Enhancing Operational Effectiveness and Streamlining Performance of Verification Platform/Dashboard Using Java and Python

Internship report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By: Hritik Rai

191210

Under the supervision of

Dr. Amit Kumar

Assistant Professor (Senior Grade)

Department of Computer Science & Engineering And

Information Technology

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT, SOLAN,

HIMACHAL PRADESH – 173234

CERTIFICATE

This is to certify that the work which is being presented in the internship report titled **“Enhancing Operational Effectiveness and Streamlining Performance of Verification Platform/Dashboard Using Java and Python”** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat, is an authentic record of work carried out by **Hritik Rai** during the said period; February 2023 – till date, ensuring proper care towards the rules and regulations as specified by the Non-Disclosure Agreement signed between Hritik Rai and Tartan.

Hritik Rai
191210
Jaypee University of Information Technology Waknaghat, Solan, H.P.

The above statement made is correct to the best of our knowledge.

Niyati Rabadiya (Manager)

TartanHQ

Dr. Amit Kumar (Assistant Professor)
Jaypee University of Information Technology Waknaghat, Solan, H.P

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: __________________________ __**Department:** ________________ **Enrolment No** ________

Contact No. ____________________________**E-mail.** ____________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ________________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

This is a matter of pleasure for me to acknowledge my deep sense of gratitude to my college, Jaypee University of Information Technology, for giving me an opportunity to explore my abilities via this internship program. I would like to express my sincere gratitude to our Training and Placement officer, Mr. Pankaj Kumar, and our faculty Coordinator, Dr. Nafis U Khan, for this opportunity. I also wish to express my gratitude to my internship supervisors for their valuable guidance and advice towards my internship.

I would like to record my sincere appreciation and gratitude towards all the officials, coaches, trainers, mentors, and employees of Tartan; without whose kind assistance, my internship program would not have been proceeding in a swift direction. The facts and other vital information provided by them have contributed to making this report as comprehensive as possible. I am indeed thankful to them.

Last but not least, I would like to express my sincere thanks to all my family members, friends, and well-wishers for their immense support and best wishes throughout the internship duration and the preparation of this report, and I wish they would continue to contribute towards my well-being.

I believe that this report will be a valuable asset not only for academic institutions but will also be useful for all those who are interested to learn about internship experiences in auditing and consulting firms.

Hritik Rai

191210

Jaypee University of Information Technology,

Waknaghat, Solan, H.P

Table of Contents

Content	**Page no.**
Certificate	i
Plagiarism Certificate	ii
Acknowledgment	iv
Table of Figures	v
Abstract	vi
Chapter-1 INTRODUCTION	1-11
Chapter-2 TOOLS AND TECHNOLOGIES	12-30
Chapter-3 SYSTEM DEVELOPMENT	31-42
Chapter-4 PERFORMANCE ANALYSIS	43-47
Chapter-5 CONCLUSION	48
REFERENCES	49

LIST OF FIGURES

Figure Name
Figure 1: What Tartan does
Figure 2: Comparison of Tartan and its competitors
Figure 3: What is Springboot
Figure 4: Using python for automation
Figure 5: Python Lambda function to manage 22lac data entries
Figure 6: Overview of AWS Lambda
Figure 7: Python Lambda function to manage 22lac data entries
Figure 8: Cloud Object storage
Figure 9: AWS Glue Crawler
Figure 10: Dashboard overview
Figure 11: Sending an invitation to the end user
Figure 12: Prompt that is displayed after an invite has been sent.
Figure 13: Email received by customer
Figure 14: Verification dashboard's landing page
Figure 15: Organization name
Figure 16: Login using payslip/payroll
Figure 17: File upload
Figure 18: Prompt shown after the completion of the entire process
Figure 19: Retrieving Cibil Data
Figure 20: SSL tunnel
Figure 21: Third-party integration of APIs
Figure 22: SQS using Docker
Figure 23: SQS using Java

ABSTRACT

The operational effectiveness and performance of a verification platform or dashboard play a crucial role in ensuring accurate and efficient data validation processes. This abstract presents an approach to enhance the operational effectiveness and streamline the performance of a verification platform using Java and Python.

In the backend, the Java programming language is utilized to develop a scalable and robust architecture for the verification platform. It provides a foundation for handling concurrent requests, managing data storage and retrieval, and ensuring secure communication. Java frameworks such as Spring Boot can be employed to expedite development, integrate with databases, and enable efficient testing. Python is employed to enhance the platform's performance by writing scripts that enhance the speeding time. These functionalities can be leveraged to preprocess and validate data, detect anomalies or patterns, and automate decision-making processes within the verification platform.

The integration of Java and Python is achieved through well-defined APIs and communication channels. Java-based services expose endpoints that allow Python scripts to interact with the backend infrastructure. This enables seamless data exchange, integration of machine learning models, and real-time feedback to enhance the verification process. Along with that various AWS services like AWS EC2, AWS S3, etc were used

Overall, this approach combining Java and Python provides a comprehensive solution for enhancing the operational effectiveness and streamlining the performance of a verification platform. By leveraging the strengths of both languages, the platform can efficiently handle data validation tasks, leverage machine learning capabilities, and adapt to evolving verification requirements.

Chapter 01: INTRODUCTION

1.1 Introduction

About TartanHQ

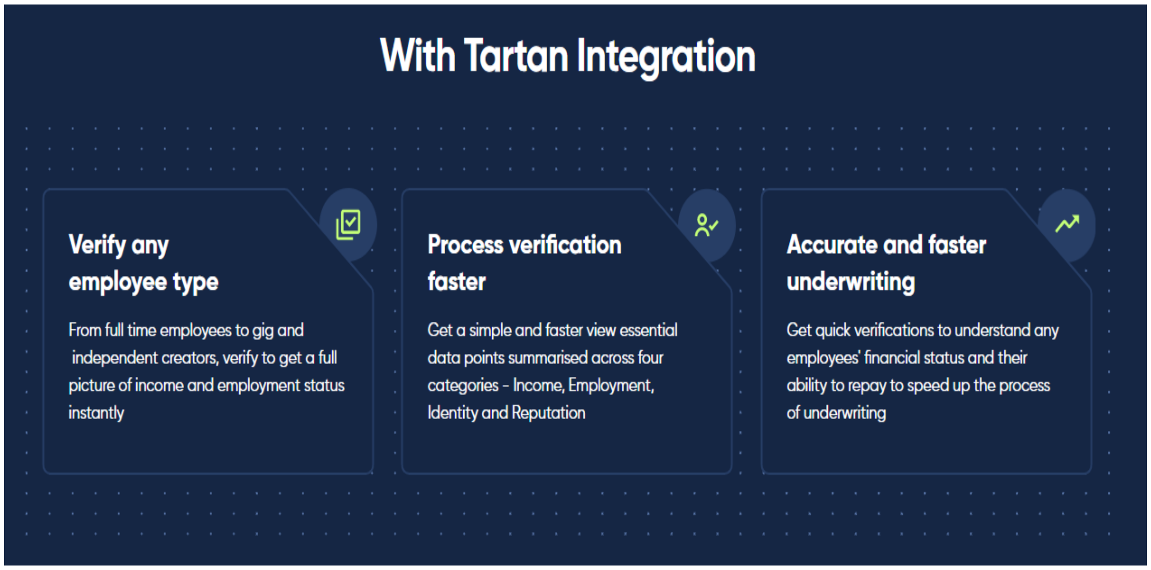
TartanHQ is a creative association that represents considerable authority in giving thorough innovation and business task arrangements. They streamline verifications for your customers with easy, real-time data connectivity to their payroll or HRMS through a single API. Their essential objective is to effectively engage organizations with powerful computerized change systems, empowering them to explore the intricacies of the advanced scene.

At TartanHQ, they offer a different scope of administrations customized to meet the particular necessities of organizations. They succeed in conveying strong programming arrangements and improvement administrations, working intimately with clients to grasp their exceptional necessities. Their mastery stretches out to custom programming advancement, as well as web and versatile application improvement. Furthermore, TartanHQ succeeds in planning and executing effective information the executives frameworks. They perceive the meaning of information-driven navigation and help organizations in putting together and dissecting their information successfully. By utilizing state-of-the-art innovations and carrying out information-driven systems, they empower organizations to determine important experiences that drive development and work on general effectiveness.

Additionally, TartanHQ gives exhaustive counseling administrations pointed toward advancing business processes and upgrading functional productivity. Through a close joint effort with clients, they recognize regions for development, smooth out work processes, and carry out prescribed procedures. Their industry-explicit bits of knowledge and suggestions increase the value of their counseling ability.

Past their specialized proficiencies, TartanHQ puts areas of strength for encouraging a culture of development inside associations. They offer preparation and studios to assist groups with creating abilities in arising advances, for example, man-made brainpower, AI, and blockchain. By outfitting organizations with the most recent devices and information, TartanHQ supports the development and positions them ahead in a quickly developing computerized scene. Tartan HQ values conveying excellent client encounters. They focus on areas of strength for building with their clients, working cooperatively to guarantee their fulfillment. Through customized help and mindful client assistance, TartanHQ lays out trust and unwavering quality, prompting long-haul organizations.

Moreover, TartanHQ stays committed to remaining at the front line of innovative headways. They put resources into continuous innovative work to remain refreshed with the most recent patterns and arising advances. This responsibility empowers them to offer inventive arrangements that line up with the advancing necessities of their clients and the business. TartanHQ likewise puts significance on manageability and social obligation. They effectively look for eco-accommodating practices in their activities and elevate naturally cognizant ways to deal with innovation and business processes. They endeavor to limit their carbon impression and make positive commitments to the networks they serve. TartanHQ encourages a culture of cooperation, variety, and constant learning. They draw in top abilities from different foundations and give a strong climate for proficient development and improvement. By developing a culture of inclusivity and development, TartanHQ tackles the aggregate skill and imagination of its colleagues to convey remarkable outcomes. In rundown, TartanHQ is a groundbreaking association that joins specialized skill, business discernment, development, and a promise to consumer loyalty. Through their scope of administrations, including programming improvement, informing the executives, counseling, and cultivating development, they engage organizations to flourish in the advanced age. With an emphasis on manageability, social obligation, and cooperative inside culture, TartanHQ stands apart as a dependable and groundbreaking accomplice for associations looking for exhaustive arrangements and groundbreaking development.



With Tartan Integration

Verify any employee type

From full time employees to gig and independent creators, verify to get a full picture of income and employment status instantly

Process verification faster

Get a simple and faster view essential data points summarised across four categories - Income, Employment, Identity and Reputation

Accurate and faster underwriting

Get quick verifications to understand any employees' financial status and their ability to repay to speed up the process of underwriting

Figure 1: What Tartan does [1]

1.2 Description of Industry

FinTech (Financial technology)

The term FinTech is made up of two words, Finance and Technology; it is basically the aggregation of the two fields where the technology is used to innovate and improve newer domains in the world of finance. From the ability for consumers or users to view and make financial transactions online to applications that allows them to make the financial transaction within seconds to anyone across the globe to means that enable financial institutions to make rapid financial transactions in huge volume and velocity, everything is the subdomain of innovation in finance aided with technology. FinTech's closest examples in our day-to-day life would be the implementation of apps like Paytm, GPay, etc., or the investment apps like Grow and Zerodha.

One of the biggest opportunities provided by the advent of the fintech in the current market space is in terms of aiding the common man with financial literacy as well as financial accessibility, it has brought more and more control to people's lives in terms of their financial decisions and investments, it has opened the investment opportunities which were earlier beyond the reach of the common man.

FinTech's growth is primarily due to the opportunity for smaller players to compete in the same space as traditional banks and financial institutions. Thanks to FinTech, it's not about who is the biggest but who is the fastest and most responsive to effectively respond to the ever-changing consumer demands. Moreover, the solutions offered by fintech companies are no longer "universal." Instead, it provides targeted (often niche) services that fill gaps in specific financial needs. It may also be offered at a much lower cost than the services offered by traditional financial providers.

1.3 Description of Verification Dashboard

Verification Dashboard

The verification dashboard given by your association fills in as a stage for your clients, like HDFC and ABFL, to work with the verification cycle for their clients. This dashboard permits your's clients to go through different sorts of checks effectively, including Know Your Client (KYC), email verification, and pay verification.

KYC Verification: KYC verification is a fundamental cycle that banks and monetary establishments should stick to check the character and validity of their clients. Through your verification dashboard, clients can give the essential individual data and reports expected for KYC consistency. This might incorporate submitting ID records like visas or driver's licenses, evidence of address, and other applicable archives. The verification dashboard smoothes out this interaction by safely gathering and checking the data given by clients.

Email Verification: Email verification is urgent for guaranteeing the exactness and legitimacy of client contact data. By coordinating email check usefulness into your dashboard, clients can confirm their email tends by following a check connection or entering a remarkable code shipped off their email. This step assists your clients with keeping up with precise client records and works with compelling correspondence channels.

Pay Verification: Pay verification is much of the time expected by monetary establishments to evaluate a client's monetary security and financial soundness. Through your verification dashboard, clients can safely give data connected with their pay, for example, work subtleties, compensation explanations, or expense reports. This data is then handled and checked to approve the client's pay claims. This assists your clients with settling on informed choices in regard to advance applications, credit appraisals, or monetary administrations.

In general, your verification dashboard goes about as a focal center for clients of your client associations to go through check processes helpfully and safely. It empowers clients to give essential data and documentation to KYC consistency, confirms email locations to keep up with precise correspondence channels, and works with paycheck-to-survey monetary validity. By smoothing out these cycles, your verification dashboard improves functional productivity and assists your clients with pursuing very much educated choices in light of checked client data.

Moreover, the verification dashboard guarantees a consistent encounter for both your clients and their clients by offering an easy-to-understand interface and instinctive route. It is intended to give a smooth and effective verification process, limiting the time and exertion expected from clients.

Security is a foremost part of the verification dashboard. It integrates vigorous safety efforts to safeguard delicate client information and keep up with consistence with pertinent information protection guidelines. This incorporates encryption conventions, secure information stockpiling, and access controls to defend the respectability and secrecy of client data.

The verification dashboard likewise offers far reaching detailing and investigation capacities. It gives constant bits of knowledge and measurements to your clients, permitting them to screen the advancement of verifications, recognize any bottlenecks, and gain important information-driven experiences into their client base. These experiences can help with working on functional proficiency, advancing dynamic cycles, and improving client care.

Also, the verification dashboard can be redone and custom fitted to meet the particular marking and prerequisites of your clients. It very well may be consistently coordinated with their current frameworks and work processes, guaranteeing a consistent and firm insight for the two clients and clients.

How it works

1. Verification

The user shares their employment records with their consent to verify their salary

2. Employment Type

The user selects their employment type and their employer. We support Fortune 500 employers in India

3. HRMS Login

User connects to the employment database by entering their username and password

4. Seamless Integration

The user can now seamlessly and securely connect to their HRMS system within 30 seconds

The advantages of the verification dashboard reach out past the verification cycle itself. By smoothing out and mechanizing the verification strategies, it diminishes manual exertion and administrative work for both your clients and their clients. This recoveries time as well as upgrades consumer loyalty by giving a helpful and proficient verification experience.

By and large, the confirmation dashboard fills in as an integral asset for your clients, like HDFC and ABFL, to smooth out their verification processes and work on functional effectiveness. It offers a protected, easy-to-use stage for clients to go through KYC, email, and pay confirmations. By working with these verifications, the dashboard assists your clients with keeping up with consistency, pursuing informed choices, and giving a consistent client experience.

Working of Verification Dashboard

As a component of our administration offering, we give outer client APIs that permit a consistent mix of our administrations into your framework. These APIs empower you to communicate with our administrations automatically, computerizing different cycles and improving the general usefulness of your framework.

For instance, if you need to use the sync administration, which includes scratching important pieces of information from verified finance dashboards of end-client clients, you can make Programming interface solicitations to flip the sync administration on or off through the setup page on our dashboard. Along these lines, you have some control over the synchronization cycle from your own framework without the requirement for manual intercession.

Also, in the event that you wish to use the picture administration, which uses computer based intelligence to extricate data of interest from payslips in PDF or picture design, you can utilize the pertinent APIs to empower or cripple the representation highlight. Contingent upon your prerequisites, you can pick either quick track and non-quick track modes to improve the handling rate or exactness of information extraction.

Moreover, we give webhooks for of sending reactions to your applications being handled. These webhooks contain checked data and information focuses separated from the payslips or some other pertinent sources. By incorporating these webhooks into your framework, you can get constant updates on the situation with your applications and access the separated data flawlessly.

In outline, our outer client APIs permit you to control and use our administrations automatically, coordinating them into your framework such that best suits your necessities. This degree of coordination enables you to mechanize processes, upgrade usefulness, and get ongoing updates from our administrations, giving a consistent encounter to both you and your end-client clients.

Here are a few extra insights concerning our outer client APIs and how they can be incorporated into your framework:

API Documentation: We give extensive documentation to our APIs, including endpoints, demand/reaction designs, verification strategies, and model code pieces. This documentation fills in as a source of perspective aide, assisting you with understanding how to really collaborate with our administrations.

Authentication and Security: Our APIs utilize industry-standard validation components, for example, Programming interface keys, OAuth, or other secure techniques, to guarantee that main approved frameworks and clients can get to and use the administrations. This safeguards your information and keep up with the honesty of your framework.

Data Integration: Our APIs empower consistent information coordination between your framework and our administrations. For instance, if you need to match up information from verified finance dashboards, you can utilize the applicable Programming interface endpoints to recover and handle the information inside your framework. Essentially, for the imagine administration, you can submit payslips through Programming interface calls to separate data of interest and integrate them into your work processes.

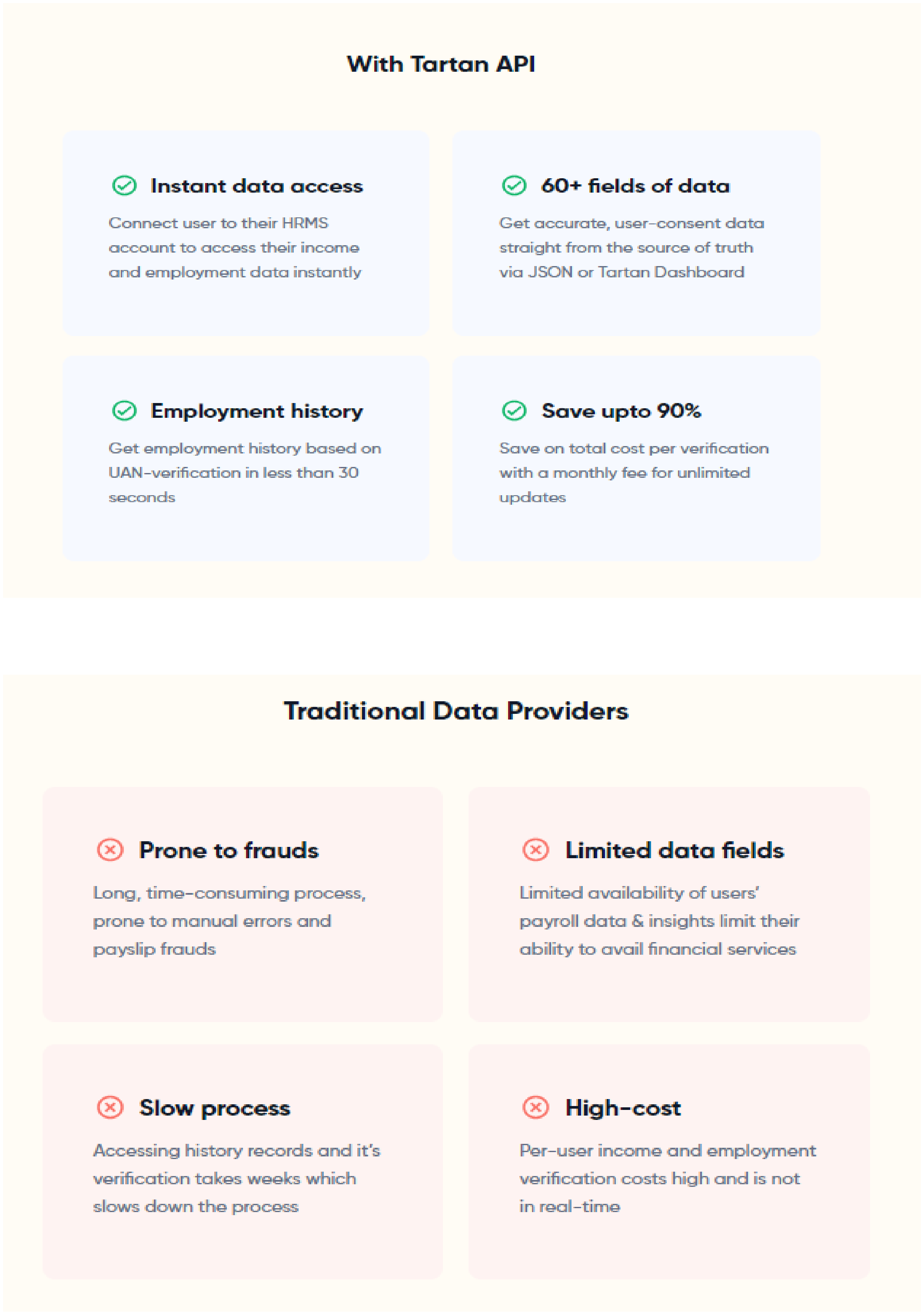
Customization and Flexibility: Our APIs are intended to be adaptable, permitting you to fit the mix to your particular necessities. You can pick which administrations to empower or debilitate, set design choices, and control the progression of information between your framework and our own. This adaptability empowers you to adjust our administrations to your special business necessities.

Error Handling and Monitoring: Our APIs give point-by-point mistakes taking care of components and status codes, permitting you to smoothly deal with special cases and blunders. Furthermore, we offer checking and logging capacities to follow Programming interface utilization, recognize likely issues, and guarantee smooth activities.

Backing and Designer People group: We offer help channels, like documentation, FAQs, and engineer discussions, to help you during the incorporation cycle. Our devoted help group is accessible to address any specialized questions or concerns you might have, guaranteeing a smooth coordination experience.

By utilizing our outside client APIs, you can open the maximum capacity of our administrations and flawlessly incorporate them into your framework. This joining engages you to mechanize processes, improve proficiency, and convey a more thorough answer for your end-client clients.

Here is how Tartan is different from the rest of its competitors in the use of API.



With Tartan API

✓ **Instant data access**
Connect user to their HRMS account to access their income and employment data instantly

✓ **60+ fields of data**
Get accurate, user-consent data straight from the source of truth via JSON or Tartan Dashboard

✓ **Employment history**
Get employment history based on UAN-verification in less than 30 seconds

✓ **Save upto 90%**
Save on total cost per verification with a monthly fee for unlimited updates

Traditional Data Providers

✗ **Prone to frauds**
Long, time-consuming process, prone to manual errors and payslip frauds

✗ **Limited data fields**
Limited availability of users' payroll data & insights limit their ability to avail financial services

✗ **Slow process**
Accessing history records and it's verification takes weeks which slows down the process

✗ **High-cost**
Per-user income and employment verification costs high and is not in real-time

Figure 2: Comparison of Tartan and its competitors [1]

Chapter 02: Tools and Technologies

2.1 Overview

The internship at Tartan HQ involved working in the capacity of Full Stack Engineer, learning skills in languages and tools like Java-based frameworks like Spring Boot, and Log4j2 alongside cloud-based technologies like AWS Lambda, S3, EC2, ETL job, AWS crawler, AWS dynamo db, etc. Some tasks were also defined in Python too.

Apart from all this, this internship also had a focus on soft skills and team management tools, the agile methodology to work, and with a daily scrum to align the project requirements and functionalities to tasks across teams. The constant iterations of the code reviews and code fixes were essential in building a better knowledge base of the said technology alongside of getting a grasp of good coding practices in line with the set practices of the code already established in the company's codebase.
The following technologies were used in this project:

Java

Java is a powerful and vast general-purpose programming language Java which is primarily used in desktop and mobile application development, big data processing, embedded systems, and many other fields.
Java is one of the languages that support OOPS(Object Oriented Programming Systems) and describes everything in terms of classes and objects hence it is highly viable in building real-world systems. Java has a wide variety of frameworks to choose from while building powerful enterprise-level code and scalable systems.
Major Applications include:

- Mobile Application Development,
- Desktop Application Development,

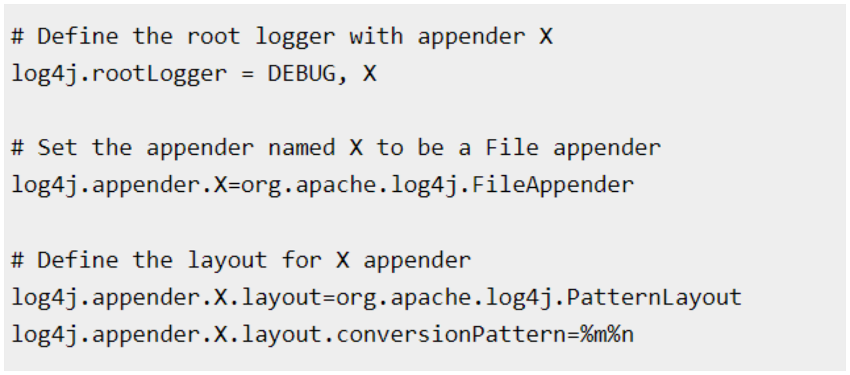
- Web Apps
- Web Servers
- Games, etc.

Java gains its popularity and wide applications due to the following major reasons why java is used:

- It is based majorly on OOPS(Object Oriented Programming Systems) paradigm which makes it easy to use and conceptualize.
- It has a great community support
- Supports concepts like garabage collection etc.

Log4j

Logging is an essential feature of any given project, The purpose of the log is to act as a warning sign when something bad happens. Regularly reviewing the logs will help detect malicious attacks on your system. Given the large amount of log data generated by the system, it is not practical to manually inspect all these logs daily. Java supports logging using log4j2. The logging can be done on multiple platforms or the configurations to append logs on both local file as well as system console, in log4j we can configure what severity of issue we want to log be it error, warning and other issues etc.

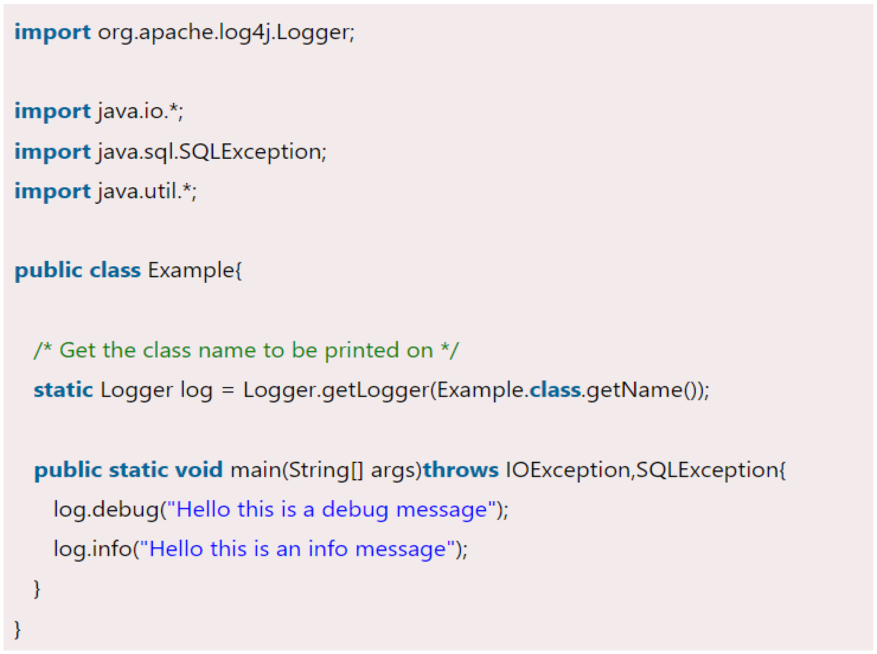


```
# Define the root logger with appender X
log4j.rootLogger = DEBUG, X

# Set the appender named X to be a File appender
log4j.appender.X=org.apache.log4j.FileAppender

# Define the layout for X appender
log4j.appender.X.layout=org.apache.log4j.PatternLayout
log4j.appender.X.layout.conversionPattern=%m%n
```

Fig 3 : Log4j Configuration [2]



```
import org.apache.log4j.Logger;

import java.io.*;
import java.sql.SQLException;
import java.util.*;

public class Example{

  /* Get the class name to be printed on */
  static Logger log = Logger.getLogger(Example.class.getName());

  public static void main(String[] args)throws IOException,SQLException{
    log.debug("Hello this is a debug message");
    log.info("Hello this is an info message");
  }
}
```

Fig 4 : Log4j Example[2]

Springboot

Spring Boot is a generally embraced system for fostering the backend of a confirmation dashboard. It offers a few benefits that go with it a well known decision among designers.

One of the critical advantages of utilizing Spring Boot is its capacity to speed up advancement. With its smoothed out and obstinate methodology, it wipes out standard code and gives reasonable defaults, empowering engineers to quickly assemble backend applications. This productivity is additionally improved by the broad Spring biological system, which offers a large number of highlights and libraries for big business application advancement. From reliance infusion to information access, security, and web administrations, Spring offers strong help for different parts of backend advancement.

For a check dashboard based on a microservices engineering, It is an optimal decision to Spring Boot. It gives worked in elements like help enlistment and revelation (utilizing Spring Cloud Netflix Aha) and unified design the board (utilizing Spring Cloud Config). These elements work with the production of autonomous and versatile administrations, guaranteeing consistent reconciliation and adaptability inside the engineering.

Design in Spring Boot is direct, on account of its show over-arrangement approach. It use auto-design, which naturally arranges parts in view of classpath conditions and reasonable defaults. Also, externalized setup takes into account adaptability and simple customization, as properties or YAML records can be utilized to adjust settings.

Spring Boot accompanies an installed server, dispensing with the requirement for a different application server during sending. This element works on sending and further develops conveyability. Also, Spring Boot offers strong testing capacities, incorporating great with famous structures like JUnit. Designers can compose exhaustive experiments, including unit tests, mix tests, and start to finish tests, guaranteeing the dependability and rightness of their code.

Security is a basic part of a check dashboard, and Spring Boot succeeds around here. It coordinates flawlessly with Spring Security, empowering engineers to effectively carry out validation and approval components. Support for different confirmation techniques, job based admittance control, and secure correspondence guarantees that delicate information and usefulness are safeguarded.

Coordination with information bases is one more strength of Spring Boot. It offers consistent coordination with famous information bases through Spring Information JPA or other ORM systems. This coordination improves on information base tasks, like Muck activities, exchange the executives, and inquiry age, diminishing how much standard code.

Checking and measurements are significant for backend applications. Spring Boot coordinates well with observing and measurements systems like Micrometer and Spring Boot Actuator, permitting designers to accumulate application measurements, screen endpoints, and gain significant bits of knowledge into execution.

Spring Boot benefits from an enormous and dynamic local area, which guarantees continuous help and updates. Normal deliveries, bug fixes, and new elements stay up with the latest, dependable, and all around kept up with.

Ultimately, Spring Boot consistently incorporates with frontend advances, making it viable with well known structures like Rakish, Respond, and Vue.js. This similarity works with smooth correspondence and information trade between the backend and frontend parts of the confirmation dashboard.

In outline, Spring Boot offers a complete and highlight rich system for backend improvement. Its productivity, broad biological system, versatility, security highlights, testing capacities, data set mix, observing help, local area driven refreshes, and frontend similarity pursue it an ideal decision for fostering a strong and effective backend for a check dashboard.

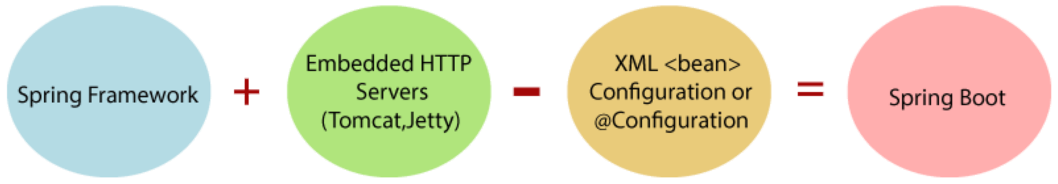
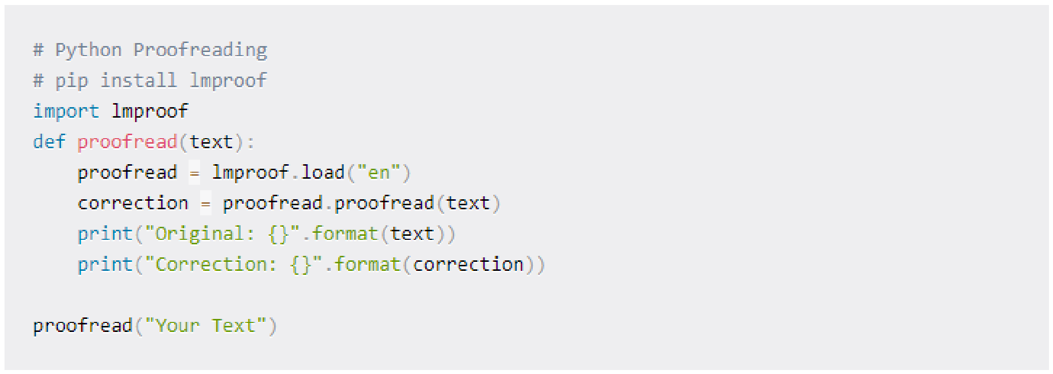

Spring Framework
+
Embedded HTTP Servers (Tomcat,Jetty)
-
XML <bean> Configuration or @Configuration
=
Spring Boot

Figure 3: What is springboot[4]

Python

Python is a great decision for computerizing errands, composing contents, and coordinating with AWS Lambda with regards to a verification dashboard. Here are some normal use situations where Python can be advantageous:

Task Automation: Python gives a large number of libraries and modules that can be utilized to mechanize redundant undertakings in the verification dashboard. Whether it's information handling, record control, or connecting with APIs, Python's straightforwardness and broad library biological system make it appropriate for computerizing such undertakings.



```python
# Python Proofreading
# pip install lmproof
import lmproof
def proofread(text):
    proofread = lmproof.load("en")
    correction = proofread.proofread(text)
    print("Original: {}".format(text))
    print("Correction: {}".format(correction))

proofread("Your Text")
```

Figure 4: Using python for automation [7]

Scripting: Python's scripting capacities make it ideal for composing contents to perform explicit activities or execute a progression of orders inside the confirmation dashboard. For instance, you can compose a Python content to scratch information from verified finance dashboards, process the information, and store it in a data set.

AWS Lambda services: AWS Lambda permits you to run code without provisioning or overseeing servers. Python is one of the upheld programming dialects for AWS Lambda, making it simple to compose serverless capabilities that can be set off by occasions or Programming interface demands. You can use

Python to compose Lambda works that incorporate with the confirmation dashboard, perform information handling, send webhooks, or handle different errands on a case by case basis.

API Integration: Python has strong libraries for working with Relaxing APIs, permitting you to effortlessly associate with outer administrations and coordinate them into the verification dashboard. You can utilize libraries like solicitations or aiohttp to send HTTP demands, recover information, and cycle reactions from different APIs.

Data Manipulation and Analysis: Python offers strong libraries like Pandas, NumPy, and matplotlib that work with information control, investigation, and perception. You can use these libraries to process and examine the information got from the confirmation dashboard, produce experiences, and make visual portrayals of the information.

Testing and Quality Assurance: Python gives testing systems like unittest and pytest, which can be utilized to compose mechanized tests for the verification dashboard. These tests can guarantee the unwavering quality and accuracy of the code, approving its way of behaving and getting potential issues from the get-go in the advancement cycle.

Python's flexibility, usability, and broad local area support settle on it a well known decision for robotizing errands, composing contents, and incorporating with different innovations like AWS Lambda inside a verification dashboard. Its rich environment of libraries and structures empowers engineers to fabricate strong and effective arrangements.

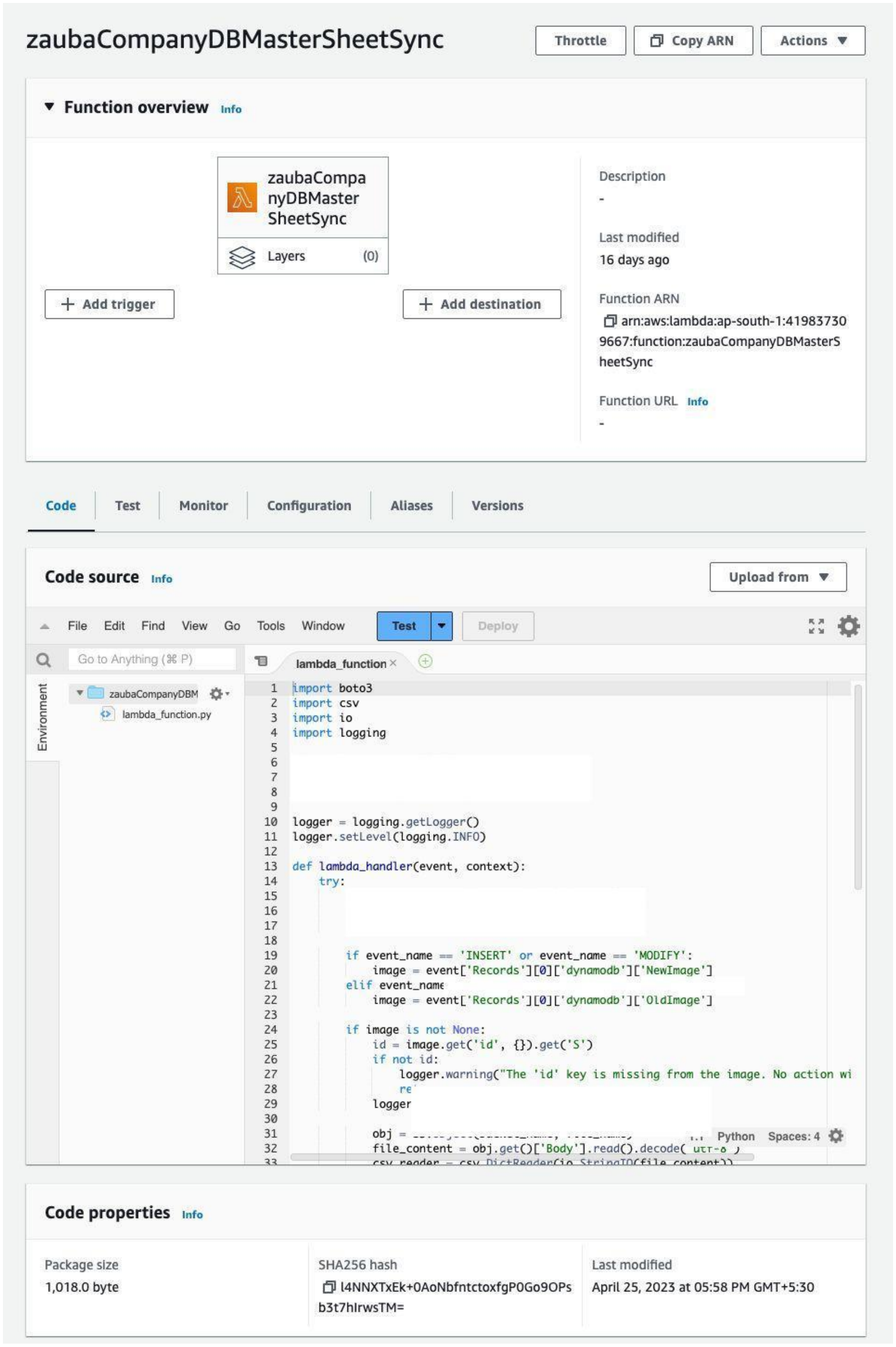
Web Scraping: Python has incredible libraries like BeautifulSoup and Scrapy that work with web scratching. You can utilize them to separate information from sites, including validated finance dashboards, and incorporate that information into your verification dashboard. This permits you to robotize the most common way of social occasion data from different sources.

Data Validation and Cleansing: Python gives devices to information approval and purifying, which can be pivotal in a confirmation dashboard. You can utilize normal articulations (Regex) to approve and disinfect client sources of info or information acquired from outside sources. Python's underlying string control abilities make it simple to clean and arrange information prior to handling or putting away it.

Integration with Database Systems: Python gives strong libraries to interfacing with various data set frameworks like MySQL, PostgreSQL, or MongoDB. You can utilize these libraries to interface with the information base where you store check information, perform data set tasks like questioning, embedding, or refreshing records, and recover information for additional handling or examination.

Generating Reports and Visualizations: Python's information control and perception libraries can be used to create reports and representations inside the verification dashboard. You can utilize libraries like Matplotlib, Plotly, or Seaborn to make diagrams, charts, and other visual portrayals of confirmation information, giving significant experiences to end-clients.

Error Logging and Monitoring: Python structures like Jar or Django can be utilized to fabricate the confirmation dashboard itself, giving an easy to understand connection point to end-clients. Also, you can use Python logging modules to execute mistake logging and checking, helping you track and analyze issues inside the dashboard.



Description
-

Last modified
16 days ago

Function ARN
arn:aws:lambda:ap-south-1:419837309667:function:zaubaCompanyDBMasterSheetSync

Function URL Info
-

```
import boto3
import csv
import io
import logging

[redacted]

logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):
    try:
        [redacted]

        if event_name == 'INSERT' or event_name == 'MODIFY':
            image = event['Records'][0]['dynamodb']['NewImage']
        elif event_name[redacted]
            image = event['Records'][0]['dynamodb']['OldImage']

        if image is not None:
            id = image.get('id', {}).get('S')
            if not id:
                logger.warning("The 'id' key is missing from the image. No action wi
                re[redacted]
            logger[redacted]

            obj = [redacted]
            file_content = obj.get()['Body'].read().decode('utf-8')
            csv_reader = csv.DictReader(io.StringIO(file_content))
```

Package size
1,018.0 byte

SHA256 hash
l4NNXTxEk+0AoNbfntctoxfgP0Go9OPsb3t7hlrwsTM=

Last modified
April 25, 2023 at 05:58 PM GMT+5:30

Figure 5: Python Lambda function to manage 22lac data entries

AWS Lambda

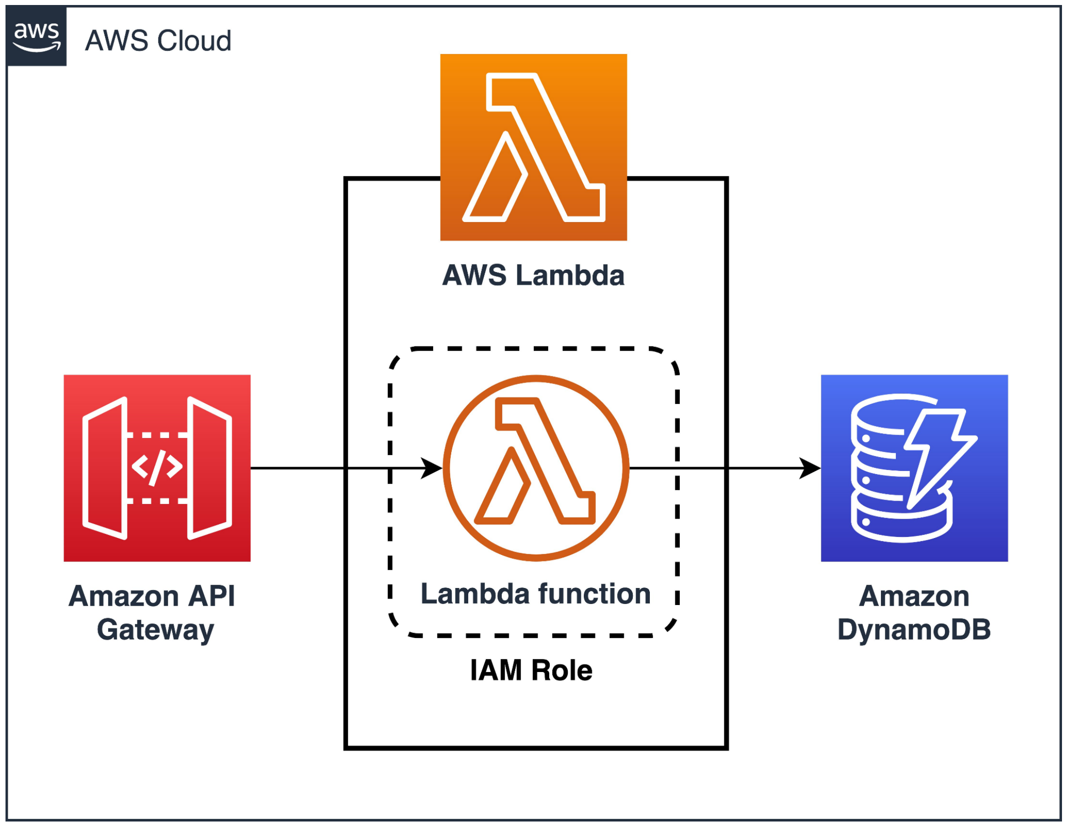


Figure 6: Overview of AWS Lambda [6]

Serverless Architecture: AWS Lambda permits you to carry out a serverless engineering, where you can execute code without provisioning or overseeing servers. This dispenses with the requirement for foundation the board, empowering you to zero in on building the confirmation dashboard's usefulness. You can convey your confirmation administrations as Lambda capabilities, which are set off by occasions or Programming interface demands.

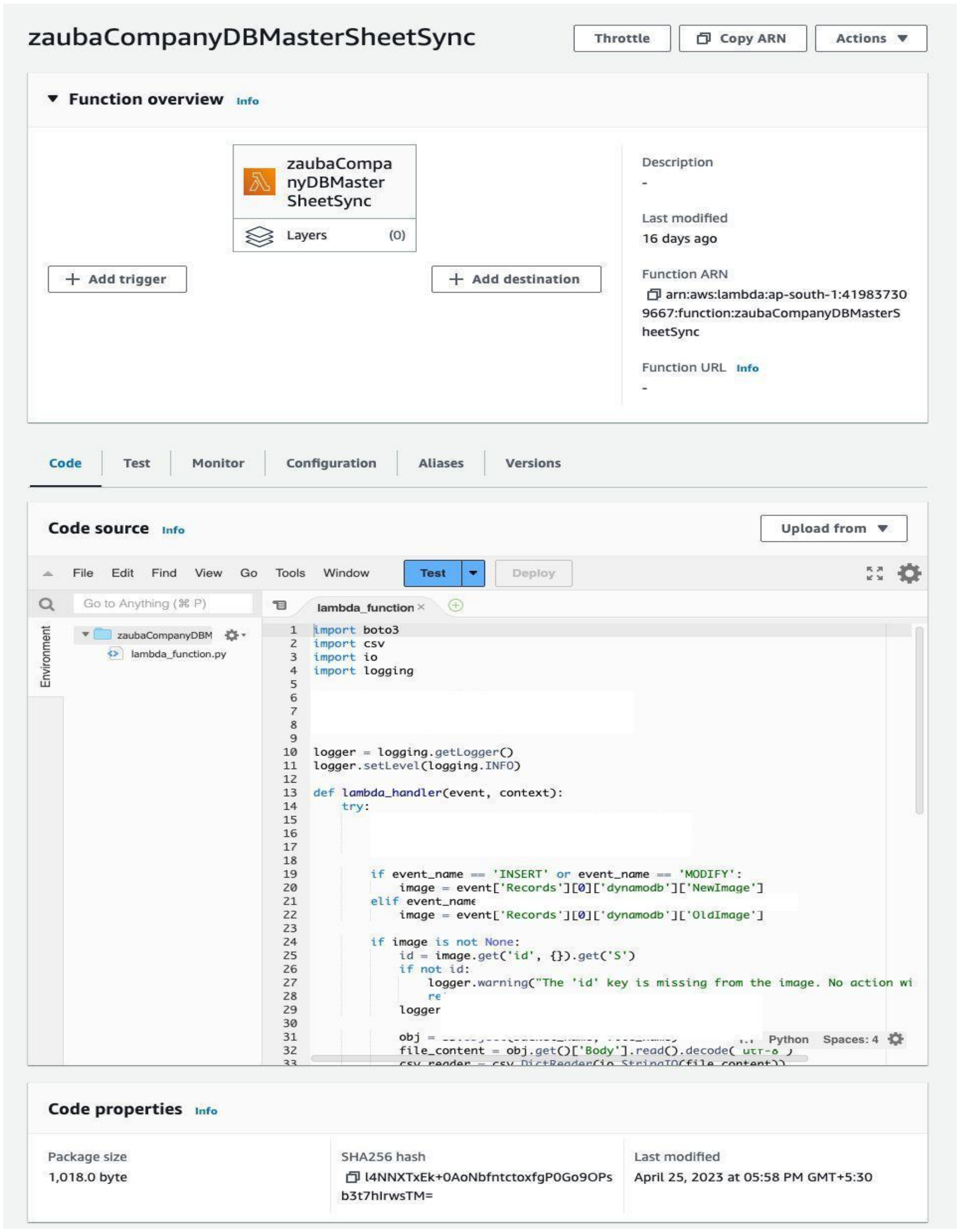
Integrations and APIs: Lambda capabilities can act as coordination focuses between various parts of the verification dashboard and outside administrations. For example, you can compose Lambda capabilities to communicate with outside confirmation frameworks, like finance dashboards, and recover information for check purposes. The Lambda capabilities can be conjured by the confirmation dashboard's APIs, empowering consistent reconciliation and information trade.

Data Processing and Validation: AWS Lambda permits you to carry out information handling and approval rationale inside the confirmation dashboard. For instance, you can compose Lambda capabilities to approve and purge client inputs, apply business rules to confirm information, or perform complex estimations. Lambda capabilities give a versatile and proficient method for taking care of information handling errands, guaranteeing that the dashboard works flawlessly considerably under high burden.

Asynchronous Processing: AWS Lambda upholds nonconcurrent execution, which is helpful for long-running or asset-escalated errands inside the confirmation dashboard. For example, in the event that specific confirmation processes require critical computational assets or include outer Programming interface calls, you can offload those errands to Lambda capabilities. By conjuring these capabilities nonconcurrently, you can decouple the handling from the principal dashboard stream, further developing execution and versatility.

Cost Advancement: AWS Lambda offers cost enhancement benefits by charging just for the genuine register time consumed by your capabilities. With Lambda, you don't have to pay for inactive server time, which can bring about cost investment funds, particularly while managing inconsistent check demands. Lambda consequently scales in view of the approaching responsibility, guaranteeing that you have the important assets to deal with check demands proficiently without bringing about superfluous expenses.

Error Handling and Retry Mechanisms: AWS Lambda gives worked in blunder dealing with and retry systems. At the point when a Lambda capability experiences a mistake, it tends to be designed to retry the activity or trigger a warning for additional examination consequently. This guarantees the power and dependability of the confirmation dashboard, as mistakes can be logged, observed, and tended to on time.



Description
-

Last modified
16 days ago

Function ARN
arn:aws:lambda:ap-south-1:419837309667:function:zaubaCompanyDBMasterSheetSync

Function URL Info
-

```
import boto3
import csv
import io
import logging

[redacted]

logger = logging.getLogger()
logger.setLevel(logging.INFO)

def lambda_handler(event, context):
    try:
        [redacted]

        if event_name == 'INSERT' or event_name == 'MODIFY':
            image = event['Records'][0]['dynamodb']['NewImage']
        elif event_name [redacted]
            image = event['Records'][0]['dynamodb']['OldImage']

        if image is not None:
            id = image.get('id', {}).get('S')
            if not id:
                logger.warning("The 'id' key is missing from the image. No action wi
                re [redacted]
            logger [redacted]

            obj = [redacted]
            file_content = obj.get()['Body'].read().decode( utr-8 )
            csv_reader = csv.DictReader(io.StringIO(file_content))
```

Package size	SHA256 hash	Last modified
1,018.0 byte	l4NNXTxEk+0AoNbfntctoxfgP0Go9OPsb3t7hlrwsTM=	April 25, 2023 at 05:58 PM GMT+5:30

Figure 7: Python Lambda function to manage 22lac data entries

AWS S3

Data Storage: AWS S3 gives a profoundly versatile and sturdy item stockpiling administration. You can utilize S3 to store different kinds of information connected with the verification dashboard, for example, client archives, confirmation results, setup documents, or some other important records. S3 offers high accessibility, information overt repetitiveness, and solidness, it is protected and available to guarantee that your information.

File Uploads and Downloads: Verification dashboards frequently include record transfers and downloads. AWS S3 can act as the stockpiling backend for these tasks. Clients can transfer archives, payslips, or other expected records to S3 straightforwardly from the dashboard. Likewise, confirmation results or created reports can be put away in S3 and made accessible for download through the dashboard. This empowers consistent record dealing with and guarantees proficient information trade between the dashboard and clients.

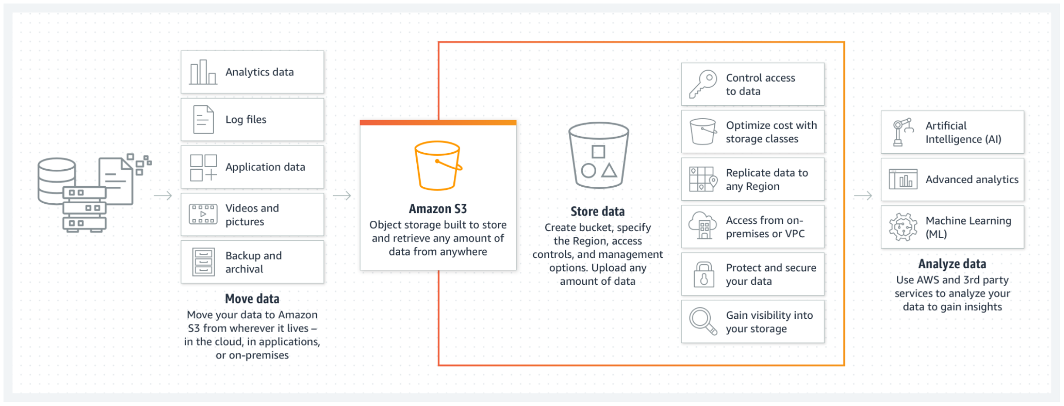
Data Archiving and Backup: AWS S3 gives long haul stockpiling abilities, making it ideal for information filing and reinforcement purposes. You can arrange S3 to store authentic check information, logs, or different records for consistence or review necessities. S3's forming highlight permits you to keep a past filled with record updates, guaranteeing information honesty and empowering simple recovery of past variants if necessary.

Secure Data Access: AWS S3 offers fine-grained admittance control instruments, permitting you to characterize access approaches and consents for the put away information. You can set up access control records (leg tendons), container approaches, or use AWS Personality and Access The executives (IAM) to oversee who can get to and control the confirmation information put away in S3. This guarantees information security and privacy, forestalling unapproved admittance to delicate data.

Data Processing and Analytics: S3 can be coordinated with other AWS administrations like AWS glue crawler, Amazon Athena, or Amazon Redshift for information handling and examination purposes. Verification information put away in S3 can be handled, changed, or examined utilizing these administrations to produce experiences, run questions, or assemble information pipelines. This empowers you to separate important data from the confirmation information and determine noteworthy insight.

Scalability and Performance: AWS S3 is intended to deal with gigantic measures of information and simultaneous solicitations. It offers high versatility and can consistently deal with expanding stockpiling and recovery requests. This adaptability guarantees that the confirmation dashboard can oblige developing client bases and expanding information volumes without compromising execution.

Integration with Other AWS services: AWS S3 incorporates well with other AWS administrations, like AWS Lambda, Amazon Programming interface Door, or Amazon CloudFront. For instance, you can design Lambda capabilities to set off upon record transfers to S3, empowering robotized handling or information approval. Programming interface Door can be utilized to give secure and controlled admittance to S3 assets through APIs. CloudFront can further develop the conveyance speed and dependability of records put away in S3 by reserving content at edge areas.



Analytics data
Log files
Application data
Videos and pictures
Backup and archival
Move data
Move your data to Amazon S3 from wherever it lives – in the cloud, in applications, or on-premises
Amazon S3
Object storage built to store and retrieve any amount of data from anywhere
Store data
Create bucket, specify the Region, access controls, and management options. Upload any amount of data
Control access to data
Optimize cost with storage classes
Replicate data to any Region
Access from on-premises or VPC
Protect and secure your data
Gain visibility into your storage
Artificial Intelligence (AI)
Advanced analytics
Machine Learning (ML)
Analyze data
Use AWS and 3rd party services to analyze your data to gain insights

Figure 8: Cloud Object storage [9]

AWS EC2

While building a verification dashboard, AWS EC2 (Elastic Compute Cloud) cases can be used for different purposes. Here are some particular use instances of AWS EC2 inside a confirmation dashboard:

Application Hosting: You can convey the verification dashboard application on EC2 occurrences. EC2 gives virtual servers in the cloud, permitting you to run your application and web server stack. You have full command over the working framework, designs, and programming stack, empowering you to fit the climate as indicated by the particular requirements of the dashboard.

Backend Processing: EC2 examples can be utilized for backend handling assignments inside the confirmation dashboard. For instance, on the off chance that there are computationally escalated undertakings like information change, complex estimations, or AI calculations engaged with the confirmation cycle, you can offload those assignments to EC2 examples. These examples give adaptable figure assets to deal with such handling prerequisites effectively.

Database Hosting: EC2 occasions can have data sets expected by the verification dashboard. Whether you are utilizing social information bases like MySQL or

PostgreSQL, or NoSQL data sets like MongoDB or Amazon DynamoDB, you can introduce and design the data set programming on EC2 cases. This permits you to have full command over the information base climate and designer it to the particular necessities of the dashboard.

Scalability and Auto Scaling: EC2 gives auto scaling abilities, which permit your confirmation dashboard to consequently change the quantity of examples in view of interest. As the client load increments or diminishes, EC2 can increase or down the quantity of occasions, guaranteeing ideal execution and cost proficiency. This aides in dealing with shifting check jobs and obliging spikes in client traffic.

Load Balancing: When the verification dashboard encounters high traffic or requires adaptation to non-critical failure, you can use EC2 cases related to a Versatile Burden Balancer (ELB). The ELB conveys approaching solicitations across numerous EC2 occurrences, guaranteeing that the heap is equitably disseminated and the dashboard remains profoundly accessible and responsive.

Hybrid Architectures: On the off chance that your verification dashboard requires joining with on-premises frameworks or other cloud administrations, you can set up EC2 examples as a piece of a mixture engineering. EC2 occurrences can act as the scaffold between the confirmation dashboard in the cloud and the on-premises frameworks, empowering secure and productive correspondence and information trade.

Development and Testing: EC2 occurrences are likewise valuable for improvement and testing purposes. You can arrangement EC2 occurrences with the fundamental programming and setups to repeat the creation climate or establish secluded conditions for testing new elements, changes, or updates to the confirmation dashboard.

AWS EC2 gives adaptability, versatility, and command over the foundation for facilitating, handling, and putting away information inside a verification dashboard. It permits you to fit the climate to suit the particular prerequisites of the dashboard, scale assets in view of interest, and coordinate with other AWS administrations or on-premises frameworks flawlessly.

AWS ETL Job, Crawler, Dynamo DB

Amazon ETL (Extract, Transform, Load), AWS Glue Crawler, and Amazon DynamoDB can be utilized together in an application to work with information extraction, change, and capacity. This is an outline of the way these administrations can cooperate:

Amazon ETL (AWS glue): Amazon ETL, fueled by AWS glue, is a completely overseen remove, change, and burden (ETL) administration that helps you plan and change your information for examination. It gives a visual connection point to characterizing ETL work processes and naturally produces code to play out the changes.

AWS glue Crawler: AWS glue Crawler is a part of AWS glue that naturally finds and inventories metadata about your information sources. It filters information stores like Amazon S3, data sets, and information stockrooms to deduce outline and segment structure. The crawler distinguishes the information arrangement, mapping, and other pertinent metadata.

Amazon DynamoDB: Amazon DynamoDB is a completely overseen NoSQL data set help given by Amazon Web Administrations. It offers an adaptable, versatile, and profoundly accessible information base answer for applications that require low-inactivity information access. DynamoDB stores information in a key-esteem design and is known for its speed and versatility.

To comprehend how these administrations can be utilized together, how about we think about a model situation:

Assume you have a lot of information put away in different sources, like Amazon S3 or data sets, and you need to remove, change, and burden this information into Amazon DynamoDB for additional handling or questioning.

Information Extraction: The initial step is to extricate the information from your sources. You can utilize AWS glue to characterize ETL occupations that indicate the information sources (e.g., S3 pails, data sets), the information designs (e.g., CSV, JSON), and any essential accreditations for getting to the information.

Information Change: When the information is extricated, you can utilize AWS glue to perform changes on the information. This can incorporate cleaning, sifting, collecting, or improving the information in light of your application necessities. You can outwardly characterize the changes in AWS glue's ETL interface or compose custom code utilizing Apache Flash.

Information Stacking: After the information is changed, you can utilize AWS glue to stack the changed information into Amazon DynamoDB. AWS glue furnishes connectors to associate with different information stockpiling frameworks, including DynamoDB. You can indicate the table construction and planning between the changed information and the DynamoDB outline.

AWS glue Crawler (Metadata List): To monitor the information put away in Amazon DynamoDB and make it effectively discoverable, you can set up an AWS glue Crawler. The crawler can occasionally check the DynamoDB tables, surmise the diagram, and update the metadata list in AWS glue. This metadata list helps in sorting out and questioning the information productively.

By consolidating these administrations, you can computerize the method involved with separating, changing, and stacking information from various sources into Amazon DynamoDB. This empowers you to fabricate applications that can effectively deal with huge volumes of information and influence the adaptability and execution advantages of DynamoDB.

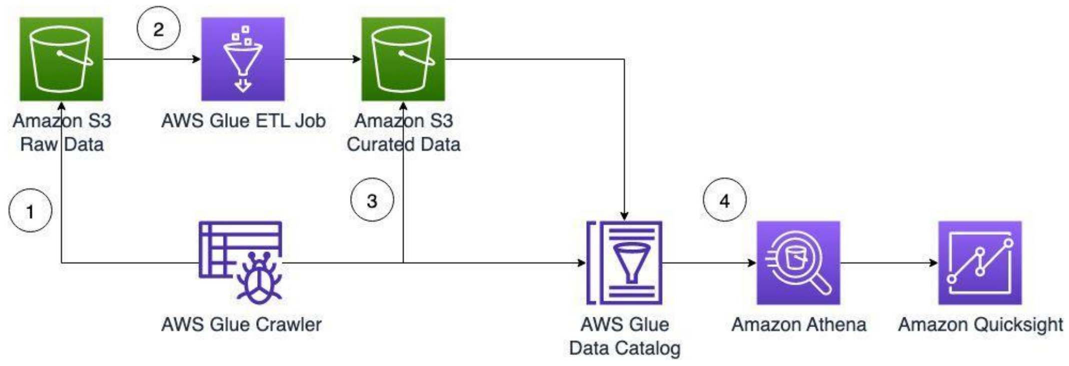


Figure 9: AWS Glue Crawler [9]

Chapter 03: System Development

Feasibility Study

In this feasibility study, the following project components will be examined:

- Technological feasibility
- Operational feasibility
- Financial feasibility

Technological feasibility

Utilizing Spring Boot to foster a verification dashboard is in fact practical and favorable. Spring Boot works on improvement with its show over-config approach and gives a ready-to-use env. It uses the broad highlights of the Spring framework and offers incredible help for building restful APIs and carrying out safety efforts. Spring Boot coordinates well with front-end innovations, and its full-grown local area and documentation give significant assets. In general, Spring Boot is a reasonable decision for building a verification dashboard, taking into account its fast improvement capacities and vigorous toolset.

Operational feasibility

Utilizing Spring Boot to construct a verification dashboard offers high functional possibility. It works on simplification of maintenance and deployment, gives versatility and execution advancements, upholds observing and logging, works with setup the executives, coordinates well with DevOps practice, is viable with cloud platforms, and empowers proactive checking and auditing. These elements add to a smooth and proficient operation of the dashboard. Furthermore, it offers vigorous error handling with, broad testing systems, application monitoring and management abilities, and customary updates.

Financial feasibility

The financial feasibility of involving Spring Boot for a verification dashboard relies upon a few variables. While Spring Boot itself is free and open-source, there are other expense contemplations. Improvement expenses can be limited on the off chance that the dev team is knowledgeable about Spring Boot, prompting quicker deployments and decreased costs. Versatility and asset efficiency can add to cost reserve funds by permitting the application to deal with expanded client loads without critical equipment ventures. Infrastructure costs fluctuate contingent upon the sending climate, with cloud stages offering adaptable evaluating choices. Support expenses can be decreased because Spring Boot offers flexible pricing options. Combination costs and authorizing consistency ought to likewise be thought of. Training and support might be required assuming the advancement group is new to Spring Boot. Customization and continuous upkeep should be represented as far as advancement and backing costs. Generally, the financial feasibility relies upon an extensive assessment of these variables and the normal profit from interest concerning further developed proficiency and consumer loyalty.

Requirements

The capability of the system to fulfill the conditions sought by the users is known as a system requirement. By dividing the needs into functional and nonfunctional requirements,system requirement analysis is accomplished.

Functional requirements

1. Authentication

- User authentication and authorization should be supported via the API.
- Users ought to be able to make accounts and sign in with them.
- User roles and permissions ought to be established and upheld.

2. Data Management

- The Create, Read, Update, and Delete operations of the API should be able to manage data.
- In a database or file system, data ought to be kept.
- Data inputs should be verified and consistency monitored by the API.

3. Data Filtering & Sorting

- The API should support sorting and filtering of data according to various criteria.
- Users should have the option to query data using particular criteria.

4. Error Handling

- The errors should be handled hasslefree and the status codes returned should be correct for all the edge cases.
- Error messages should be accurate

5. Testing

- API functions should pass all the test cases with 100 percent coverage.
- Mocks should be used properly and dependency injection should be done for unit testing for all the layers.

6. Documentation

- Documentation should be correct and sorted for all the functions including the correct naming conventions.

Non-Functional Requirements

1. Scalability
2. Performance
3. Readability

4. Reliability
5. Usable API
6. Maintainability

Technical Requirements

1. An IDE for writing clean code is called Java.
2. Postman, an API development and usage platform.
3. Mysql server offers connectivity and querying capabilities for database administration systems
4. Confluence for API documentation
5. Git version control

Hardware Configurations

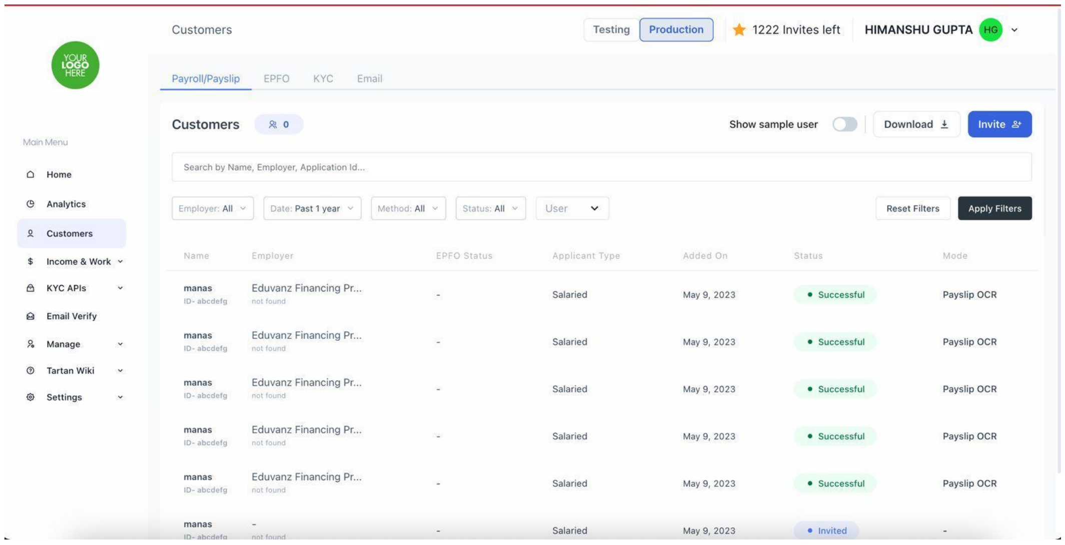
1. Macbook Pro 13inch M1 chip
2. Memory 8.0 GiB
3. Disk Capacity 256 GB
4. Monitor 13”
5. Mouse
6. Keyboard

Software Configurations

1. Operating System MacOS
2. Language Java
3. Runtime environment JDK 1.8 runtime
4. Package Manager Maven

Implementation

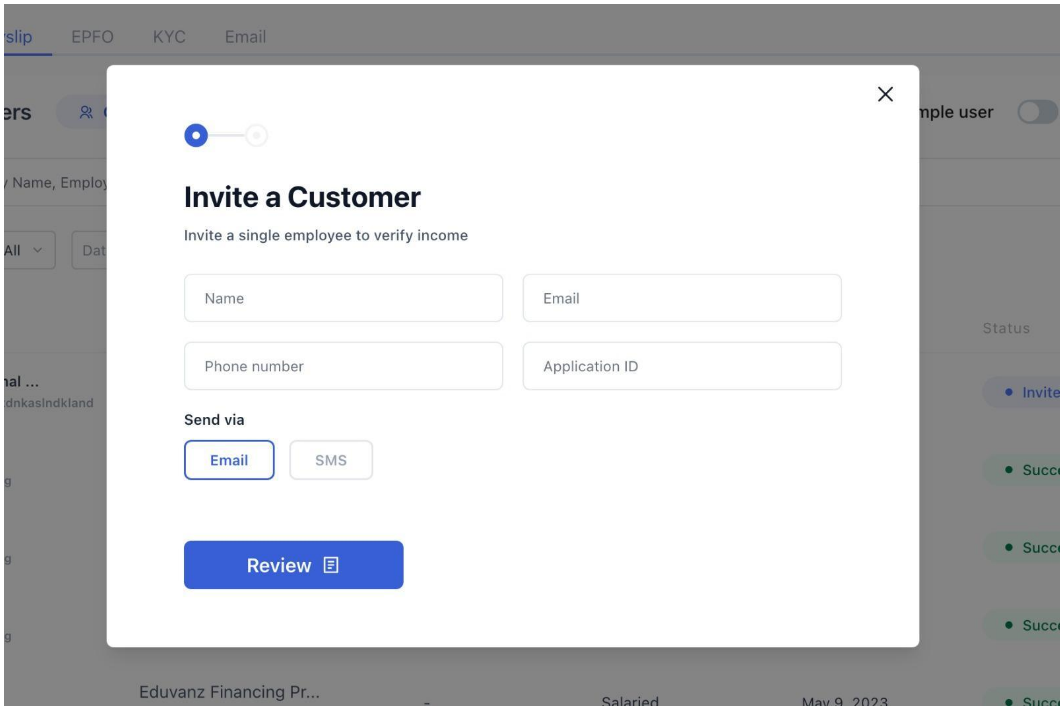
Whenever an end user logs in to his premade profile of Tartan, he sees this kind of dashboard in front of him. The customer is the client bank for example ICICI, HDFC, etc. And the end user is the one who has his banking account made in HDFC or similar banks and he wants to verify his payslips using our verification dashboard. Hence, this dashboard connects the end users to our application and the banks use us as their portals.



Name	Employer	EPFO Status	Applicant Type	Added On	Status	Mode
manas ID- abcdefg	Eduvanz Financing Pr... not found	-	Salaried	May 9, 2023	● Successful	Payslip OCR
manas ID- abcdefg	Eduvanz Financing Pr... not found	-	Salaried	May 9, 2023	● Successful	Payslip OCR
manas ID- abcdefg	Eduvanz Financing Pr... not found	-	Salaried	May 9, 2023	● Successful	Payslip OCR
manas ID- abcdefg	Eduvanz Financing Pr... not found	-	Salaried	May 9, 2023	● Successful	Payslip OCR
manas ID- abcdefg	Eduvanz Financing Pr... not found	-	Salaried	May 9, 2023	● Successful	Payslip OCR
manas ID- abcdefg	- not found	-	Salaried	May 9, 2023	● Invited	-

Figure 10: Dashboard overview

After this, the next step in this journey is creating an invite to the end user by the clients for their verification of KYC, payslips, payrolls, emails, etc. The invite is sent by the banks themselves to the end users. There is also na option of sending the invite through email or through SMS. Details of the end users such as their name, email id, phone number, and application id are necessary for proceeding.



Invite a Customer

Invite a single employee to verify income

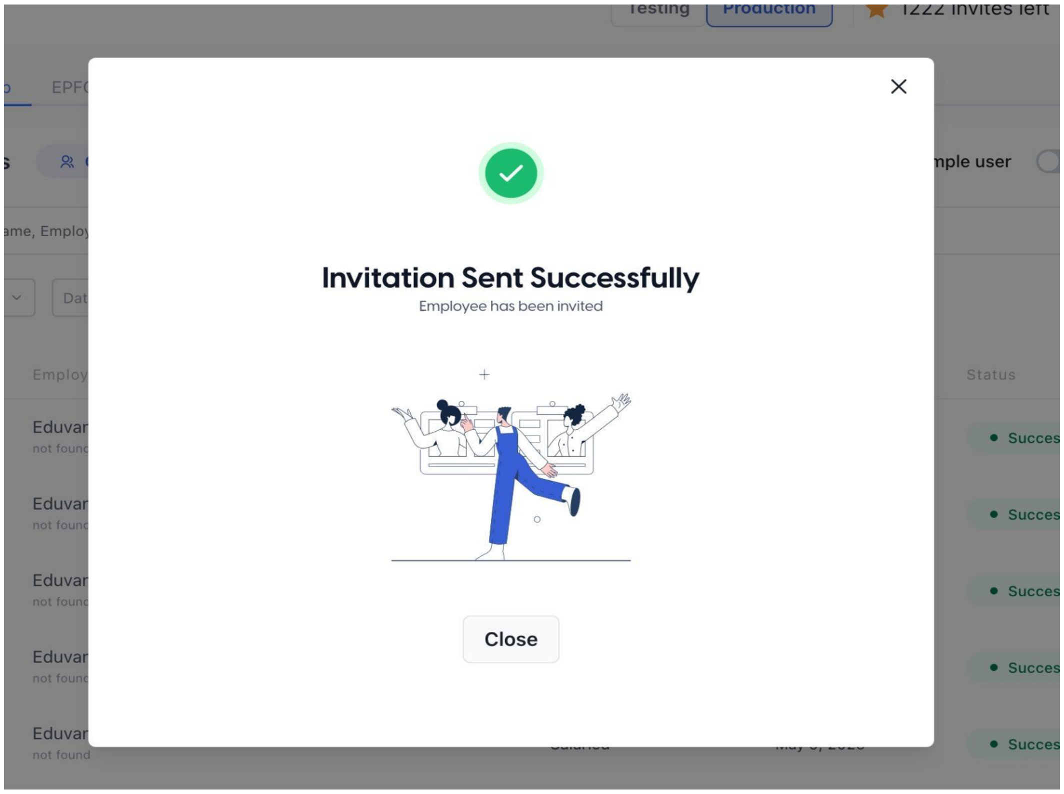
Name	Email
Phone number	Application ID

Send via

[✓ Email] SMS

Review

Figure 11: Sending an invitation to the end user



Invitation Sent Successfully

Employee has been invited

Close

Figure 12: Prompt that is displayed after an invite has been sent.

On the end user's side, he receives an email such as shown below. There is a button on which he can click and verify his work and income.

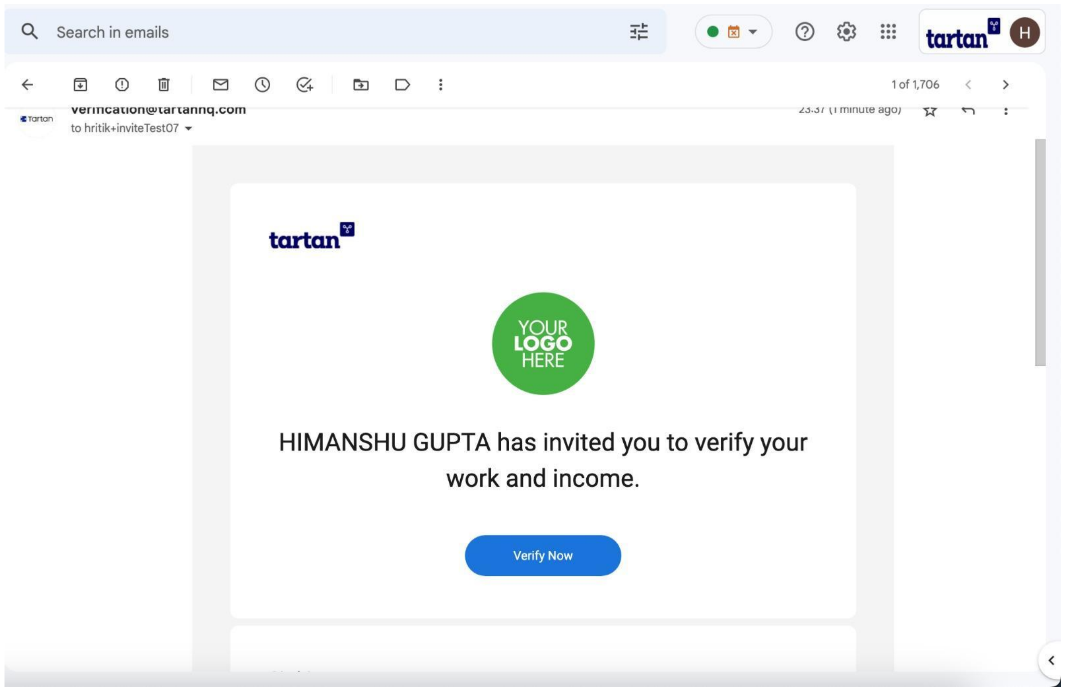
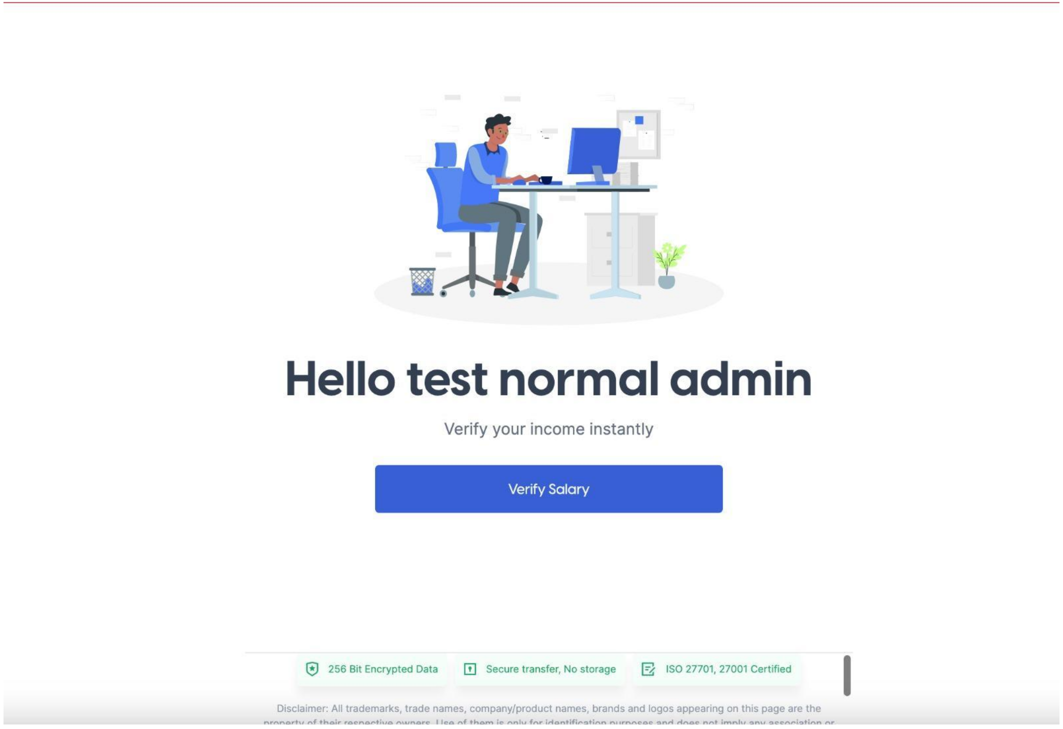

HIMANSHU GUPTA has invited you to verify your work and income.

Figure 13: Email received by customer

After clicking on the verification button, the end user proceeds toward the landing page of our verification dashboard. There he finds a prompt that will lead him further on in his journey.

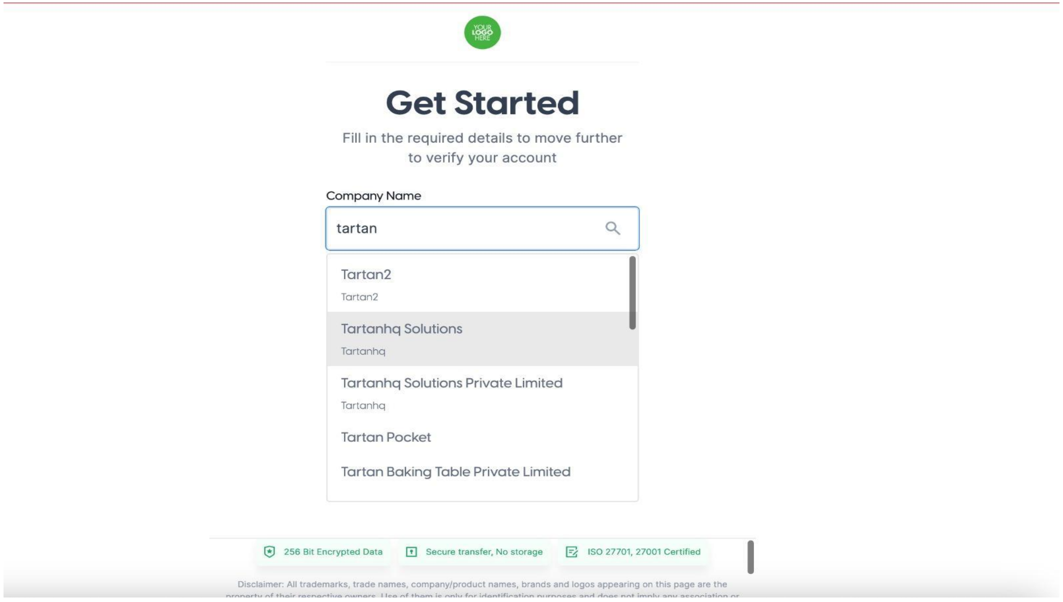

Hello test normal admin

Verify your income instantly

Disclaimer: All trademarks, trade names, company/product names, brands and logos appearing on this page are the property of their respective owners. Use of them is only for identification purposes and does not imply any association or

Figure 14: Verification dashboard's landing page

Then, the end user would be redirected to a form where he has to fill in his organization details.


Get Started

Fill in the required details to move further to verify your account

Company Name: tartan

- Tartan2 — Tartan2
- Tartanhq Solutions — Tartanhq
- Tartanhq Solutions Private Limited — Tartanhq
- Tartan Pocket
- Tartan Baking Table Private Limited

Disclaimer: All trademarks, trade names, company/product names, brands and logos appearing on this page are the property of their respective owners. Use of them is only for identification purposes and does not imply any association or

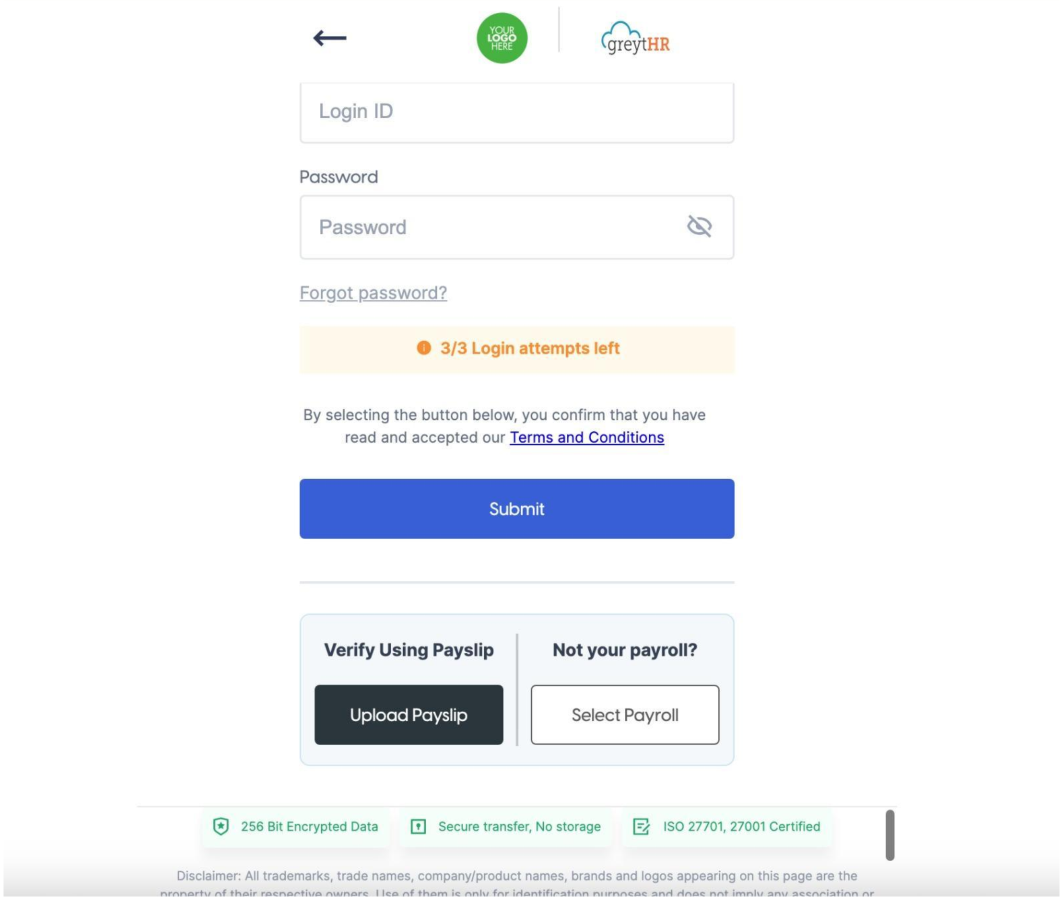
Figure 15: Organization name

After that, he could log in using his email id and password. He gets 3 attempts only because of security issues. Apart from that, he also gets an option to follow either of two data flows which are

- Visualize
- Sync

In Visualize, the user follows a payslip upload flow. From there, the data points are retrieved, and then verification is carried out.

In Sync, the user is logged in to the system through his payroll. From there, the data points are retrieved, and then verification is carried out.

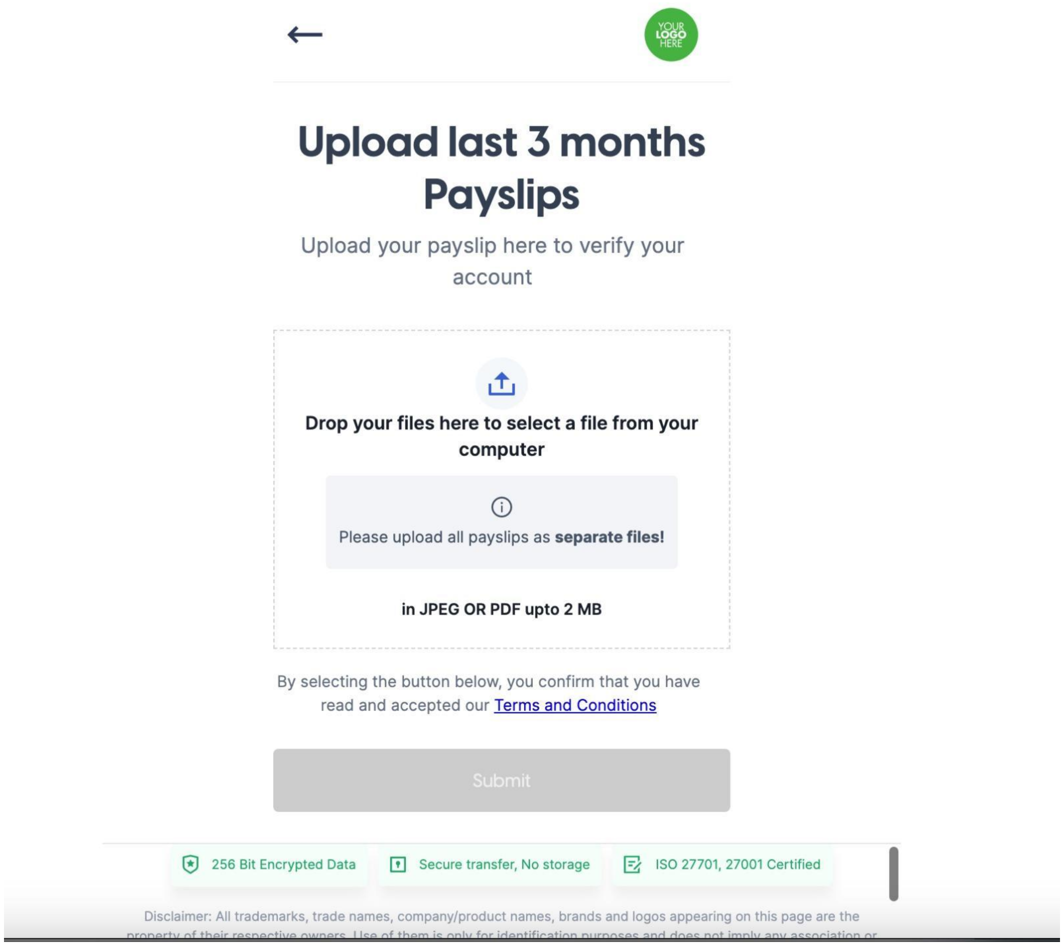


By selecting the button below, you confirm that you have read and accepted our Terms and Conditions

Disclaimer: All trademarks, trade names, company/product names, brands and logos appearing on this page are the property of their respective owners. Use of them is only for identification purposes and does not imply any association or

Figure 16: Login using payslip/payroll

Post this step, the end user is required to upload his payslips for the last three months. After that, an AI-generated algorithm is carried out for verification.


YOUR LOGO HERE

Upload last 3 months Payslips

Upload your payslip here to verify your account

Drop your files here to select a file from your computer

Please upload all payslips as **separate files!**

in JPEG OR PDF upto 2 MB

By selecting the button below, you confirm that you have read and accepted our Terms and Conditions

Submit

256 Bit Encrypted Data

Secure transfer, No storage

ISO 27701, 27001 Certified

Disclaimer: All trademarks, trade names, company/product names, brands and logos appearing on this page are the property of their respective owners. Use of them is only for identification purposes and does not imply any association or

Figure 17: File upload

Once the customer has completed the verification process, a flow is initiated to utilize AI technology for extracting data points from the customer's payslip. This process includes the following steps:

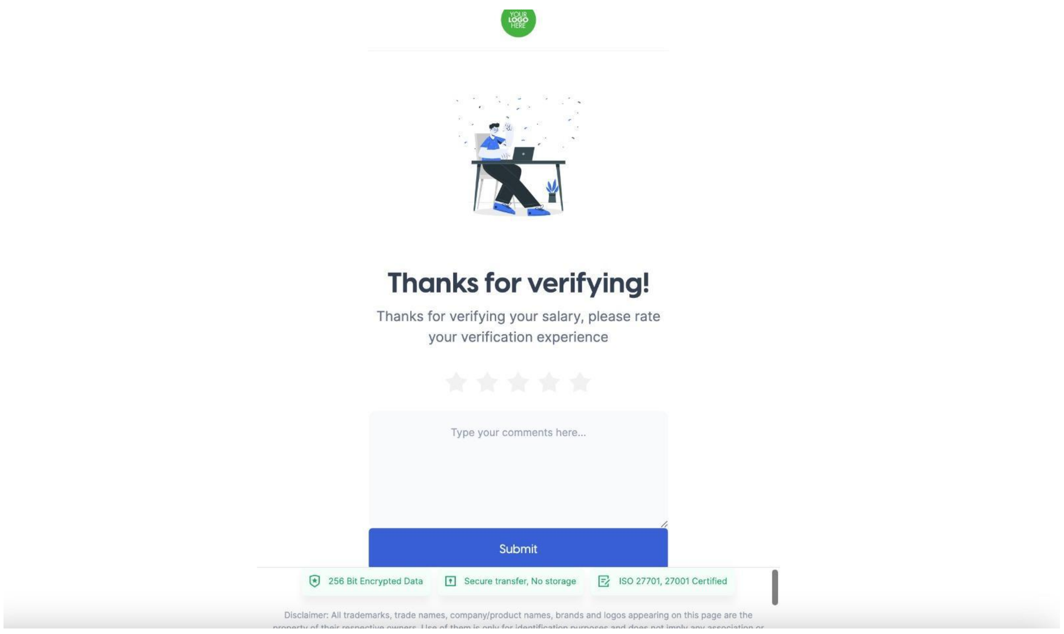
1. Data extraction: The AI algorithm scans the payslip document and identifies relevant data points, such as deductions, allowances, salary amounts, tax information, and other relevant details.

2. Quality checks: These data points are then validated against an automated AI algorithm to ensure accuracy and integrity. The parameters performed include:

- Payslip is cropped
- Payslip is fraud
- Payslip is truncated

3. Mapping to Java object: When the information passes the quality checks, it is mapped into a Java object. This item is intended to catch and store exhaustive data from the payslip, including monetary subtleties, PII (like name, date of birth, and Government managed retirement Number), private location, and some other significant information expected for finance handling.

4. Database storage: The Java object, containing the mapped client data, is then saved into a safe data set. This guarantees the information is put away in an organized way and can be handily gotten to for finance the executives and other significant purposes.



Thanks for verifying!

Thanks for verifying your salary, please rate your verification experience

256 Bit Encrypted Data

Secure transfer, No storage

ISO 27701, 27001 Certified

Disclaimer: All trademarks, trade names, company/product names, brands and logos appearing on this page are the

Figure 18: Prompt shown after the completion of the entire process

Additional features:

- If an unauthenticated user tries to access the restricted URLs, they will be shown an appropriate message.

Chapter 4: Performance Analysis

There are various problems that were resolved using the methodology mentioned above. Some of them are listed below.

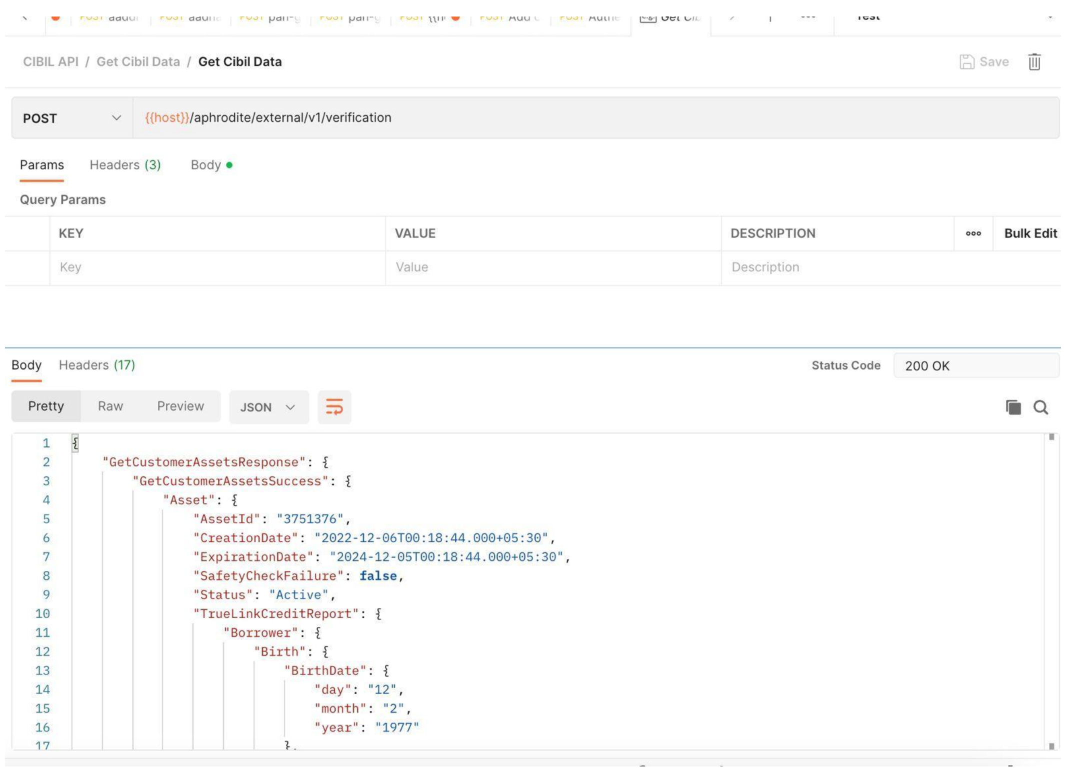
1) Reduction in searching time from 30ms to 20ms.

   With the help of multiple indices of databases and Elasticsearch, searching time could be made efficient. This could be done through Index optimization, choosing relevant fields, using appropriate data types, and defining explicit mappings that define how fields should be indexed and stored, allowing Elasticsearch to optimize the search process.

   Along with that, query caching, implementing pagination, leveraging filter queries, etc. were used in optimizing the time.

2) Integration of Third-party APIs

   We have integrated third-party APIs. These API calls were supposed to be highly secure, so we had to create a secure SSL tunnel using the certificate and key files to encrypt the ping. This is for cibil APIs provided by Transunion.


CIBIL API / Get Cibil Data / Get Cibil Data

POST {{host}}/aphrodite/external/v1/verification

Params Headers (3) Body

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Headers (17) Status Code 200 OK

Pretty Raw Preview JSON

```
{
    "GetCustomerAssetsResponse": {
        "GetCustomerAssetsSuccess": {
            "Asset": {
                "AssetId": "3751376",
                "CreationDate": "2022-12-06T00:18:44.000+05:30",
                "ExpirationDate": "2024-12-05T00:18:44.000+05:30",
                "SafetyCheckFailure": false,
                "Status": "Active",
                "TrueLinkCreditReport": {
                    "Borrower": {
                        "Birth": {
                            "BirthDate": {
                                "day": "12",
                                "month": "2",
                                "year": "1977"
```

Figure 19: Retrieving Cibil Data

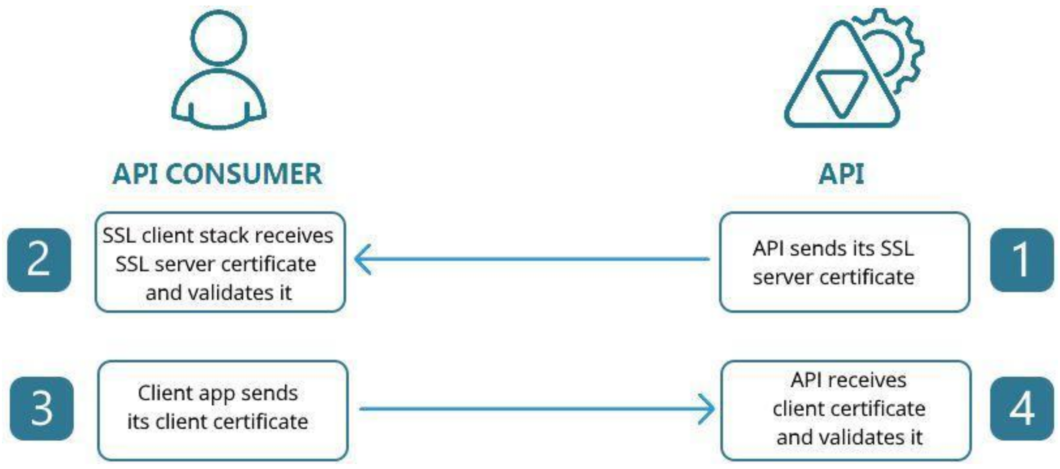
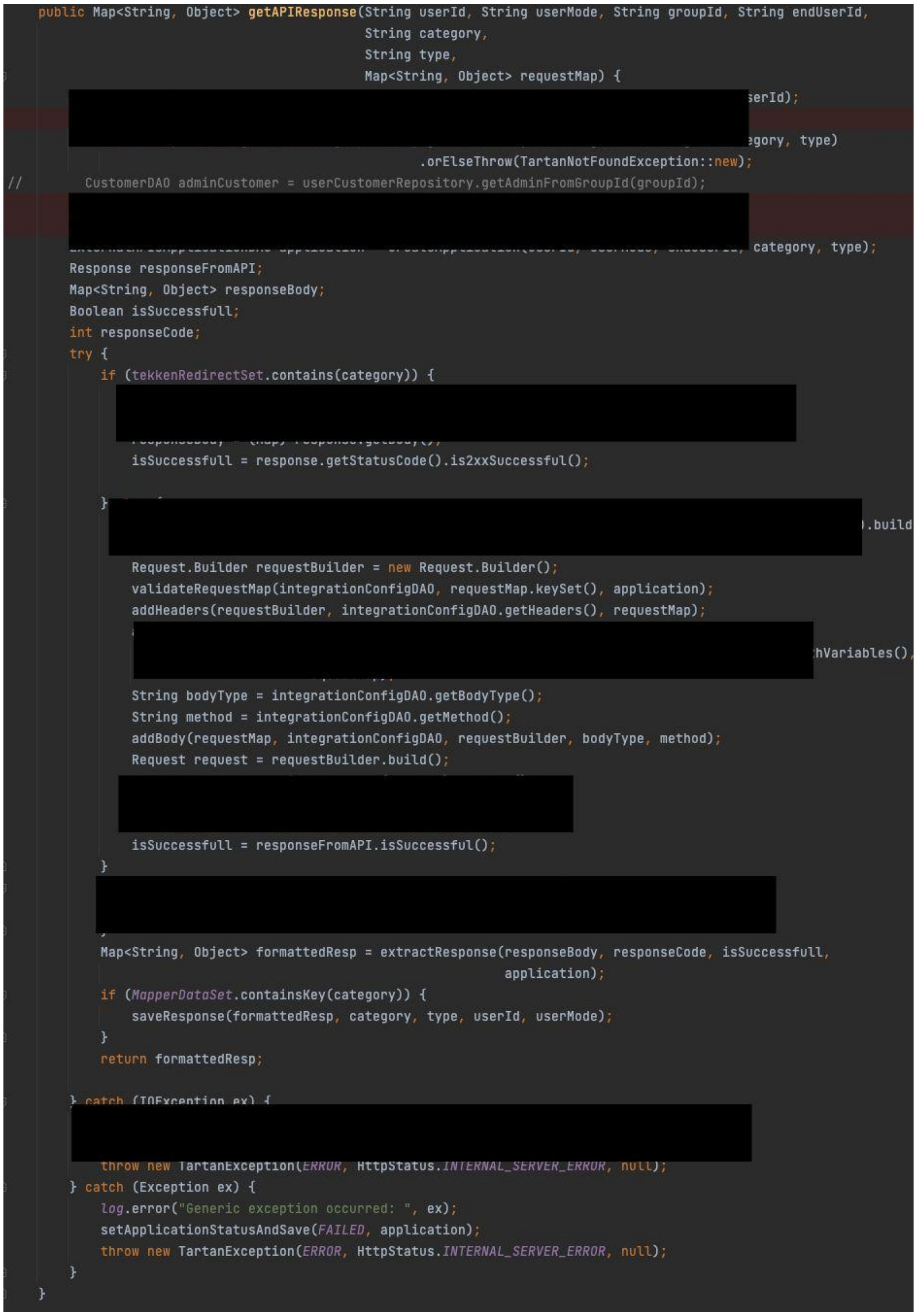

API CONSUMER
API
1
API sends its SSL server certificate
2
SSL client stack receives SSL server certificate and validates it
3
Client app sends its client certificate
4
API receives client certificate and validates it

Figure 20: SSL tunnel [6]



```java
    public Map<String, Object> getAPIResponse(String userId, String userMode, String groupId, String endUserId,
                                              String category,
                                              String type,
                                              Map<String, Object> requestMap) {
        [redacted] serId);
                                                                          egory, type)
                                                     .orElseThrow(TartanNotFoundException::new);
//        CustomerDAO adminCustomer = userCustomerRepository.getAdminFromGroupId(groupId);
        [redacted] category, type);
        Response responseFromAPI;
        Map<String, Object> responseBody;
        Boolean isSuccessfull;
        int responseCode;
        try {
            if (tekkenRedirectSet.contains(category)) {
                [redacted]
                isSuccessfull = response.getStatusCode().is2xxSuccessful();

            } [redacted] ).build
                Request.Builder requestBuilder = new Request.Builder();
                validateRequestMap(integrationConfigDAO, requestMap.keySet(), application);
                addHeaders(requestBuilder, integrationConfigDAO.getHeaders(), requestMap);
                [redacted] hVariables(),
                String bodyType = integrationConfigDAO.getBodyType();
                String method = integrationConfigDAO.getMethod();
                addBody(requestMap, integrationConfigDAO, requestBuilder, bodyType, method);
                Request request = requestBuilder.build();
                [redacted]
                isSuccessfull = responseFromAPI.isSuccessful();
            }
            [redacted]
            Map<String, Object> formattedResp = extractResponse(responseBody, responseCode, isSuccessfull,
                                                                application);
            if (MapperDataSet.containsKey(category)) {
                saveResponse(formattedResp, category, type, userId, userMode);
            }
            return formattedResp;

        } catch (IOException ex) {
            [redacted]
            throw new TartanException(ERROR, HttpStatus.INTERNAL_SERVER_ERROR, null);
        } catch (Exception ex) {
            log.error("Generic exception occurred: ", ex);
            setApplicationStatusAndSave(FAILED, application);
            throw new TartanException(ERROR, HttpStatus.INTERNAL_SERVER_ERROR, null);
        }
    }
```

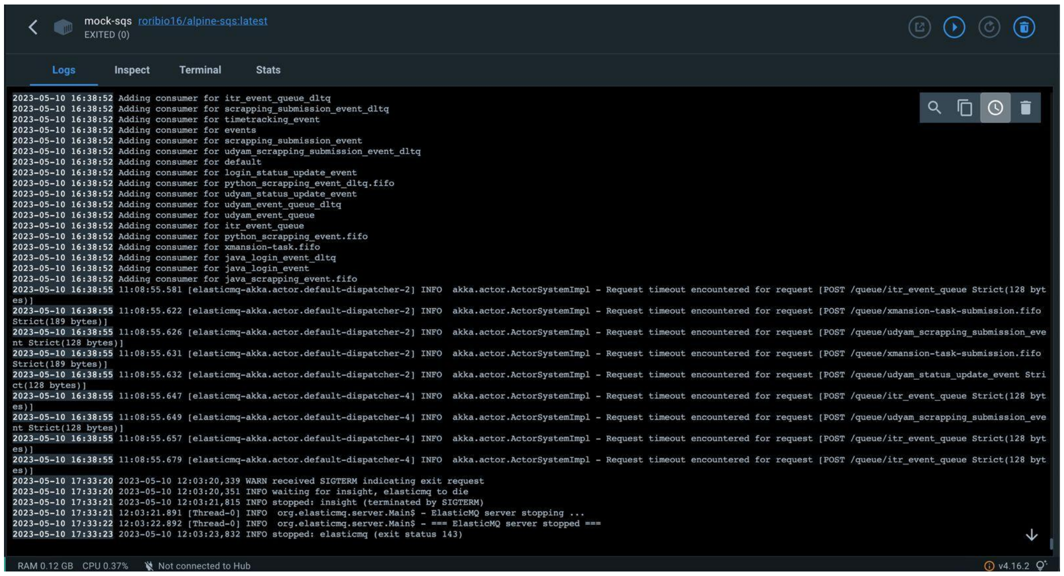
Figure 21: Third-party integration of APIs

3) Wrote test cases for multiple services in our code
4) Deployed aws queue(SQS) for handling concurrency in our services

This refers to using SQS as a messaging system to manage the processing of multiple requests or tasks concurrently. At the point when various services or parts need to interact with one another, simultaneousness can turn into a problem. AWS SQS helps address this issue by decoupling the sender and recipient parts. It gives a dependable and versatile message queueing service that permits one part to send messages (requests or tasks) to a line while another part (worker) processes these messages asynchronously.

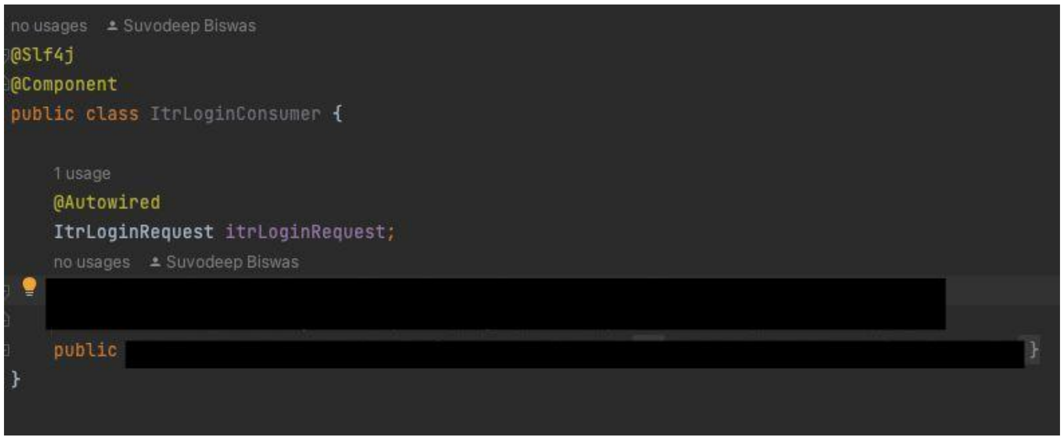
There are 3 components to this.

- Sender (one who sends messages to an SQS Queue)
- Queue (ensures the messages are not lost)
- Worker (one who processes messages)



```
2023-05-10 16:38:52 Adding consumer for itr_event_queue_dltq
2023-05-10 16:38:52 Adding consumer for scrapping_submission_event_dltq
2023-05-10 16:38:52 Adding consumer for timetracking_event
2023-05-10 16:38:52 Adding consumer for events
2023-05-10 16:38:52 Adding consumer for scrapping_submission_event
2023-05-10 16:38:52 Adding consumer for udyam_scrapping_submission_event_dltq
2023-05-10 16:38:52 Adding consumer for default
2023-05-10 16:38:52 Adding consumer for login_status_update_event
2023-05-10 16:38:52 Adding consumer for python_scrapping_event_dltq.fifo
2023-05-10 16:38:52 Adding consumer for udyam_status_update_event
2023-05-10 16:38:52 Adding consumer for udyam_event_queue_dltq
2023-05-10 16:38:52 Adding consumer for udyam_event_queue
2023-05-10 16:38:52 Adding consumer for itr_event_queue
2023-05-10 16:38:52 Adding consumer for python_scrapping_event.fifo
2023-05-10 16:38:52 Adding consumer for xmansion-task.fifo
2023-05-10 16:38:52 Adding consumer for java_login_event_dltq
2023-05-10 16:38:52 Adding consumer for java_login_event
2023-05-10 16:38:52 Adding consumer for java_scrapping_event.fifo
2023-05-10 16:38:55 11:08:55.581 [elasticmq-akka.actor.default-dispatcher-2] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/itr_event_queue Strict(128 byt
es)]
2023-05-10 16:38:55 11:08:55.622 [elasticmq-akka.actor.default-dispatcher-2] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/xmansion-task-submission.fifo
Strict(189 bytes)]
2023-05-10 16:38:55 11:08:55.626 [elasticmq-akka.actor.default-dispatcher-2] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/udyam_scrapping_submission_eve
nt Strict(128 bytes)]
2023-05-10 16:38:55 11:08:55.631 [elasticmq-akka.actor.default-dispatcher-2] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/xmansion-task-submission.fifo
Strict(189 bytes)]
2023-05-10 16:38:55 11:08:55.632 [elasticmq-akka.actor.default-dispatcher-2] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/udyam_status_update_event Stri
ct(128 bytes)]
2023-05-10 16:38:55 11:08:55.647 [elasticmq-akka.actor.default-dispatcher-4] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/itr_event_queue Strict(128 byt
es)]
2023-05-10 16:38:55 11:08:55.649 [elasticmq-akka.actor.default-dispatcher-4] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/udyam_scrapping_submission_eve
nt Strict(128 bytes)]
2023-05-10 16:38:55 11:08:55.657 [elasticmq-akka.actor.default-dispatcher-4] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/itr_event_queue Strict(128 byt
es)]
2023-05-10 16:38:55 11:08:55.679 [elasticmq-akka.actor.default-dispatcher-4] INFO  akka.actor.ActorSystemImpl - Request timeout encountered for request [POST /queue/itr_event_queue Strict(128 byt
es)]
2023-05-10 17:33:20 2023-05-10 12:03:20,339 WARN received SIGTERM indicating exit request
2023-05-10 17:33:20 2023-05-10 12:03:20,351 INFO waiting for insight, elasticmq to die
2023-05-10 17:33:21 2023-05-10 12:03:21,815 INFO stopped: insight (terminated by SIGTERM)
2023-05-10 17:33:21 12:03:21.891 [Thread-0] INFO  org.elasticmq.server.Main$ - ElasticMQ server stopping ...
2023-05-10 17:33:22 12:03:22.892 [Thread-0] INFO  org.elasticmq.server.Main$ - === ElasticMQ server stopped ===
2023-05-10 17:33:23 2023-05-10 12:03:23,832 INFO stopped: elasticmq (exit status 143)
```

Figure 22: SQS using Docker



```
no usages  Suvodeep Biswas
@Slf4j
@Component
public class ItrLoginConsumer {

    1 usage
    @Autowired
    ItrLoginRequest itrLoginRequest;
    no usages  Suvodeep Biswas
    [redacted]
    public [redacted] }
}
```

Figure 23: SQS using Java

5) Refactored highly complicated code into more readable
6) Solved p0 which are the emergent bugs that occurred in production

Chapter 5: Conclusions

All in all, the project planned to upgrade the functional viability and smooth out the presentation of a verification dashboard utilizing Java Spring Boot, Python and AWS services. By utilizing these advances, the project effectively accomplished its goals. The usage of Java Spring Boot as the essential backend language took into account quick turn of events, simplified configuration, and easy integration with different parts. The Spring environment gave a large number of libraries that worked with the execution of critical functionalities like client management, data handling, verification work processes, task, etc.

The joining of AWS services, especially AWS EC2, S3, further better the adaptability, dependability, and accessibility of the verification platform. The dashboard additionally used AWS SQS (Simple Queue Service) to deal with concurrency in administrations, guaranteeing proficient task handling and keeping away from bottlenecks. This enhanced the overall experience and responsiveness of the verification stage.

Overall, the project's fruitful execution of Java Spring Boot, AWS benefits, and fitting structural decisions brought about an improved verification dashboard. It exhibited the significance of using the right innovations and utilizing cloud administrations to enhance functional viability, smooth out the execution, and convey a powerful and productive verification solution.

References

1. https://www.tartanhq.com/

2. https://www.java.com/

3. https://spring.io/

4. https://start.spring.io/

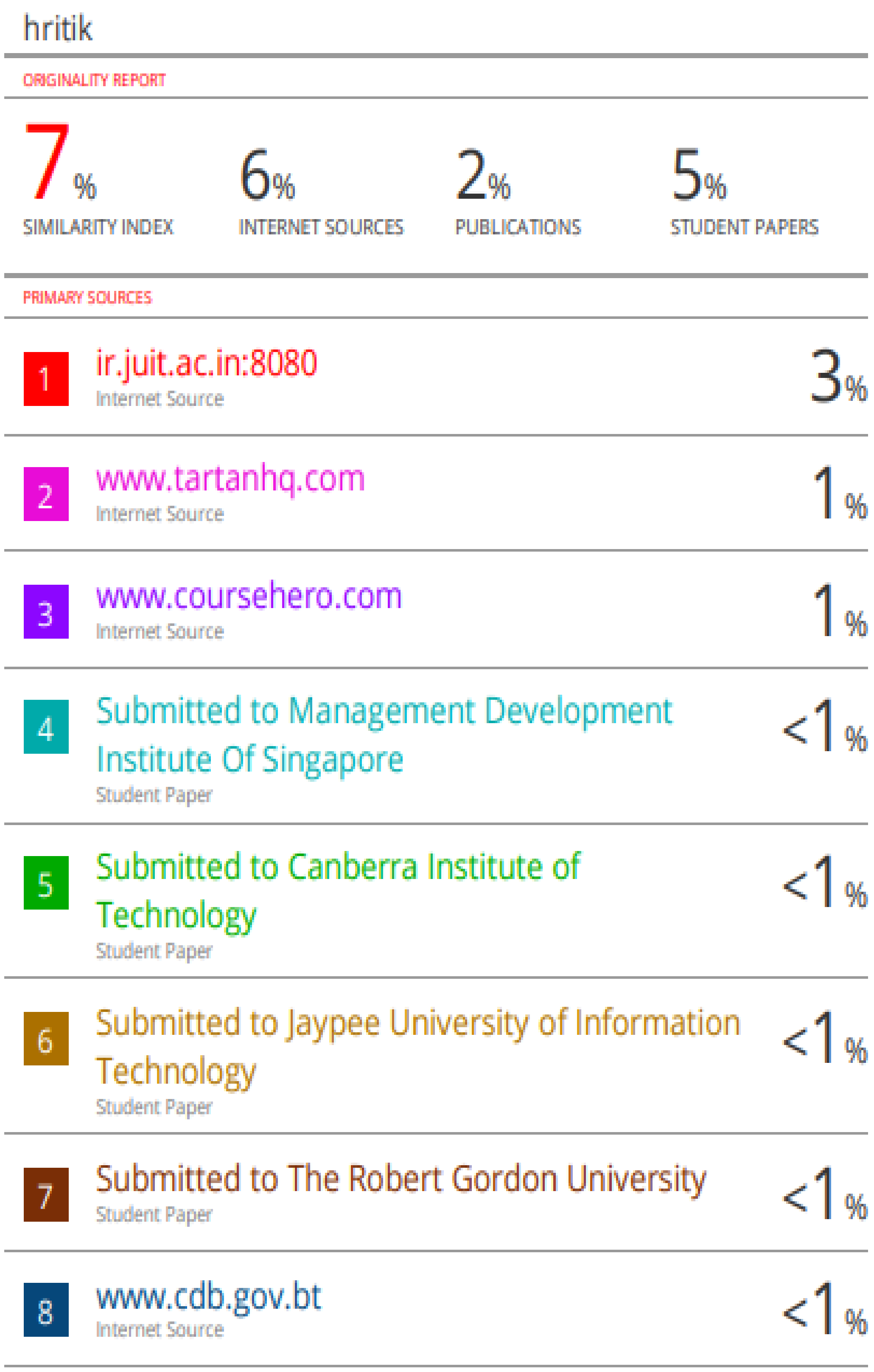
5. https://www.javatpoint.com/spring-boot-tutorial

6. https://aws.amazon.com/lambda/

7. https://www.python.org/

8. https://aws.amazon.com/ec2/

9. https://aws.amazon.com/s3/



hritik

ORIGINALITY REPORT

7%	6%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ir.juit.ac.in:8080 Internet Source	3%
2	www.tartanhq.com Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	Submitted to Management Development Institute Of Singapore Student Paper	<1%
5	Submitted to Canberra Institute of Technology Student Paper	<1%
6	Submitted to Jaypee University of Information Technology Student Paper	<1%
7	Submitted to The Robert Gordon University Student Paper	<1%
8	www.cdb.gov.bt Internet Source	<1%