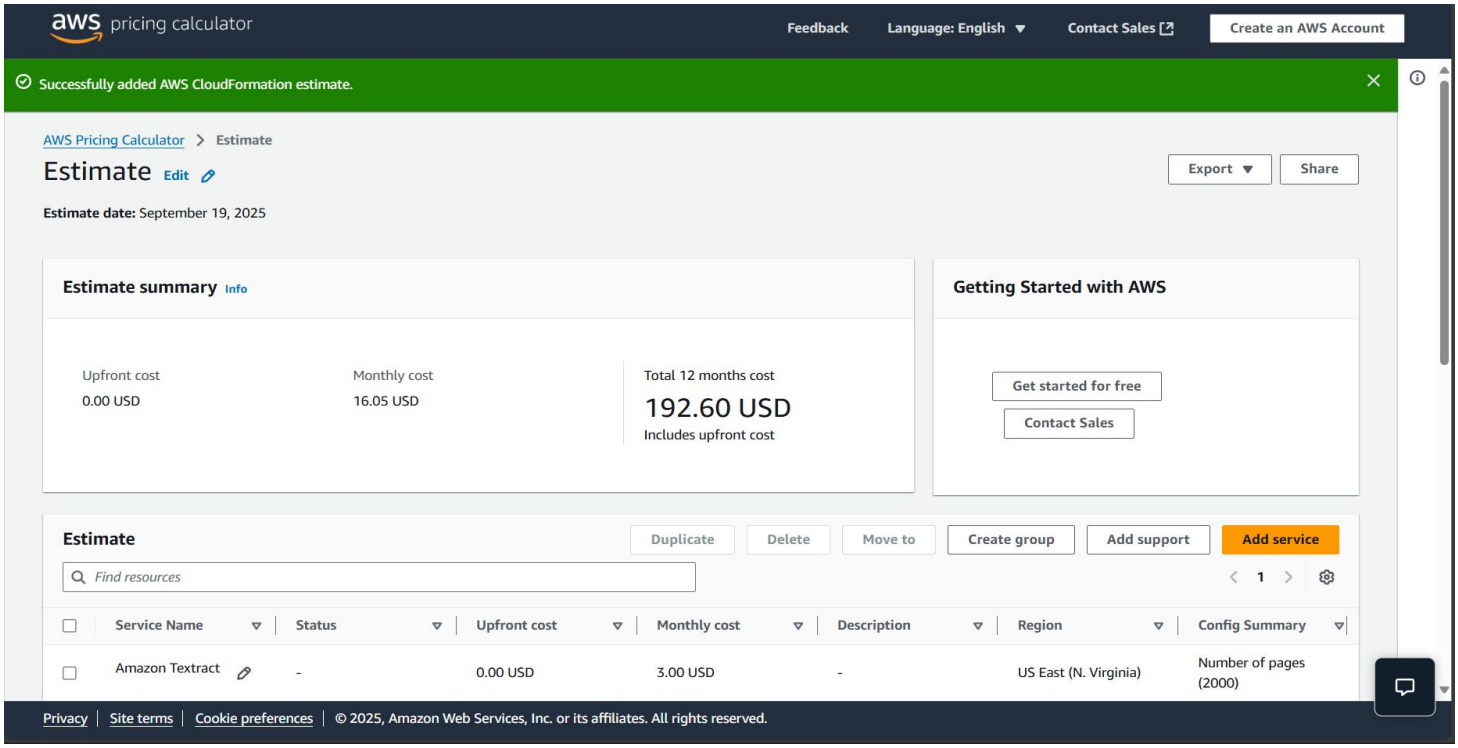
1.1.5 LIMITATIONS

- Limited time resource: The time resource constraint is a very crucial one as the project deliverables are to be produced in a very short time, meaning the development and implementation of this project ought to be rushed a little to meet the time limit.
- Troubleshooting : Also stemming down from the earlier stated limitation, limited time resource, the nonavailability of adequate helpful resources to assist with finding solutions to minor setbacks faced ate up into the set schedule.

1.1.6 ASSUMPTIONS

- End users already have a system that takes in resumes from applicants.
- End users' system performs some level of security checks by authorizing user credentials before allowing them enter the system.

1.1.7 ESTIMATED COST



aws

Contact your AWS representative: Contact Sales

Export Date: 09/19/2025 **Language:** English

Estimate url

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	16.04 USD	192.48 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon Textract	-	US East (N. Virginia)	0.00 USD	3.00 USD
Status	-			
Description:	-			
Config summary	Number of pages (2000)			

Name	Group	Region	Upfront cost	Monthly cost
Amazon Comprehend	-	US East (N. Virginia)	0.00 USD	5.01 USD
Status	-			
Description:	-			
Config summary	Number of documents asynchronous (250), Average characters in a document asynchronous (1500) Number of documents asynchronous (250), Average characters in a document asynchronous (1500) Model training run time (30 minutes), Model retention (1), Documents (250), Characters (1500) Number of documents asynchronous (250), Average characters in a document asynchronous (1500)			

Name	Group	Region	Upfront cost	Monthly cost
AWS Lambda	-	US East (N. Virginia)	0.00 USD	0.32 USD
Status	-			
Description:	-			
Config summary	Architecture (x86), Architecture (x86), Invoke Mode (Buffered), Amount of ephemeral storage allocated (512 MB), Number of requests (1000 per month), Concurrency (1000), Time for which Provisioned Concurrency is enabled (10 minutes), Number of requests for Provisioned Concurrency (1000 per month)			

Name	Group	Region	Upfront cost	Monthly cost
Amazon API Gateway	-	US East (N. Virginia)	0.00 USD	0.36 USD
Status	-			
Description:	-			
Config summary	HTTP API requests units (thousands), Average size of each request (34 KB), REST API request units (thousands), Cache memory size (GB) (None), WebSocket message units (thousands), Average message size (32 KB), Requests (10 per month), Messages (per second), Requests (100 per month)			

Name	Group	Region	Upfront cost	Monthly cost
Amazon DynamoDB	-	US East (N. Virginia)	0.00 USD	1.00 USD
Status	-			
Description:	-			
Config summary	Table class (Standard), Average item size (all attributes) (5 KB), Data storage size (4 GB)			

Name	Group	Region	Upfront cost	Monthly cost
Amazon Simple Storage Service (S3)	-	US East (N. Virginia)	0.00 USD	0.14 USD
Status	-			
Description:	-			
Config summary	S3 Standard storage (5 GB per month), PUT, COPY, POST, LIST requests to S3 Standard (1000), GET, SELECT, and all other requests from S3 Standard (1000) S3 INT Average Object Size (16 MB), Percentage of Storage in INT-Frequent Access Tier (1), S3 INT storage (1 GB per month)			

Name	Group	Region	Upfront cost	Monthly cost
AWS X-Ray	-	US East (N. Virginia)	0.00 USD	0.00 USD
Status	-			

Name	Group	Region	Upfront cost	Monthly cost
Amazon Simple Storage Service (S3)	-	US East (N. Virginia)	0.00 USD	0.14 USD

Status -

Description: -

Config summary S3 Standard storage (5 GB per month), PUT, COPY, POST, LIST requests to S3 Standard (1000), GET, SELECT, and all other requests from S3 Standard (1000) S3 INT Average Object Size (16 MB), Percentage of Storage in INT-Frequent Access Tier (1), S3 INT storage (1 GB per month)

Name	Group	Region	Upfront cost	Monthly cost
AWS X-Ray	-	US East (N. Virginia)	0.00 USD	0.00 USD

Status -

Description: -

Config summary Sampling rate (1), Add X-Ray Insights (Yes), Number of requests per month (250), Number of queries per month (250), Traces retrieved per query (10)

Name	Group	Region	Upfront cost	Monthly cost
Amazon CloudWatch	-	US East (N. Virginia)	0.00 USD	1.04 USD

Status -

Description: -

Config summary Number of Metrics (includes detailed and custom metrics) (3), GetMetricData: Number of metrics requested (1000), Number of other API requests (500), Standard Logs: Data Ingested (0.125 GB), Standard Logs Delivered to CloudWatch Logs (0.125 GB), Number of Lambda functions (4), Number of requests per function (5 per month)

Name	Group	Region	Upfront cost	Monthly cost
AWS CloudTrail	-	US East (N. Virginia)	0.00 USD	0.60 USD

Status -

Description: -

Config summary Management events units (100 thousands), Write management trails (1), Read management trails (1), Data events units (100 thousands), S3 trails (1), Lambda trails (1), Insight events units (millions), Trails with Insight events (1), Write management events (500 per month), Read management events (1000 per month), S3 operations (1 per month), Lambda data events (5 per month)

Name	Group	Region	Upfront cost	Monthly cost
AWS CloudFormation	-	US East (N. Virginia)	0.00 USD	4.57 USD

Status -

Description: -

Config summary Number of third-party extensions managed (100), Average duration per operation (30 seconds), Total number of operations per extension (2 per day)

Acknowledgement

AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services. Learn more

1.1.8 PROJECT ACTIVITY SCHEDULES

PHASES	WEEKS	ACTIVITIES
Planning and strucuring	Week 1	1. Definition of project objectives and goal. 2. Research and knowledge acquisition of project. 3. Design of project architecture. 4. Cost Estimation 5. Team review of individual projects.
Implementation	Week 2	1. Re-design architecture to address raised concerns 2. Produce a final architecture to implement. 3. Implement architecture in AWS console. 4. Test and Evaluate implemented design. 5. Monitor and troubleshoot any errors that may arise.

CHAPTER 2: REVIEW OF SIMILAR SYSTEMS

2.1.0 PROCESSES OF THE EXISTING SYSTEM

There have been various development of tools and software that makes job applications easier. For the purposes of the proposed Automated Resume Keyword Extraction Tool, a few of the earlier stated existing solutions such as Resumatch.io and Jobscan have been reviewed with a lot of similarities found across board. However, a few gaps were noticed:

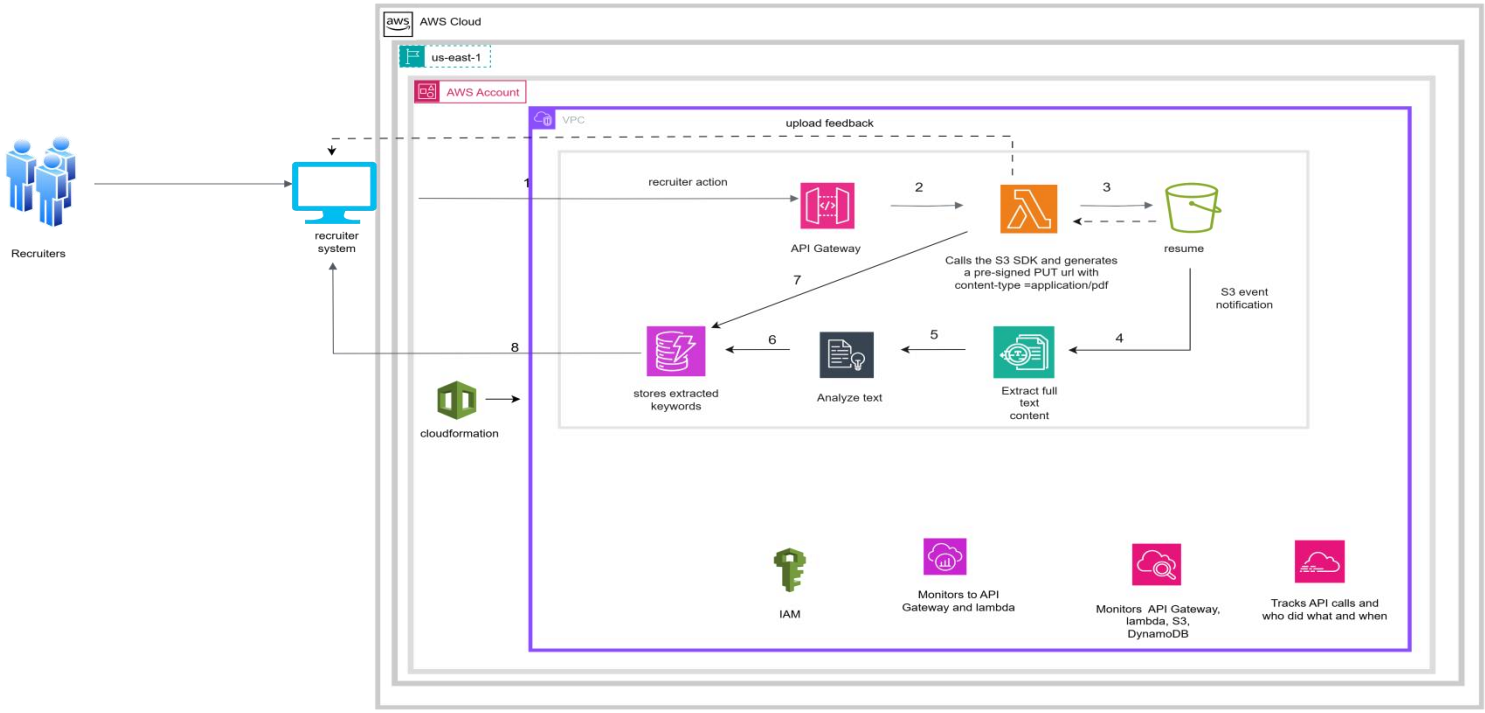
- Main focus is on keyword overlap but doesn't handle synonyms well.
- Inability to deeply analyze context. (eg. "Python" in programming is weighted the same as "Python for data analysis at a certain company").

2.1.1 PROPOSED SOLUTION

A high-performance, serverless, and cost-efficient AWS tool that automates resume ingestion, extracts key qualifications, and makes them searchable, providing recruiters with a scalable, secure solution that reduces time-to-hire, boosts productivity, and ensures long-term cost savings.

- Integration via API Gateway allows recruiters to continue using their existing system.
- S3 with pre-signed URLs ensures secure, PDF-only uploads.
- Lambda functions handles the triggering of textract to extract text content and comprehend to analyze and categorize extracted text. (skills, education, job titles, experience).
- CloudWatch, CloudTrail & X-Ray provide monitoring, logging, and debugging.
- AWS Managed VPC
- DynamoDB stores structured candidate data for fast querying and filtering.
- Cloud formation was used to provision my resources as infrastructure as code (IaC)
- IAM for access control.

2.1.2 ARCHITECTURE OF THE PROPOSED TOOL



2.1.3 COMPONENT DESCRIPTION

This section talks about the AWS services that were selected and the role they will play in terms of making our architecture into a working tool.

- VPC: Private cloud where our services reside in in the AWS cloud.
- Region: Location where resources would be deployed to.
- API Gateway: Allows end users to interact with the tool's back-end.
- Back-end: REST API and a lambda function that handles resume processing.
- Simple Storage Service (S3): Stores uploaded resumes
- Textract: Extracts the full text content of the uploaded resume.
- Comprehend: Processes and analyzes the extracted text to identify key phrases and patterns.
- Dynamo DB: Stores important resume metadata and outputs of the comprehend service.
- Cloud watch and Cloud trail: Offer monitoring capabilities to the tool.
- X-ray: Provides monitoring and troubleshooting capabilities.
- IAM: Determines who gets access to what and their allowed actions.
- Cloudformaion: Used to provision all the necessary resources needed for our tool to execute successfully.

2.1.4 DEVELOPMENT TOOLS AND ENVIRONMENTS

Under this section, the various environments and tools used to architect and implement the tool would be stated and talked about briefly.

Environments

- AWS Cloudshell: An in-built command line interface of the aws cloud that performs similar functions as local computer's terminals or command prompt.
- Visual Studio Code: Used to edit codes.

Tools

- Draw.io : For designing the solution architecture
- Postman: Test API URLs and uploading resumes.
- Github: Code versioning and control

2.1.5 BENEFITS OF IMPLEMENTING THE PROPOSED SYSTEM

With the implementation of the proposed system, Automated Resume Keyword Extraction Tool, below are some merits that would be enjoyed:

- **Consistency in Screening**: Based on the criteria selection used during the implementation of the tool, the tool ensures a standardized evaluation process

- **Improved Candidate Matching**: The tool would be able to automatically identify and extract keywords related to technical skills, education, certifications, and job titles, and them match them against job requirement leading to a more accurate candidate shortlisting and ensures that qualified candidates are not overlooked.

- **Increased Efficiency and Time Savings** Manual resume screening is incredibly time-consuming, especially when dealing with hundreds or thousands of applications. An automated tool can process resumes in seconds, extracting relevant skills, qualifications, and experience instantly. This frees up HR professionals to focus on more strategic tasks like candidate engagement and interviews rather than spending hours reading through resumes.

CHAPTER 3: METHODOLOGY

3.1.0 OVERVIEW

This section sheds light on the requirements to be factored and tasks to carry out to ensure the system functions accurately.

3.1.1 REQUIREMENT SPECIFICATION

Here, the system's functionalities and constraints would be outlined.

Functionalities

- Allowing users to upload resumes only in PDF format.
- Extraction and processing of text content of the uploaded file.
- Analysis and structuring of extracted text.
- Resume data should be stored in a database.
- Allowing users access stored data in the database to assist them screen candidates.

Constraints

- Integration with an already existing HR system.

3.1.2 STAKEHOLDERS OF THE SYSTEM

This section identifies those who are affected directly or indirectly, and may have an impact in the success of this project.

- DevOps Engineer
- End Users
- Service Suppliers

3.1.3 REQUIREMENT GATHERING PROCESS

The processes that were employed in determining and gathering the requirements for the automated resume keyword extraction tool are considered in this sub-section. Here, the requirements of the stakeholders must be understood to be able to produce the system's specification, functional requirements, and non-functional requirements.

- Literature Review: Here, similar tools were analyzed and their gaps were capitalized on to give this project a competitive advantage.

- Requirement Review: Here, per the nature of the system being developed, the gathered requirements were analyzed thoroughly to make sure all necessary requirements were considered.

CHAPTER 4: IMPLEMENTATION AND RESULTS

4.1.0 CHAPTER OVERVIEW

This chapter takes into consideration the results of the implementation of this project, the testing measures and the logical design mapping.

4.1.1 WORKFLOW

- Users log into their system.
- On success, user makes an API call to the tool's backend when "Upload Resume" is clicked/toggled.
- Authentication of JWT access token is done and on-success, a pre-signed S3 URL is returned to the user to facilitate upload.
- User uploads resume using the pre-signed url.
- Lambda processes this action by ensuring the file is in pdf format before uploading it into the S3 bucket.
- On-success, S3 event notification invokes "Textract" to extract the full text content of the uploaded file.
- Textract on completion triggers "Comprehend" which analyzes the extracted text and searches for keywords, phrases, etc.
- Dynamo Db then stores the keywords, phrases and other resume metadata.

4.1.2 TESTING

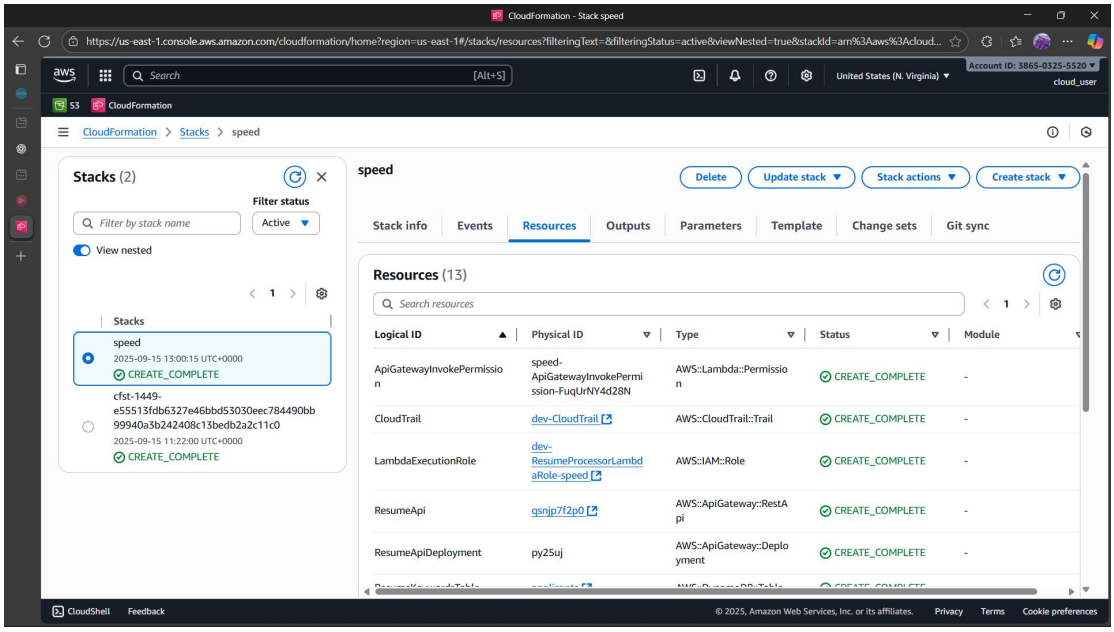


Figure 1: Successful creation of Cloud formation stack

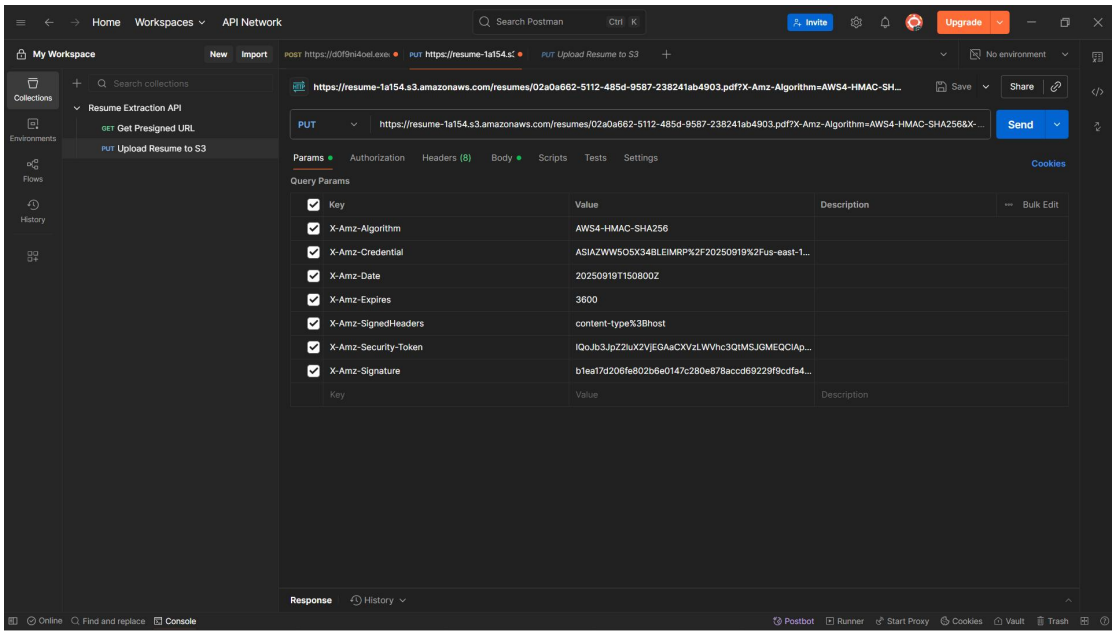


Figure 2: Pre-signed url parameters review on postman

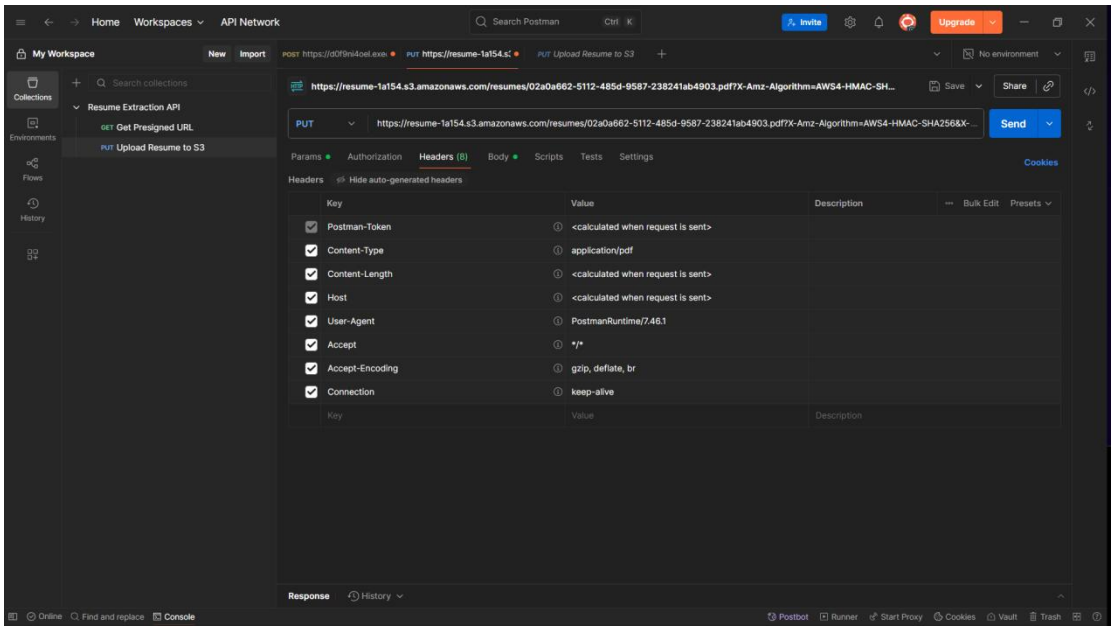


Figure 3: Pre-signed url headers review on postman

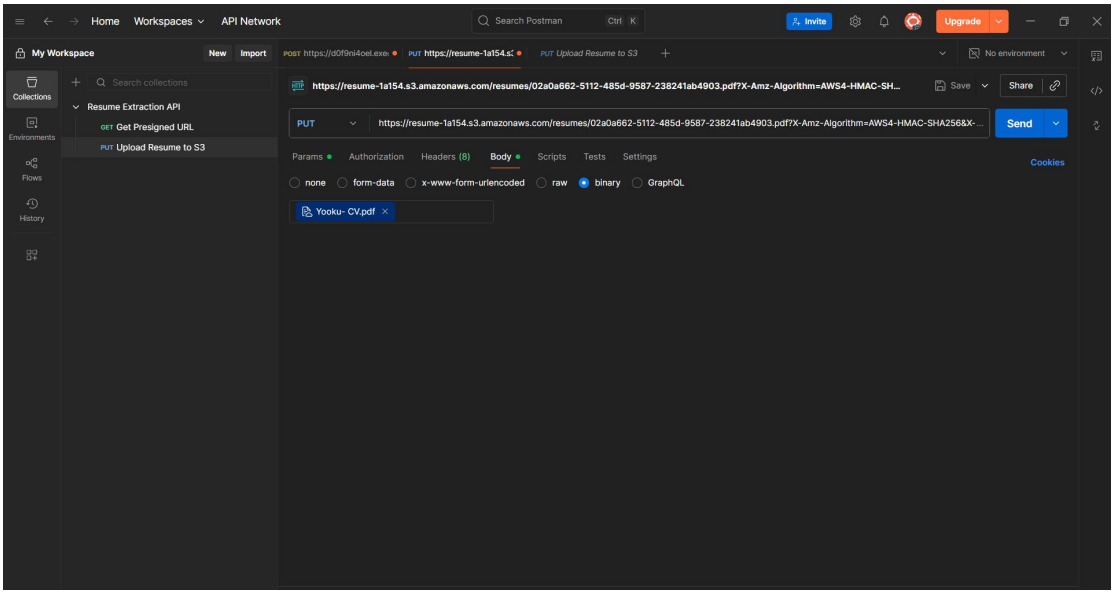


Figure 4: Selected resume file to be uploaded.

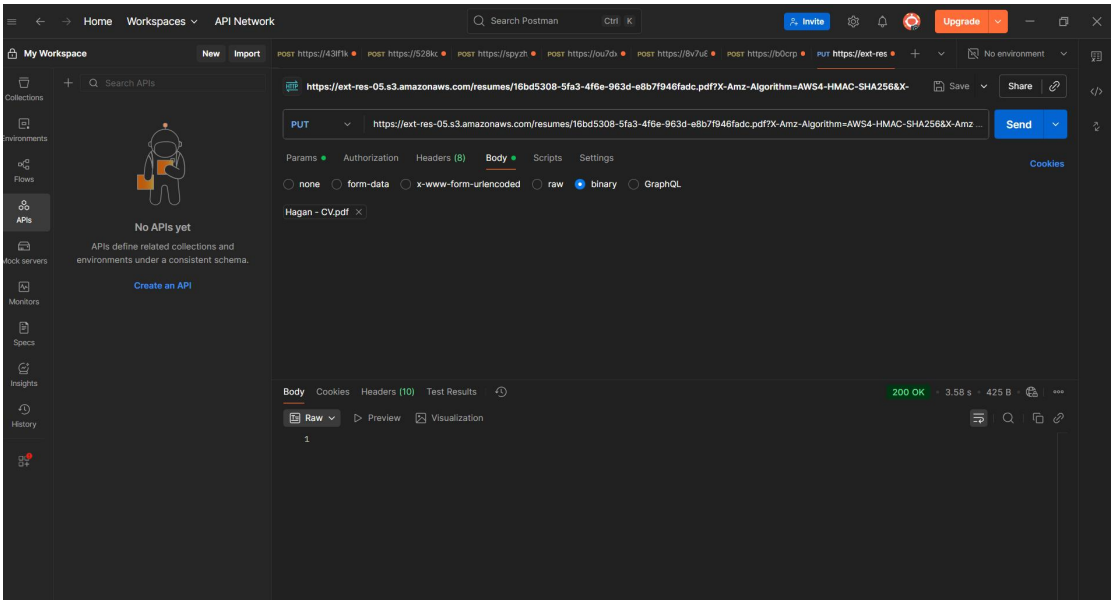


Figure 5: Successful file upload.

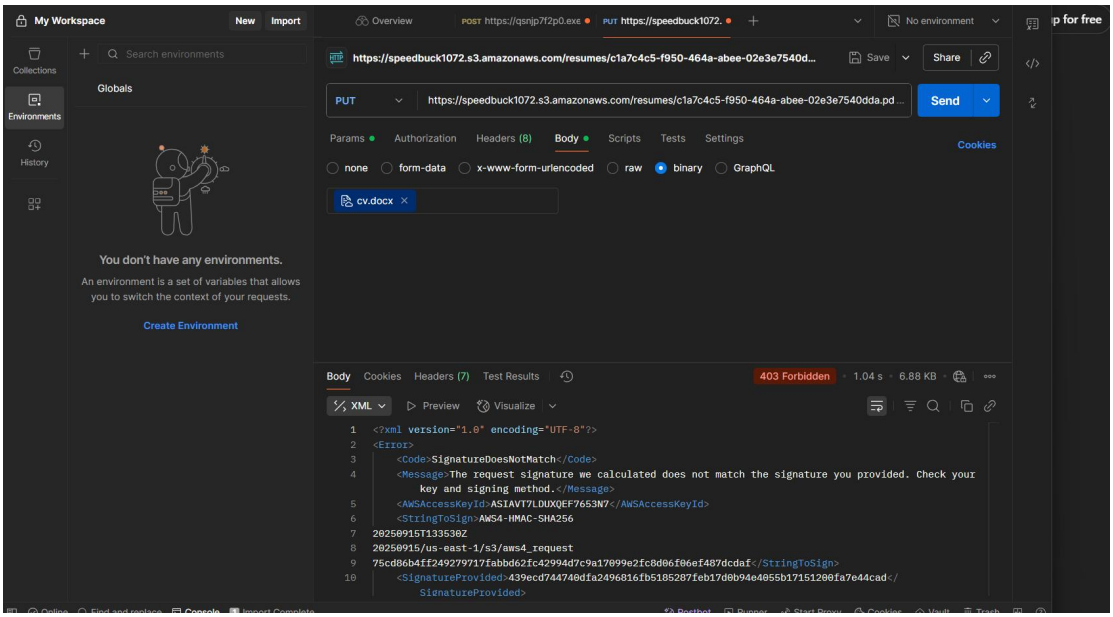


Figure 6: Validation error. Wrong file type.

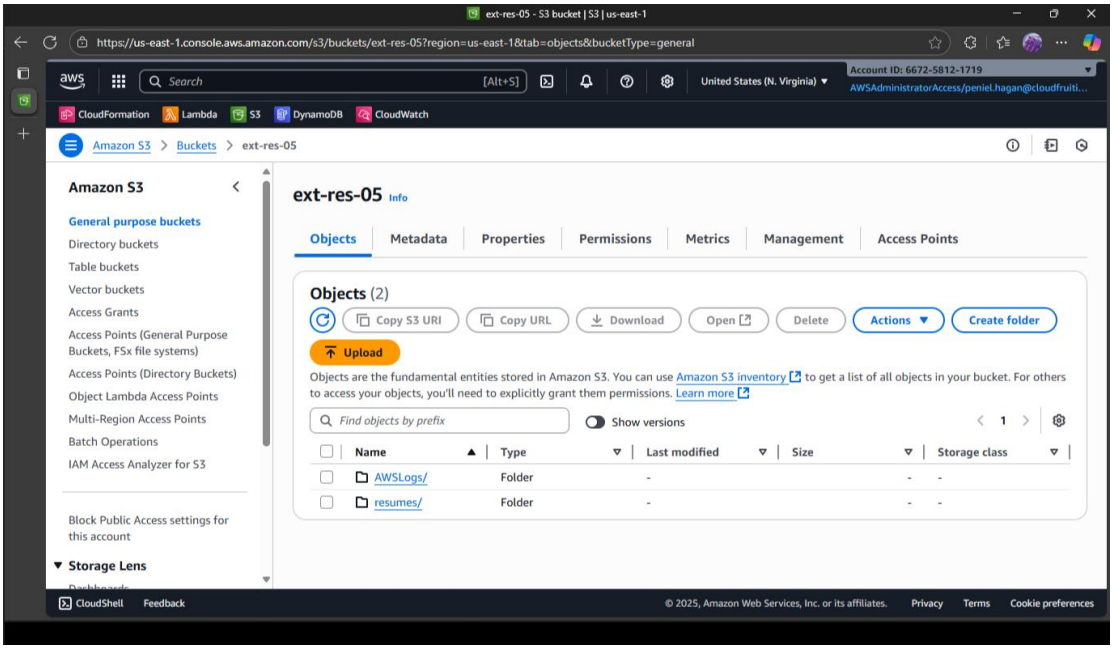


Figure 7: resume uploaded into the object resume/ of the ext-res-05 bucket.

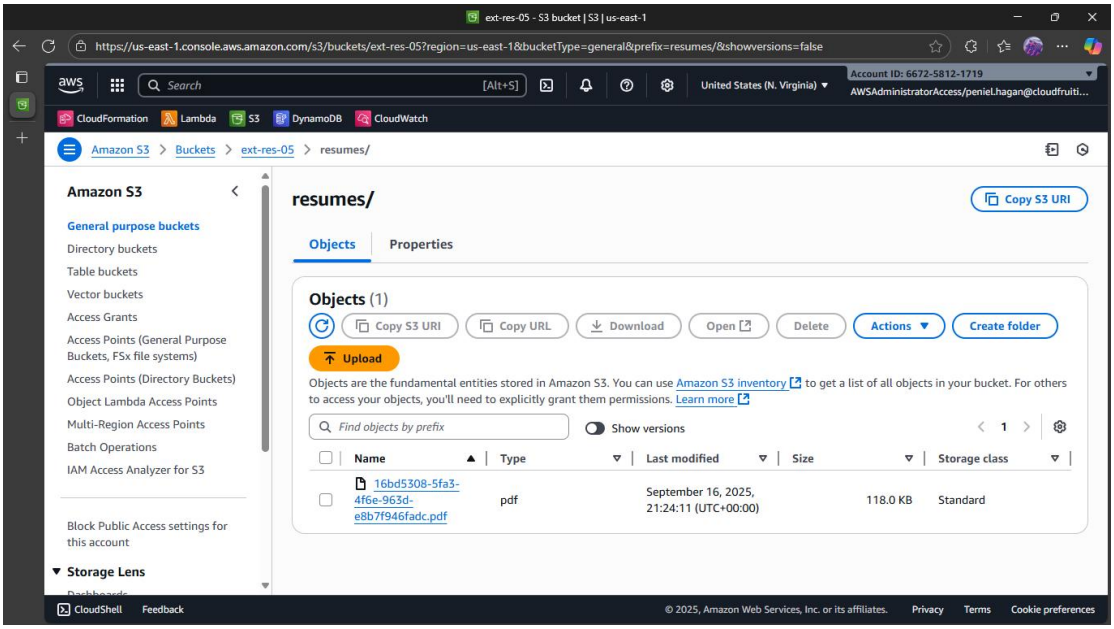


Figure 8: uploaded resume

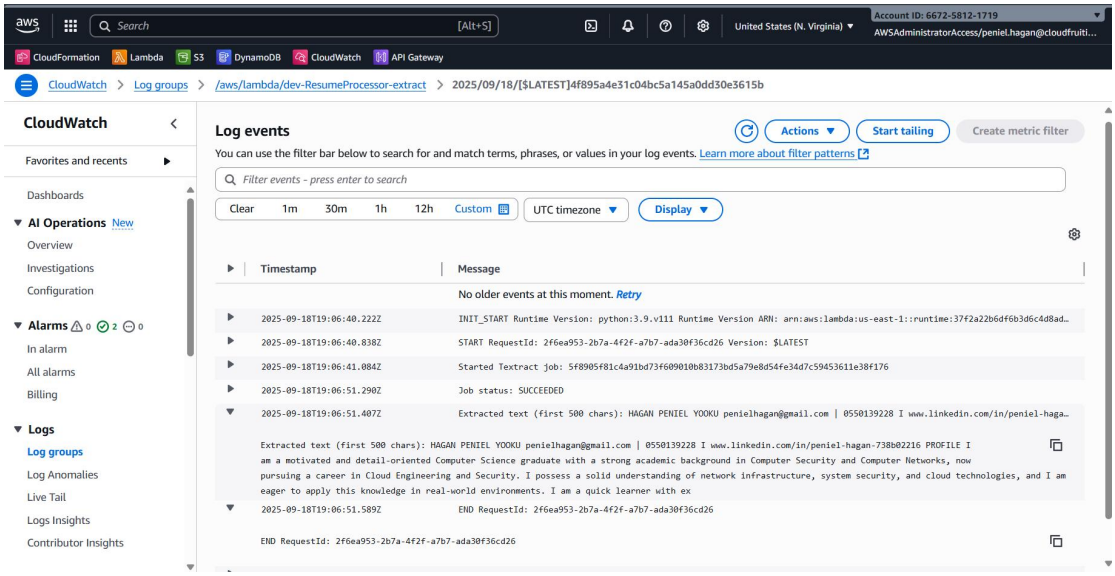


Figure 9: Automated processing of uploaded resume.

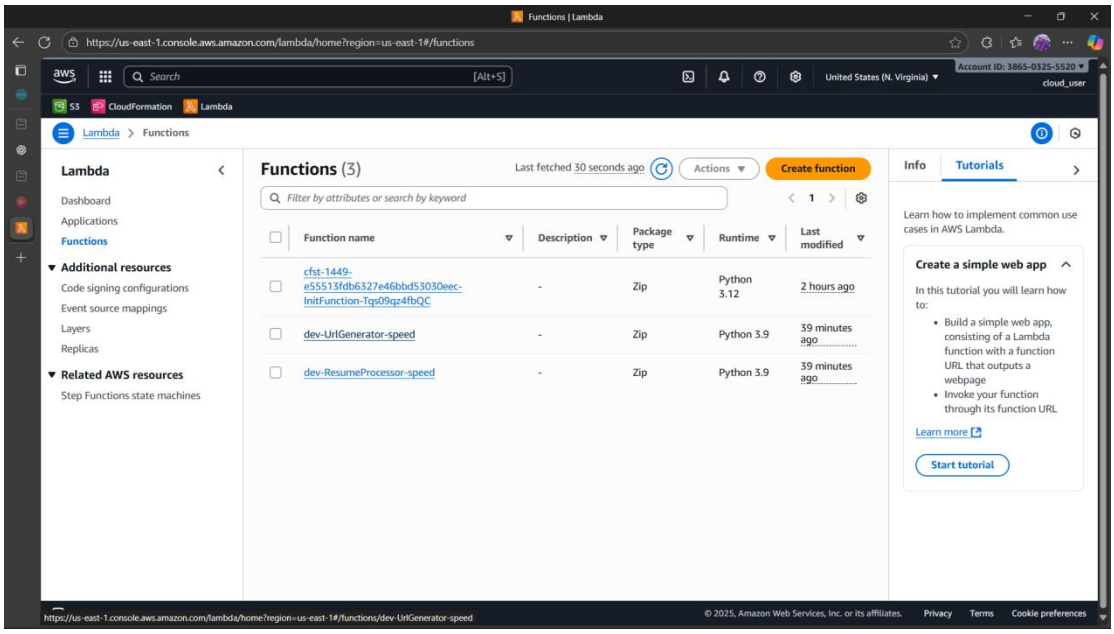


Figure 10: lambda functions.

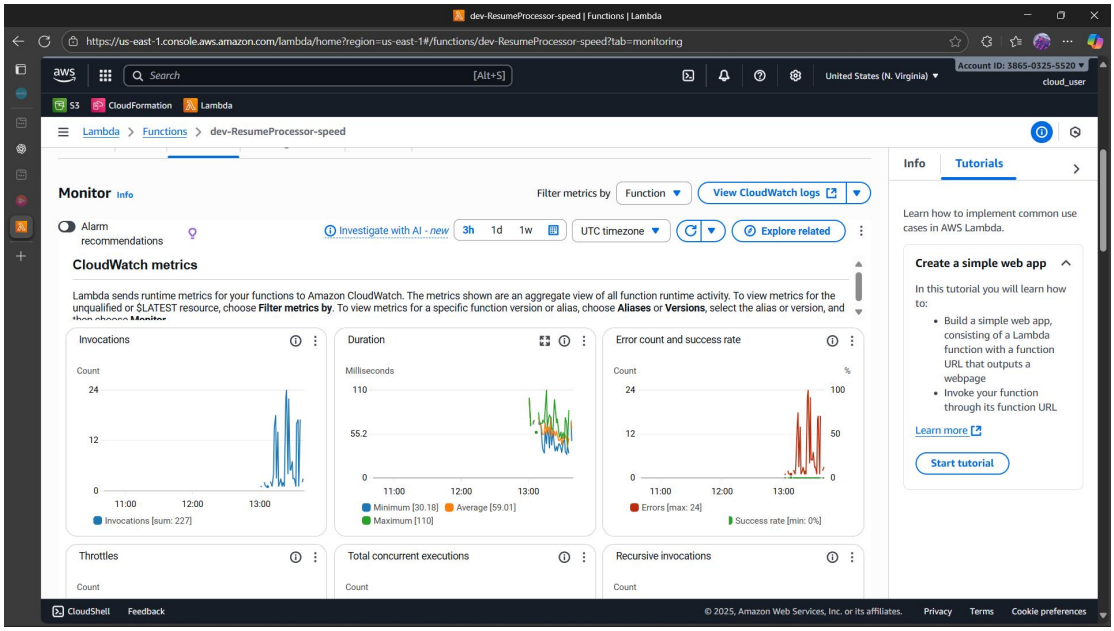


Figure 11: Resume processor lambda function cloudwatch metrics.

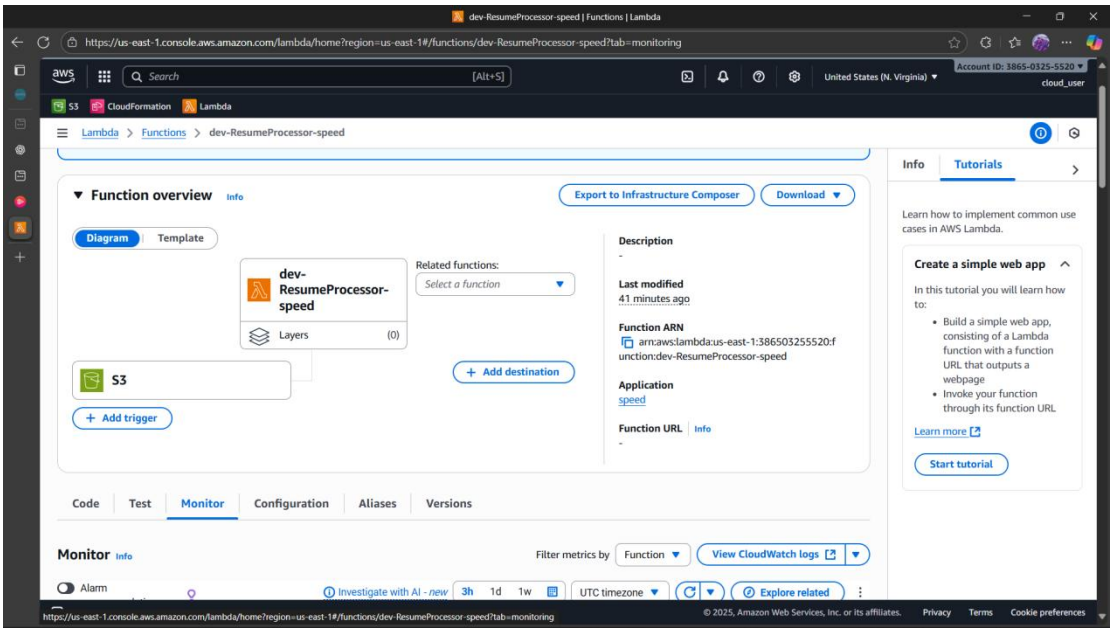


Figure 12: Resume processor lambda function overview.

4.1.3 RESULTS

After running the necessary test, it was observed that;

- The pre-signed URL was generated
- The user was able to select a file to be uploaded.
- The URL was able to validate the content-type of the file before uploading it.
- The file was successfully uploaded into the storage bucket.
- The automated processing was quickly initiated when the upload was successful.
- The end-user was able to see the skills, education level and job titles of applicants.

CHAPTER 5: FINDINGS AND CONCLUSION

5.1.0 CHAPTER OVERVIEW

In this chapter, the findings and conclusions drawn from the development, implementation and results stages of the A.R.K.E tool are highlighted. This chapter also takes into consideration the challenges faced during the design through to the implementation phase of the project, the recommendations for future work and sources of information and guides used in the completion of the project.

5.1.1 FINDINGS

- Diversity: The system was able to process all PDF resume that had different layout formats.
- Accuracy and Efficiency: The system was able to identify correctly which files were not PDF files and discarded them during the upload stage even before it was saved in the storage bucket.
- System Performance: The tool accurately processes resumes and stores them in the database with low latency.

5.1.2 CONCLUSIONS

From the findings, it was strongly seen that the tool successfully carried out it's functionalities through validation of document format type and successful processing of file with low latency thus achieving the goal of automating the resume keyword extraction process.

5.1.3 CHALLENGES/ LIMITATIONS OF THE SYSTEM

Although the system was able to meet it's goal, there were still a few roadblocks

Integration into existing HR Systems: This threw a great challenge as the solution wasn't integrated into an actual HR recruiting system to see how effective it would be in the real world.

5.1.4 LESSON LEARNT

From the idea conception stage to the implementation stage of the project, it was seen that a strong idea,good development process, reliable resources and proper overall

planning of the project was essential for its success and goal attainment. Thorough requirement analysis helped in the identification and prioritization of what the system will and will not do.

The choice of tools,environments and frameworks can influence the project timeline and project outcomes. A fitting solution aligns with project requirements and ensures scalability and future maintainability.

5.1.5 RECOMMENDATION FOR FUTURE WORK

The following are what I would recommend to be done in future works.

- Train a comprehend model with large numbers of test resumes to be able to analyze and categorize resume data perfectly and in an organization standard format.

- Integrate with a real world HR system and measure it's accuracy and efficiency.

5.1.6 REFERENCES

1. Amazon Web Services. (n.d.). Amazon Textract documentation. https://docs.aws.amazon.com/textract/
2. Amazon Web Services. (n.d.). Troubleshooting CloudFormation. In AWS CloudFormation user guide. https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/troubleshooting.html
3. AWS re:Post. (n.d.). Amazon Textract file upload error. https://repost.aws/questions/QUEy4W8VsfRSa7J9gh0FNlPQ/amazon-textract-file-upload-error